MÉTODOS DE SELEÇÃO DE VARIÁVEIS COM APRENDIZADO DE MÁQUINA PARA ESTRATÉGIAS DE ARBITRAGEM ESTATÍSTICA CONFIGURADAS COMO PREDIÇÃO DE SÉRIES TEMPORAIS

Nicholas Barbosa Richers

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Produção, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Produção.

Orientador: Carlos Alberto Nunes Cosenza

Rio de Janeiro
Fevereiro de 2022

MÉTODOS DE SELEÇÃO DE VARIÁVEIS COM APRENDIZADO DE MÁQUINA PARA ESTRATÉGIAS DE ARBITRAGEM ESTATÍSTICA CONFIGURADAS COMO PREDIÇÃO DE SÉRIES TEMPORAIS

Nicholas Barbosa Richers

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE PRODUÇÃO.

Orientador: Carlos Alberto Nunes Cosenza

Aprovada por: Prof. Carlos Alberto Nunes Cosenza
Prof. Cesar das Neves
Prof. Claudio Henrique dos Santos Grecco
Prof. Roberto Ivo da Rocha Lima Filho

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2022

Richers, Nicholas Barbosa

Métodos de Seleção de Variáveis com Aprendizado de Máquina para Estratégias de Arbitragem Estatística Configuradas como Predição de Séries Temporais/Nicholas Barbosa Richers. – Rio de Janeiro: UFRJ/COPPE, 2022.

XVII, 147 p.: il.; 29, 7cm.

Orientador: Carlos Alberto Nunes Cosenza

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Produção, 2022.

Referências Bibliográficas: p. 142 – 146.

1. Arbitragem Estatística. 2. Aprendizado de Máquina. 3. Análise de Regressão. 4. Seleção de Variáveis. 5. Agrupamento Hierárquico. 6. Viés de Seleção. I. Cosenza, Carlos Alberto Nunes. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Produção. III. Título.

"Não são as espécies mais fortes
que sobrevivem, nem as mais
inteligentes e sim as mais
suscetíveis a mudanças"
Charles Darwin
Mt20:16

Agradecimentos

Ao meu pai por me proporcionar tudo que foi necessário ao longo desta jornada e por seu apoio incondicional.

Ao Professor Cosenza, que me recebeu de braços abertos e deu toda a liberdade para desenvolver esse trabalho da melhor forma possível.

A todos que se desprenderam de seus afazeres para ler este trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MÉTODOS DE SELEÇÃO DE VARIÁVEIS COM APRENDIZADO DE MÁQUINA PARA ESTRATÉGIAS DE ARBITRAGEM ESTATÍSTICA CONFIGURADAS COMO PREDIÇÃO DE SÉRIES TEMPORAIS

Nicholas Barbosa Richers

Fevereiro/2022

Orientador: Carlos Alberto Nunes Cosenza

Programa: Engenharia de Produção

A pesquisa aplicada em finanças é dificultada pelo acesso a dados de boa qualidade, o que acarreta no desenvolvimento de estratégias falso positivas. Neste trabalho os dados serão fornecidos por um torneio, organizado por fundo de risco neutro que configurou um problema de arbitragem estatística como um problema de predição de séries temporais. Nesse contexto, dispomos de uma quantidade substancial de variáveis e nos apoiamos na literatura de seleção de variáveis a fim de desenvolver estratégias capazes de superar um dado referencial com consistência.

Inicialmente as predições geradas por um estimador serão decompostas a fim de observar a origem do excesso de volatilidade. A seguir, será exposto um método de avaliação que contabiliza todas as tentativas a fim de garantir retornos positivos. Implementou-se diversos métodos de seleção de variáveis para atenuar a multicolinearidade e foi proposta uma pequena extensão para um algoritmo de agrupamento de atributos para minimizar o risco de superestimar variáveis sem importância.

Para atingir melhores resultados, removeu-se o ruído da matriz de correlação e utilizou-se uma medida de codependência robusta a relações não lineares. Em seguida, foi desenvolvido um mapeamento de regimes que descrevem o comportamento dos dados ao longo do tempo, porém não foi possível prever com precisão a mudança de regime. Apesar de resultados muito bons quando a suposição é verdadeira.

Esta pesquisa é limitada à literatura de seleção de variáveis. Contudo, dentro deste escopo não é possível afirmar que o referencial foi superado com consistência, apesar de o ter feito na maior parte das observações. Por essa razão algoritmos de paridade de risco foram desenvolvidos como uma solução de contorno.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

FEATURE SELECTION METHODS WITH MACHINE LEARNING FOR A STATISTICAL ARBITRAGE STRATEGIES CONFIGURED AS FORECASTING PROBLEM

Nicholas Barbosa Richers

February/2022

Advisor: Carlos Alberto Nunes Cosenza

Department: Production Engineering

Applied research in quantitative finance has high entry-barrier due the lack of high-quality data, which leads to false positive strategies. In present work the data is provided by a tournament, organized by a market-neutral hedge fund who configured a statistical arbitrage challenge as forecasting problem. In this context, there is a substantial amount of features and based on the feature selection literature the task is to develop strategies to consistently outperform a given benchmark.

Initially, the predictions created by a tree-based estimator were decomposed into linear and non-linear portions to estimate the source of volatility excess. Next, a method which accounts for all trials in order to ensure positive returns were introduced. Several feature selection methods to attenuate the multicollinearity were implemented and a small extension to a feature clustering algorithm was proposed to decrease the risk of overestimating unimportant variables.

To enhance better results we denoised the correlation matrix and proposed a codependency measure robust to nonlinear relationships. Then, was made an attempt to map regimes that describe market behavior over time, but it was not possible to accurately predict regime change. Despite reasonable results when the assumption is true.

This research is limited to feature selection literature, but within this scope we were not able to state that the benchmark can be consistently overcomed, despite having done so most of the time. So we rely on the parity risk algorithms literature to present a workaround solution for the problem.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Neste capítulo serão abordados a apresentação e a descrição do problema, seguidos pela justificativa para a realização deste estudo e por fim seus objetivos e resultados esperados.

1.1 Apresentação do Problema

Um fundo de **risco neutro** busca evitar completamente alguma forma específica de risco de mercado. Uma ação de uma empresa está sujeita a diversos tipos de risco, principalmente relacionados ao setor da empresa, ou relacionado ao país onde a empresa opera. A forma mais comum de se montar uma estratégia neutra para o mercado é através da negociação em pares (*pair trading*). O gestor irá combinar posições compradas (*long*) e vendidas (*short*), de diferentes ações a fim de aumentar os retornos obtidos por fazer uma boa seleções de ações e ainda diminuir o risco causado por movimentos de mercado mais amplos[1].

Um gestor que acompanha as Ações A e B, onde a ação A é de uma empresa com excelente corpo diretivo, crescimento de lucros consistentes, além de pertencer a um setor que se beneficia do ambiente macroeconômico no médio prazo. Por outro lado, temos a ação B que é do mesmo setor da ação A e a empresa não possui as mesmas qualidades, mas também tem aproveitado o momento favorável para o segmento. Caso haja uma mudança no cenário macro, é esperado que a Ação B venha a cair mais que a ação A. Neste exemplo o investidor deve montar uma operação comprada em A e vendida em B.

Uma outra estratégia adotada por fundos neutros é a arbitragem estatística. Os fundos neutros que adotam arbitragem estatística se utilizam de algoritmos e métodos quantitativos para encontrar discrepâncias de preços em ações com base nos

[1] What is a market neutral investment strategy?
https://www.fidelity.ca/fidca/en/investor/marketneutralstrategy (Acessado em 09/01/2022)

dados históricos. Então, com base nesses resultados, os gestores montarão posições com ações esperando uma reversão a média histórica.

O investidor deverá identificar duas ações altamente correlacionadas, em geral acima de 0.80 e seguindo as correlações dos pares de ações por meio de métodos estatísticos, um investidor deverá buscar uma posição comprada nas ações de baixo desempenho relativo e uma posição vendida nas ações de desempenho relativo acima do esperado, quando a correlação entre esses dois ativos se desviar de sua norma histórica. Esta operação busca lucrar com a **correção da correlação**, que deve retornar ao seu nível histórico. Se bem-sucedida, a convergência de preços resulta em ganhos tanto da posição comprada quanto da posição vendida.

Um grande benefício dos fundos neutros é a ênfase na mitigação do risco de mercado. Em tempos de alta volatilidade, os resultados históricos mostram que os fundos neutros para o mercado provavelmente terão um desempenho superior ao dos fundos usando outras estratégias específicas [2].

O mercado de fundos quantitativos de risco neutro tem a presença de gestores globais de peso como a AQR Capital Management, que até meados de 2019 possuía o Professor Marcos López de Prado como líder do time de pesquisa e a Renaissance Technologies, que é conhecida por manter uma equipe quase que exclusivamente por formada por PhD's sem base em finanças ou administração, como astrofísicos e revela possuir foco em modelos quantitativos baseados em **ciência de dados** e aprendizado de máquina, **aplicados ao mercado financeiro**.

Nesse contexto, o problema de predição do comportamento dos preços de ações é muito difícil. Há anos fundos quantitativos tentam abordar esse problema, porém nem sempre com sucesso[3].

Em paralelo, a área de aprendizado de máquina cresceu muito na ultima década, alavancada pelo crescimento do poder de processamento e pela disponibilidade dos dados [19]. A popularização dessas técnicas culminou no desenvolvimento de uma comunidade relevante de cientista de dados, que aplicam essas técnicas nas mas diversas áreas como saúde, educação e até serviços financeiros como análise de crédito. Contudo, a maioria das aplicações para o mercado financeiro são extremamente desafiadoras, entre várias outras razões, devido ao dinamismo do mercado [20] e também a enorme quantidade de ruído presente nos dados.

[2]Market Neutral Fund Definition.
https://www.investopedia.com/terms/m/market-neutral-fund.asp (Acessado em 09/11/2021)

[3]Quant Funds Struggled in 2019. The Outlook for This Year Is More of the Same.
https://www.barrons.com/articles/quant-funds-struggled-in-2019-the-outlook-for-this-year-is-more-of-the-same-51579782601 (Acessado em 23/10/2020)

1.1.1 O Desafio da Pesquisa Aplicada

Além dificuldades citadas na seção 1.1, há uma barreira de entrada para a pesquisa aplicada nessa área, entre elas [21][22]:

1. Dominar todas as áreas relacionadas como, curadoria de dados, computação de alto desempenho e modelagem de indicadores financeiros

2. Acesso a dados de boa qualidade que geralmente só estão disponíveis a investidores institucionais devido aos altos custos

3. Estratégias falso positivas. A falta de um fluxo contínuo de dados impossibilita o pesquisador a verificar as estratégias elaboradas, que tendem a funcionar apenas em seu conjuntos de dados estático

DE PRADO e FABOZZI [22] discutem a formação de um torneio, onde um organizador formaliza um desafio de estratégias investimento no formato de um problema preditivo (aprendizado supervisionado), para incentivar a pesquisa de maneira coletiva (*crowdsourcing*). O torneio divide o problema em três tarefas altamente especializadas: (i.) estruturar os dados de forma a constituir um problema preditivo, que é executada pelo organizador que possui uma equipe com extenso conhecimento em finanças. (ii.) Modelagem, que é feita pelos participantes do torneio, que em geral são pesquisadores com conhecimento de **ciência de dados** e estatística. (iii) Operações, que é responsável por traduzir a informação gerada pelos participantes em compra e venda de ativos, que também fica a cargo do organizador. A partir desse arranjo, o torneio pretende superar as barreiras citadas:

1. Os organizadores do torneio dispõem de seu conhecimento em finanças para definir o problema de **maneira restrita**, assim participantes do torneio podem trabalhar nesse desafio mesmo que não possuam conhecimento na área.

2. Com encriptação o organizador consegue impedir que os pesquisadores utilizem os dados fora do torneio, respeitando assim os direitos de confidencialidade dos provedores de dados.

3. Remunerando os pesquisadores sobre o desempenho fora da amostra, o organizador elimina qualquer incentivo ao sobreajuste nos dados de treinamento (dentro da amostra). Além disso os pesquisadores são incentivados a investir o próprio dinheiro em suas previsões a fim de alavancar seus ganhos.

Resumidamente, o torneio permite que uma ampla comunidade de cientistas de dados que não possuem acesso a todos os recursos necessários, contribuir com um fundo de investimentos enviando previsões de maneira **sistemática** e **não discricionária**. Finalmente, DE PRADO e FABOZZI [22] declaram que os torneios devem ajudar com a desintermediação e democratização do setor de serviços financeiros.

1.2 Descrição do Problema de Pesquisa e Hipóteses

Agora há um problema de arbitragem estatística que foi configurado como um problema de aprendizado supervisionado (**regressão**), detalhado na seção 3.1. Há também um conjunto de dados limpo e de boa qualidade, porém encriptado, descrito na seção 3.2.1. Além disso, herdou-se todo um arcabouço de métricas (ver seção 4.3.1) para um diagnóstico preciso das estratégias geradas e por fim as estratégias serão remuneradas de acordo um uma métrica de performance bem definida (seção 4.2).

Diante desse arranjo, o problema de pesquisa se resume a encontrar um modelo capaz de superar um dado referencial e gerar retornos financeiros de maneira **consistente**. Após uma extensa investigação de diversos métodos da literatura de ciência de dados aplicada a finanças, concluiu-se que uma investigação mais profunda na literatura de **seleção de variáveis** é o alternativa mais promissora para atingir o nosso objetivo.

1.3 Objetivo Geral

Este estudo objetiva realizar uma investigação das principais técnicas presentes na literatura de aprendizado de máquina aplicado a finanças, em especial a parte de **seleção de variáveis**. Dada a extensão e o grande número de técnicas desenvolvidas, a base teórica dessas técnicas será desenvolvida de maneira concisa.

1.3.1 Objetivos Específicos

Muitas das aplicações aqui tratadas são vistas como uma disciplina a parte na área de aprendizado de máquina [12], entre elas trataremos de:

- Estruturar o problema de arbitragem estatística como aprendizado supervisionado
- Apresentar o conjunto de dados utilizado
- Analisar a performance de diferentes modelos de aprendizado de máquina
- Investigar a composição do sinal encontrado por cada modelo
- Compreender um conjunto de métricas financeiras
- Desenvolver um método para identificar estratégias falso positivas
- Implementar métodos de seleção de variáveis

- Desenvolver modelos personalizados para diferentes regimes do mercado
- Comparar técnicas de otimização de portfólio
- Desenvolver um comparativo entre esses modelos com dados fora da amostra

1.4 Justificativa

DE PRADO [23] menciona que os fundos quantitativos mais bem sucedidos da indústria se baseiam nas técnicas abordadas ao longo deste trabalho, além disso entende-se que este estudo de caso, apesar de se situar no âmbito financeiro, tem a oportunidade de explorar diversas técnicas de aprendizado de máquina que podem ser usadas em outras aplicações por um cientista de dados.

1.5 Resultados Esperados

Por se tratar de uma competição, há um conjunto de métricas muito bem definidas que mensuram se um modelo é bem sucedido ou não. É esperado que haja uma evolução cronológica na performance dos modelos apresentados. Além disso, teremos a oportunidade de observar o desempenho desses modelos em dados nunca antes vistos (capítulo 9), afastando suposições de sobreajuste (*overfiting*). Ao final devemos ser capazes de concluir se a partir das técnicas desenvolvidas fomos capazes de superar um determinado referencial com **consistência**.

1.6 Roteiro da Pesquisa

Ao longo desta pesquisa abordaremos os seguinte temas:

1. Revisão Bibliográfica
2. Dados do Problema
3. Modelo Base e Métricas
4. Neutralização e Detecção de Estratégias Falso Positivas
5. Seleção de Variáveis
6. Detecção de Regimes
7. Otimização de portfólio
8. Resultados

Capítulo 2

Revisão Bibliográfica

Ao longo deste capítulos será desenvolvida uma revisão dos principais temas teóricos abordados no decorrer do trabalho, como tipos de aprendizados de máquina e seus respectivos modelos, além das técnicas de avaliação e otimização de hiperparâmetros. Nesse contexto, EVSUKOFF [1] divide as tarefas de mineração de dados em tarefas preditivas (aprendizado supervisionado) e tarefas descritivas (aprendizado não supervisionado).

2.1 Aprendizado Supervisionado

DIXON *et al.* [24] realizou um comparativo entre modelos de aprendizado de máquina e modelagem estatística. O autor afirma que o aprendizado de supervisionado é frequentemente uma forma algorítmica de estimar um modelo estatístico em que o processo de geração de dados é tratado como desconhecido. Essa afirmação difere fundamentalmente dos estimadores de máxima verossimilhança usados comumente em modelos estatísticos, que presumem que os dados foram gerados pelo modelo e normalmente sofrem com sobreajuste, especialmente quando aplicados a conjuntos de dados de alta dimensão [24].

DE PRADO [23] cita que modelos de aprendizado supervisionado requerem uma matriz de dados estruturada (geralmente no formado de dados tabulares), onde as colunas são referentes as variáveis (*features*) do problema e as linhas são as amostras. Além disso há uma variável alvo (*target*) contendo um valor específico para cada amostra. Na seção 3.2.1, os dados do problema estão disponibilizados exatamente dessa maneira. Contudo existem diferentes tipos de modelos de aprendizado supervisionado, como detalharemos a seguir.

2.1.1 Modelos de Classificação

O objetivo de um modelo de classificação, é desenvolver um heurística capaz de predizer a classe correta de um registro a partir dos valores das variáveis de entrada [1]. De forma geral essas classes são valores categóricos, não numéricos, ou valores discretos que descrevem as classes. A figura 2.1 mostra a fronteira de decisão criada por um modelo de classificação com três classes.

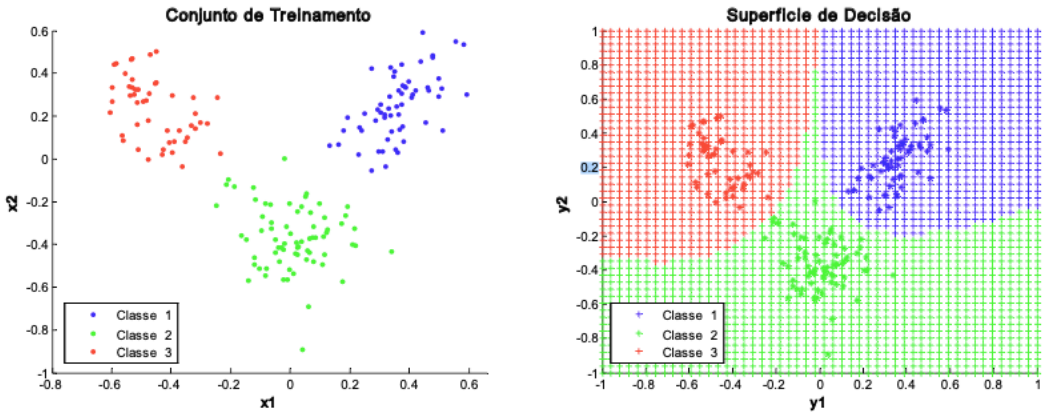


Figura 2.1: Modelos de Classificação (Retirado de EVSUKOFF [1])

2.1.2 Modelos de Regressão

Um modelo de regressão busca através da aproximação de uma função realizar a estimativa do valor de uma variável numérica, a partir do conjunto de variáveis de entrada [1].

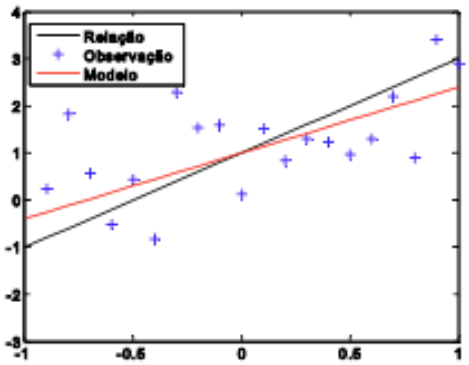


Figura 2.2: Modelos de Regressão (Retirado de EVSUKOFF [1])

2.1.3 Modelos de Ranqueamento

Além dos dois modelos citados, há os modelos de ranquemento, muito utilizados em sistemas de recomendação, processamento de linguagem natural, ordenamento de postagens em redes sociais e especialmente sistemas de busca, sendo essa última a base de toda a disciplina de recuperação da informação (*Information Retrieval*) [25]. Por essa razão, a grande maioria dos exemplos didáticos contidos na literatura seguem esse caso de uso, mas este pode ser facilmente estendido para outros problemas.

Modelos de ranqueamento aprendem de maneira direta a ranquear uma lista a partir de um modelo treinado para predizer a probabilidade de uma certa amostra estar na frente de outra $P(rank(i) > rank(j))$. Isso é feito através de uma função de score entre essas duas amostras s_i e s_j. O modelo então pode ser treinado usando um gradiente estocástico em uma função de custo definida sobre o score, onde na opinião de BURGES *et al.* [26] a função sigmoide (equação 2.1), é uma escolha natural.

$$P(rank(i) > rank(j)) = \frac{1}{1 + e^{-(s_i - s_j)}} \tag{2.1}$$

Dessa forma, devemos fazer um ranqueamento por pares *Pairwise Ranking*, que recebe como entrada um par de amostras contendo as variáveis de entrada de ambas. O modelo deverá retornar a preferencia, representado por 1, -1, entre cada par de amostras. Assim, iterativamente o gradiente estocástico ajusta o score atribuído para casa amostra. O tamanho do ajuste é determinado pela diferença dos scores $(s_i - s_j)$. Lembrando que o objetivo final é apenas a ordenação da lista.

Finalmente, o modelo então deve minimizar uma função de custo que representa a quantidade de permutações necessárias para ordenar a lista de forma correta. Ao tratarmos o problema abordado neste estudo, veremos na seção 4.3.1, que o único requerimento para uma função de custo é que ela seja capaz de produzir predições co-monotônicas com a variável alvo. Mais detalhes sobre o modelo de ranqueamento performando neste estudo serão dados da seção 2.2.2.3.

2.2 Modelos de Aprendizado de Máquina Supervisionados

Nessa seção abordaremos alguns dos modelos de aprendizado de máquina supervisionados que serão utilizados no decorrer deste estudo.

2.2.1 Modelos Lineares

2.2.1.1 Regressão Linear

A regressão linear é uma técnica clássica da estatística, que muitas vezes é utilizada como base para avaliar a performance de outros modelos de regressão. Existem algumas abordagens diferentes como estimação de máxima verossimilhança e descida gradiente para obtermos os coeficientes da regressão linear, contudo, neste estudo usaremos o método dos mínimos quadrados, que além de simples, usaremos extensões de sua formulação para outros modelos lineares.

Esse procedimento busca minimizar a soma do quadrados dos resíduos e dessa forma estimar uma solução única para o vetor de parâmetros (w). Partindo do problema de regressão original $y = X.w$ temos que [1]:

$$w = (X^T.X)^{-1}.X^T.y \tag{2.2}$$

Onde $(X^T.X)^{-1}.X^T$, é a pseudo-inversa de X. Assim, reformulando a equação 2.2 para o método dos mínimos quadrados [7]:

$$\min_w ||Xw - y||_2^2 \tag{2.3}$$

Onde $||.||_2$ é a norma L2 do vetor resultante.

2.2.1.2 Ridge

O Ridge, como vemos na equação 2.4, é uma extensão do método dos mínimos quadrados somado a componente de regularização [7].

$$\min_w ||Xw - y||_2^2 + \alpha||w||_2^2 \tag{2.4}$$

O parâmetro de regularização $||w||_2$ penaliza coeficientes de valor alto, impedindo que o modelo confie demais em poucas variáveis. Já o parâmetro $\alpha \geq 0$ controla a penalização. Quanto maior o valor de α, maior a penalização para a complexidade, portanto os coeficientes se tornam mais robustos à colinearidade.

2.2.1.3 Orthogonal Matching Pursuit

Esse modelo se aproxima de um modelo linear, adicionando uma restrição ao número de coeficientes diferentes de zero ou norma L0 [7].

$$arg\min_w ||y - Xw||_2^2 \text{ subject to } ||w||_0 \leq n_{\text{nonzero\_coefs}} \tag{2.5}$$

Essa restrição se comporta como um método de seleção de variáveis, baseado em

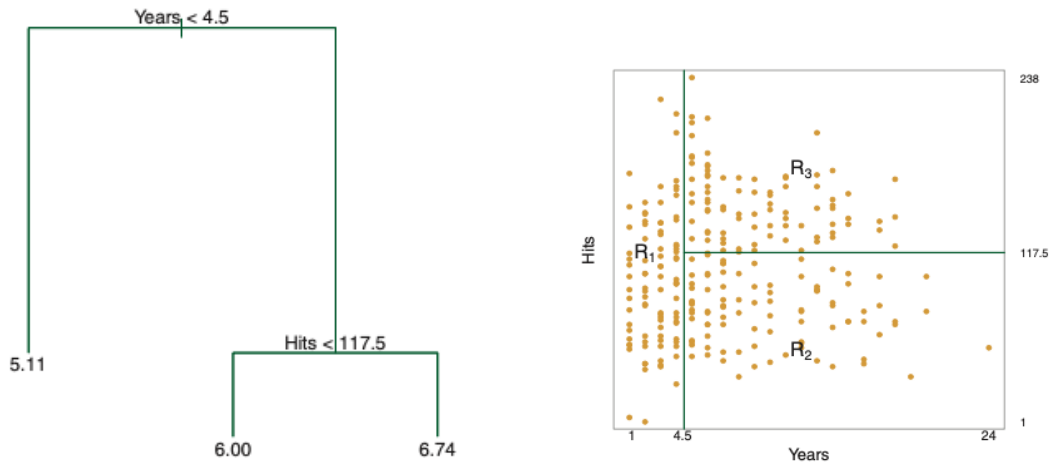
um algoritmo guloso, que a cada passo se torna mais correlacionado com o resíduo resultante.

2.2.2 Modelos Não Lineares

Ao longo dessa seção iremos detalhar alguns dos modelos de árvores de regressão utilizados ao longo deste estudo de maneira sequencial, do mais simples ao mais complexo, que utilizam os modelos mais simples como base.

2.2.2.1 Árvore de Regressão

Pela definição de JAMES *et al.* [2], um modelo de árvore de regressão consiste em uma sequência de regras de partição (*split*) dos registros começando pelo topo da árvore também designado como nó raiz e recursivamente repetindo o procedimento até chegar aos nós folhas, que representam o valor efetivamente predito pelo modelo, como visto na figura 2.3a, que prediz o salário em base logarítmica de um jogador de basebol com base na sua experiência e quantidade de rebatidas.



(a) Modelo Árvore (b) Partição Árvore

Figura 2.3: Modelo de Árvore (Retirados de JAMES *et al.* [2])

A figura 2.3b, ilustra como um modelo de árvore reparte os registros em um sub espaço de duas dimensões. O processo de construção da árvore se divide basicamente em duas etapas:

1. Dividir o espaço de variáveis de entrada $X_1, ..., X_p$ em todas as J regiões distintas e sem sobreposição $R_1, ..., R_J$.

2. Para cada observação (n), onde $n \in R_J$ atribuir um mesmo valor, geralmente a média das observações usadas para treino.

Por fim o objetivo é encontrar as regiões $R_1, ..., R_J$ que minimizam a soma do quadrado dos resíduos (RSS) dado pela equação 2.6 em um exemplo com apenas duas partições R_1 e R_2:

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \tag{2.6}$$

Contudo se extrapolarmos o passo 1 ao ponto de obtemos uma região R_j para cada amostra de treinamento e obtermos $RSS = 0$ incorreremos em sobreajuste dos dados de treino, que leva a resultados ruins na amostra de teste. Para lidar com esse problema, os algoritmos de árvore dispõem de diversos hiperparâmetros que impedem que a árvore cresça indefinidamente, como por exemplo um número mínimo de amostras por partição.

Uma outra maneira bastante popular e descrita por JAMES *et al.* [2] é a **poda**, onde deixa-se a árvore crescer até que cada região tenha ao menos a quantidade mínima de amostras e então é adicionado um parâmetro de penalização (α) para cada folha da árvore na fórmula do RSS, como descrito na equação 2.7.

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha|T| \tag{2.7}$$

A figura 2.4 ilustra o processo de partição antes e depois da poda da árvore, que remove algumas folhas (ou regiões pela figura 2.4). O parâmetro de penalização (α) assim como o mínimo de amostras por folha são hiperparâmetros a serem otimizados e dependem das características do conjunto de dados.

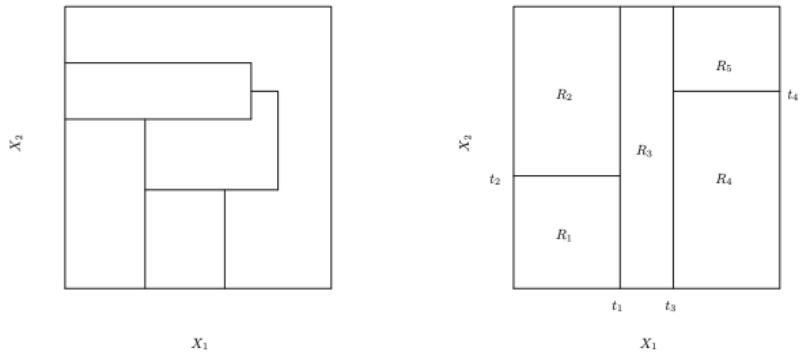


Figura 2.4: Efeito da Poda da Árvore (Retirados de JAMES *et al.* [2])

2.2.2.2 Conjunto de Árvores de Regressão

A ideia de podar uma árvore de decisão para evitar que o modelo simplesmente memorize os dados de treino e mesmo com uma performance inferior nos dados de

treino obter uma performance superior na base de teste, remete ao problema de **viés e variância** ilustrado na figura 2.5 e muito debatido na literatura de aprendizado de máquina. Quanto mais complexo for o modelo em termo de grau de liberdade, menor será o viés contudo sofrerá com maior variância [3].

Figura 2.5: Viés e Variância

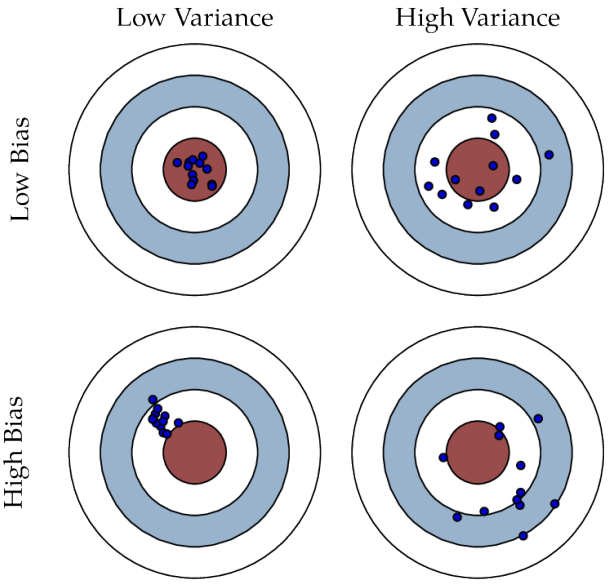


Figura 2.6: Viés e Variância (Retirado de DOMINGOS [3])

O aprendizado em conjunto (*ensemble learning*) é um paradigma de aprendizado de máquina onde múltiplos modelos fracos são treinados para resolver o mesmo problema e combinados para obter uma melhor performance [2]. Nesse contexto existem duas técnicas bastante populares que se baseiam nesse paradigma:

Bootstrap Aggregation (Bagging)

O propósito do *Bagging* é reduzir a variância das predições combinando diversos (normalmente centenas) de modelos similares que são treinados em uma amostra (com reposição) da base de teste e então retorna-se a média das predições como visto na equação 2.8 [2].

$$\hat{f}_{bag} = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x) \tag{2.8}$$

O modelo de floresta aleatória *Random Forest*, é uma extensão do Bagging, uma vez que este gera **árvores descorrelacionadas** uma das outras, sorteando um subconjunto de colunas a serem utilizadas na construção de cada árvore (ver figura 2.7).

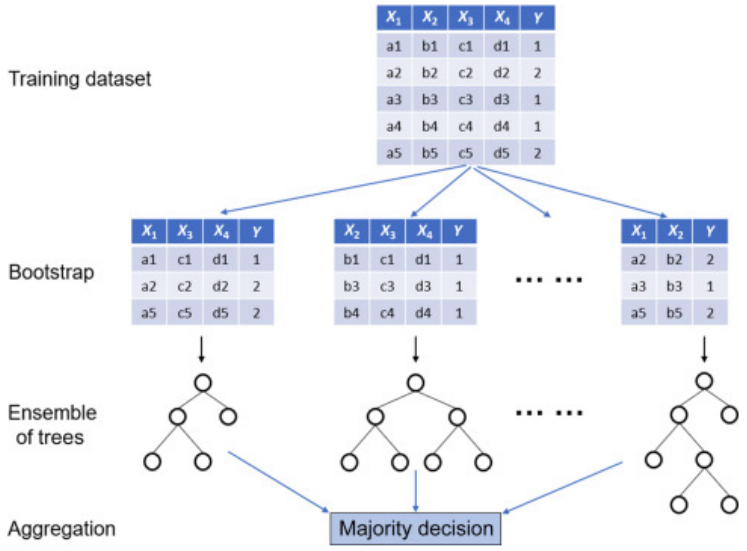


Figura 2.7: Floresta Aleatória (Retirado de MISRA e LI [4])

Boosting

A principal característica dos modelos de *boosting*, é que estes utilizam a informação gerada pelo modelo anterior (em geral milhares) para o treinamento do modelo [2] seguinte com o objetivo de minimizar o viés das predições. Dentro desta categoria, ainda podemos dividir em duas subcategorias. *Adaptative Boosting (Adaboost)* e *Gradient Boosting*.

Adaptative Boosting

As duas principais diferenças do *Adaboost* para o *Random Forest* são o tipo de árvores utilizadas e a maneira como o sorteio das amostras (*bootstrap*) é feito. No caso das árvores, o *adaboost* em geral utiliza apenas uma variável e um único ponto de corte (*stump*) como visto na figura 2.8.

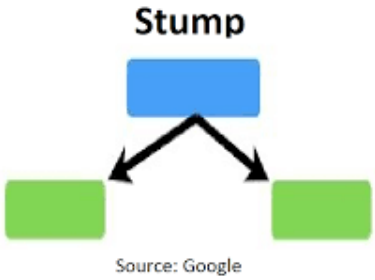


Figura 2.8: Árvore de Uma Variável (Retirado de MISRA e LI [4])

Para a primeira árvore, todas as amostras possuem a mesma probabilidade $(1/n)$ de serem sorteadas. A figura 2.9, apesar de se referir a um modelo de classificação,

ilustra bem o caso onde uma amostra com maior peso possui maior influência na decisão tomada pelo modelo.

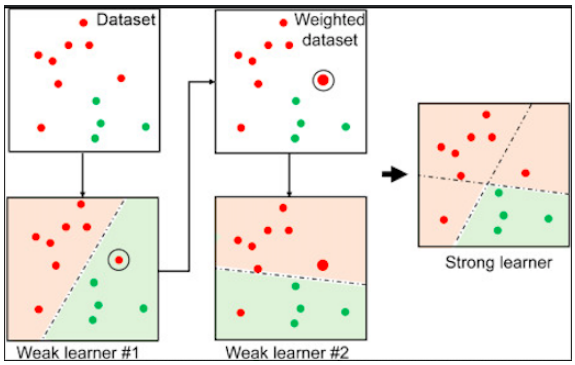


Figura 2.9: Atualização dos Pesos (Retirado de MISRA e LI [4])

Então, a partir da segunda árvore a probabilidade da amostra ser selecionada para o treino é atualizada em função do erro obtido pela árvore anterior. A m-ésima árvore terá uma importância α_m, na atualização dos pesos (ver equação 2.9).

$$\alpha_m = log\left(\frac{(1 - erro_m)}{erro_m}\right) \tag{2.9}$$

Modelos mais eficientes possuem maior importância. A seguir, a i-ésima amostra terá um peso (w_i) para o modelo seguinte (ver equação 2.10).

$$w_i \leftarrow w_i \cdot e^{\alpha_m} \tag{2.10}$$

Gradient Boosting

Modelos de *Gradient Boosting* também funcionam como uma sequência de árvores rasas (*shallow trees*), onde o resultado de cada árvore, influencia na seguinte. Contudo este difere do modelo *Adaboost*, uma vez que ao invés de atribuir pesos para cada amostra, as árvores são treinadas utilizando o resíduo (erro) da árvore anterior (ver figura 2.10 e após um novo treinamento, o resíduo é atualizado de acordo com o uma taxa de aprendizado (*learning rate*).

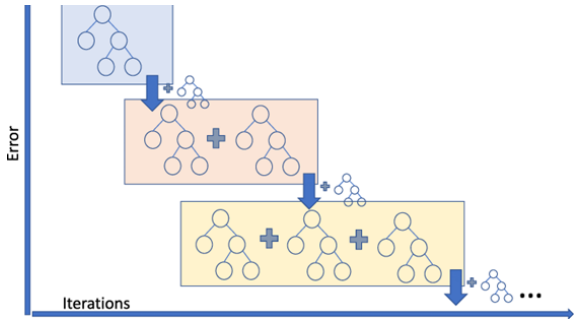


Figura 2.10: Sequência de Árvores Rasas (Retirado de MISRA e LI [4])

De acordo com HASTIE *et al.* [27], precisamos determinar uma função de custo diferenciável $L(y_i, \gamma)$, que para um problema de regressão normalmente se utiliza o erro quadrático (ver equação 2.6). Em seguida executamos o algoritmo em três passos:

1. Inicialize $f_0(x) = argmin_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$
2. Para $m = 1$ até N calcular:
   (a) Para $n = 1$ até M:

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$$

   (b) Treinar uma árvore de regressão para os alvos r_{im} determinando as partições R_{jm}, $j = 1, 2..., J_m$
   (c) Para $j = 1, 2..., J_m$, calcular:

$$\gamma_{jm} = \underset{\gamma}{argmin} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$$

   (d) Atualize $f_m(x) = f_{m-1}(x) + \eta \cdot \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$
3. Retorne $\hat{f}(x) = f_M(x)$

Uma vez determinada a função de custo, para o passo 1 o argumento que minimiza o resíduo é dado pela derivada da função de custo, que analiticamente é a média dos valores da variável alvo nos dados de treino. O passo 2 é uma recursão pelo total de árvores determinada (geralmente milhares) e é dividido em 4 etapas:

- (a) Calcular o valor dos resíduos (r_{im})
- (b) Treinar uma nova árvore a partir dos resíduos
- (c) Calcular os valores de saída das folhas (γ_{jm})
- (d) Calcular a predição da próxima árvore de acordo com a taxa de aprendizado (η)

Por fim o passo 3, onde obtemos o valor final ($f_M(x)$) predito pelo modelo somando os resíduos estimado por todas as M árvores.

Extreme Gradient Boosting (XGBoost)

O *XGBoost* é essencialmente a mesma coisa que o *Gradient Boosting*, a principal diferença está na maneira como as árvores residuais são construídas. No *XGBoost*, as árvores residuais são construídas calculando uma métrica de similaridade entre as folhas e os nós anteriores para determinar quais variáveis são usadas como raízes e nós. Essa medida de similaridade também é utilizada para a poda de nós da árvore a partir de um hiperparâmetro de regularização (λ) [28].

$$Similaridade = \frac{(\sum Residuos)^2}{Qtd.Residuos + \lambda} \tag{2.11}$$

A equação 2.11 nada mais é que o gradiente ao quadrado da função de custo ($L(y_i, f(x_i))$) dividido pela hessiana além do parâmetro de regularização, a demonstração dessa fórmula até chegar a expressão analítica da equação 2.11 pode ser encontrada no artigo original de CHEN e GUESTRIN [28]. Um outro modelo similar a este é o *LightGBM*, que também se diferencia na maneira como as árvores são construídas, focando em árvores mais profundas, mas com poucas folhas, mais informações em KE *et al.* [29].

2.2.2.3 XGBoost Ranker

Na prática, a única diferença entre o *XGBoost Ranker*, para o modelo de regressão ou classificação com *XGBoost*, é a função de custo utilizada para ajustar os parâmetros do modelo. Nesse caso, ao invés do erro médio quadrático, buscaremos minimizar a quantidade de permutações necessárias para ranquear a lista corretamente como comentado na seção 2.1.3.

Para essa tarefa o modelo *RankNet* usa o gradiente estocástico para minimizar a função de custo [30]. A partir equação 2.1, Essa função de custo deriva da função de entropia cruzada que busca penalizar a diferença da probabilidade medida com a desejada [31] e J_{ij} mede a probabilidade a amostra i estar a frente de j.

$$J_{ij} = log(1 + e^{(s_i - s_j)}) \tag{2.12}$$

Note pela equação 2.12 que J_{ij} aumenta de acordo com a diferença entre os scores das amostras $(s_i - s_j)$. Além disso estendendo para todo espaço de combinações de amostras contidos no conjunto de dados temos.

$$J = \sum_{(i,j) \in D} J_{ij} \tag{2.13}$$

.

A equação 2.13 pode ser minimizada através do gradiente estocástico. Para computarmos a função de custo de cada par (i, j), a respeito de um parâmetro θ_k,

temos que [26]:

$$\frac{\partial J_{ij}}{\partial \theta_k} = \frac{-1}{1 + e^{-(s_i - s_j)}} \left(\frac{\partial s_i}{\partial \theta_k} - \frac{\partial s_j}{\partial \theta_k} \right) \tag{2.14}$$

.

Onde $\lambda_{i,j} = \frac{-1}{1+e^{-(s_i - s_j)}}$, e caso $\lambda_{i,j} < 0$, teremos um passo ascendente para s_i e descendente s_j. O tamanho desse passo também é determinado pelo gradiente.

Durante o desenvolvimento do modelo *RankNet*, BURGES [31] descobriu que poderia usar apenas o gradiente (λ), da função de custo em relação ao score. Assim desenvolveu o modelo *LambdaRank*, que também leva em consideração a posição em que os pares se encontram na lista ordenada, priorizando ranquear corretamente amostras próximas ao topo da lista [31]:

$$\lambda_{i,j} = \frac{-|\Delta(i,j)|}{1 + e^{-(s_i - s_j)}} \tag{2.15}$$

.

O hiperparâmetro Δ pode ser otimizado através do métrica de score *NDCG* [31], padrão na literatura de recuperação da informação. Esse direcionamento, apesar de trazer bons resultados em problemas de mecanismos de busca [31], não se aplicam ao caso deste estudo, onde toda a lista possui igual importância em ser ordenada.

Por fim, há o modelo *LambdaMart* que combina o modelo *LambdaRank* com o modelo *MART*, que é um algoritmo de *boosting* semelhante aos demonstrados na seção 2.2.2.1, onde a saída do modelo é uma combinação linear das saídas de um conjunto de árvores de regressão.

$$\lambda_{i,j} = \frac{-\sigma|\Delta Z_{ij}|}{1 + e^{-\sigma(s_i - s_j)}} \tag{2.16}$$

.

Onde a diferença de utilidade gerada trocando as posições de classificação das amostas i e j como $\Delta Z_i j$. Esse modelo em geral é superior aos anteriores em termos de performance e é utilizado na maioria das bibliotecas disponíveis.

Existe uma extensa literatura a respeito desse assunto (*Learning to Rank*) e maiores detalhes sobre esses modelos vão além do escopo deste trabalho, para maiores detalhes recomenda-se a leitura de BURGES [31].

2.2.3 Combinação de Modelos

A técnica de combinar modelos, conhecida como *Ensemble Learning* é bastante utilizada para se obter um resultado superior ao obtido pelos pelos modelos individuais [32]. Essa técnica é bastante similar ao *Bagging*, já apresentado, com a diferença que utiliza-se modelos heterogêneos de qualquer natureza, como *SVM*,

Redes Neurais, *Árvores* e outros. Já no modelo *Random Forest* por exemplo, utilizamos apenas árvores. Contudo em ambos os casos, aproveitamos as vantagens de utilizar um grupo de modelos relativamente fracos e preferencialmente pouco correlacionados.

Os modelos selecionados são treinados separadamente e irão gerar previsões individualmente [32]. Estas previsões podem ser combinadas em uma média, ou um sistema de atribuição de pesos para cada modelo, de forma semelhante a uma regressão linear. Nesse contexto o modelo Ridge apresenta boas propriedades para ser utilizado como um modelo de segunda camada, sendo bastante utilizado para este fim. Contudo, qualquer modelo de aprendizado de máquina pode ser utilizado na segunda camada. Este modelo é treinado a partir das previsões do modelo de primeira camada, gerando uma segunda camada de predições. Evidentemente é possível empilharmos infinitas camadas de modelos, porém o ganho de performance a cada camada tende a zero, podendo até mesmo ser negativo em caso de haver sobreajuste.

2.3 Aprendizado Não Supervisionado

O aprendizado não supervisionado também requer uma matriz de dados estruturada, da mesma forma que o supervisionado. A principal diferença está na ausência da variável alvo. De acordo com EVSUKOFF [1], a falta da variável alvo dificulta a definição de uma função matemática para a avaliação do resultado, sendo o aprendizado não supervisionado uma atividade essencialmente descritiva, onde o próprio modelo é responsável por explicar as estruturas presentes nos dados. O aprendizado não supervisionado se divide em duas áreas principais, redução de dimensionalidade (ver seção 2.4.1) e análise de agrupamento.

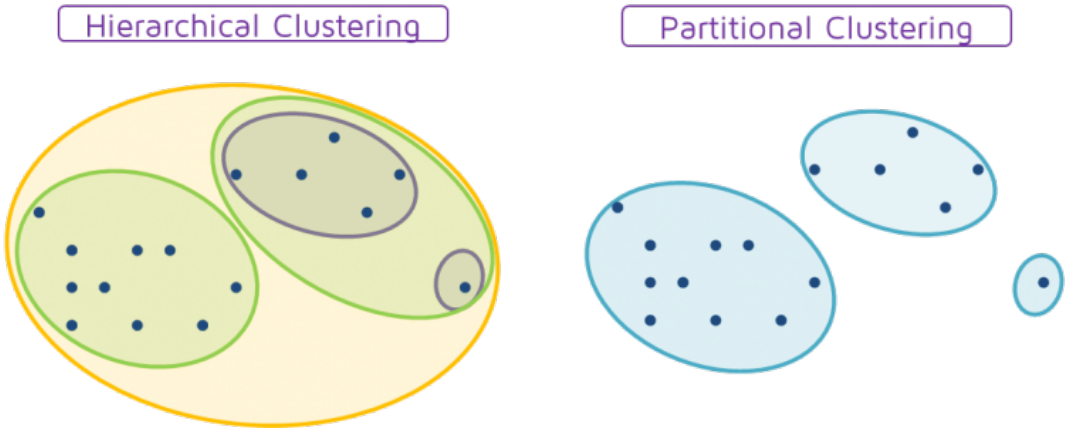


Figura 2.11: Tipos de Análise de Agrupamento (Retirado de QIAN *et al.* [5])

As principais categorias de modelos de agrupamento são divididas em modelos

de partição e hierárquicos (ver figura 2.11). Os métodos de partição buscam diretamente uma partição do conjunto de dados através da minimização de um critério de custo [1]. O algoritmo de partição mais conhecido é o *k-médias* (ver seção 2.4.2).

Os algoritmos de agrupamento hierárquicos geram uma estrutura hierárquica de agrupamentos onde cada nível representa uma partição do conjunto de dados. A estrutura de agrupamentos hierárquicos é representada por um gráfico chamado dendrograma, que representa os diversos níveis de agrupamentos (ver seção 2.4.3).

2.4 Modelos de Aprendizado de Máquina Não Supervisionados

Nesta seção abordaremos alguns dos modelos de aprendizado de máquina não supervisionados que serão utilizados no decorrer deste estudo.

2.4.1 Análise dos Componentes Principais

Análise dos Componentes Principais (PCA) é um método não supervisionado de redução de dimensionalidade que tem como objetivo encontrar a menor combinação linear das variáveis de entrada que explicam a maior parte da variância da amostra [1]. O primeiro passo é calcular a matriz de entrada normalizada (Z), e então calcular os autovalores (Λ) e autovetores (W) tal que $Z'ZW = W\Lambda$ e Λ é uma matriz diagonal com os autovalores ordenados na ordem decrescente W uma matriz ortonormal NxN. Por fim derivamos as variáveis ortogonais $P = ZW$ [12].

As componentes principais são combinações lineares das variáveis originais e número N de componentes principais é escolhido de forma que represente uma parcela grande o suficiente da variância total dos dados originais (geralmente acima de 85%). A grande desvantagem desta técnica é a perda de interpretabilidade da componente resultante [1].

Atente-se ao fato de que é necessário normalizarmos as variáveis antes de aplicarmos o PCA, por duas razões. Primeiro, centralizando os dados podemos assegurar que a primeira componente principal estará corretamente orientada na direção principal das amostras. Segundo, ao normalizarmos os dados, o PCA buscará explicar correlações ao invés das variâncias, caso contrário as primeiras componentes seriam dominadas pelas variáveis de maior variância [12].

2.4.2 Algoritmo K-Médias

O objetivo do algoritmo k-médias é encontrar uma partição no conjunto de dados em K grupos $C = C_1, ..., C_k$ através da minimização do critério de custo quadrático

[1]:

$$J(\Omega) = \sum_{t=1}^{N} \sum_{i=1}^{K} u_i(x(t))||x(t) - \omega_i||^2 \tag{2.17}$$

.

Onde $\omega_i = \frac{1}{N_i} \sum_{x(t) \in C_i}$ é o centroide do grupo C_i, $\Omega = \left[\omega_i^T, ..., \omega_K^T\right]$ representam a matriz dos centros dos grupos e $u_i(x(t))$ é a função indicadora de cara grupo.

2.4.3 Modelos Hierárquicos

Modelos hierárquicos podem ser aglomerativos ou divisivos. Nos algoritmos aglomerativos, todas as amostras são definidas como um agrupamento, o que pode ser visto na figura 2.12, com seis amostras, onde a distância entre esses agrupamentos é zero. Os dois principais hiperparâmetros desse método são a métrica de distância e o critério de ligação.

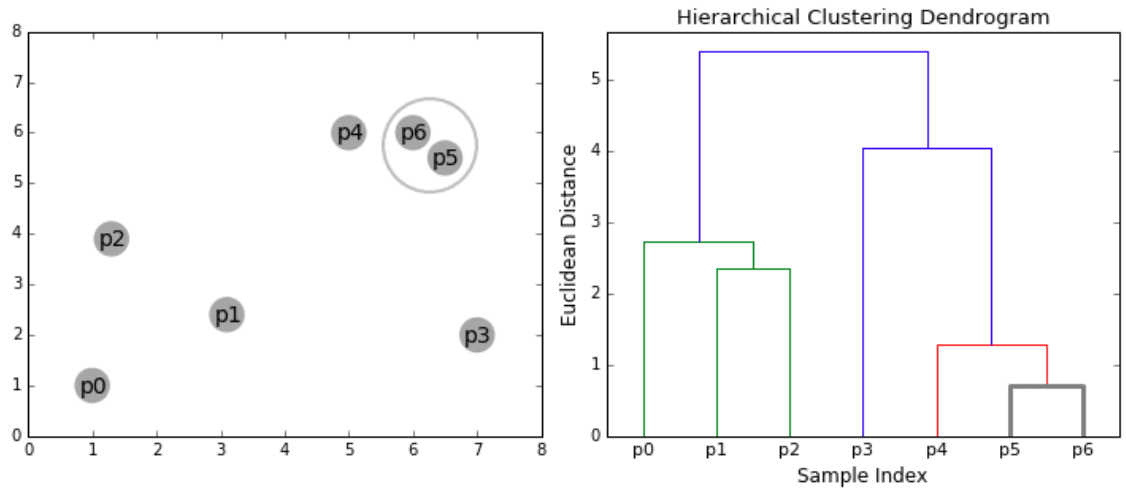


Figura 2.12: Dendograma Agrupamento Hierárquico (Retirado de QIAN *et al.* [5])

Após analisar todas as distâncias entre os 6 pontos (ver equação 2.18 para distância euclidiana), o algoritmo selecionou os pontos 5 e 6, como os mais próximos formando um agrupamento com distância aproximadamente 1 de acordo com eixo vertical.

$$D(i,j) = \sqrt{0.5 \cdot (1 - \rho(i,j))} \tag{2.18}$$

A partir daqui, além da distância entre 2 pontos, precisamos considerar as distâncias entre um ponto e um agrupamento ou entre 2 agrupamentos até que todas as amostras se encontrem em um único agrupamento. Existem diversos critérios de ligação para definir os agrupamentos, podendo ser a distância média entre todos os pontos (*average*), a menor distância entre 2 pontos do agrupamento (*single*), a

maior (*complete*) e finalmente a distância que minimiza o ganho de inércia de cada agrupamento (*ward*) descrito pela equação 2.19.

$$\Delta(A,B) = \sum_{i \in A \cup B} \|\vec{x_i} - m_{\vec{A \cup B}}\|^2 - \sum_{i \in A} \|\vec{x_i} - \vec{m_A}\|^2 - \sum_{i \in B} \|\vec{x_i} - \vec{m_B}\|^2 \tag{2.19}$$

A inércia é simplesmente a soma do quadrado das distâncias da amostras até o centro do agrupamento [1].

$$W_k = \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{2.20}$$

O último passo é determinar o número de agrupamentos, um dos critérios mais utilizados é a métrica *Gap Statistics* introduzida por TIBSHIRANI *et al.* [6], onde K^* é o número otimo de agrupamentos e pode ser obtido pela equação 2.21.

$$K^* = \underset{K}{argmin} \left\{ K | G(K) \geq G(K+1) - S'_{K+1} \right\} \tag{2.21}$$

Onde $G(K)$ representa a diferença entre a média das inércias obtidas com K agrupamentos e S'_K o desvio padrão. Ou seja, K^* é a menor diferença entre a média das inércias $G(K)$ menos a média das inércias de $K+1$ subtraído de um desvio padrão S'_{K+1}. Exatamente como ilustrado na figura 2.13 onde $K^* = 6$.

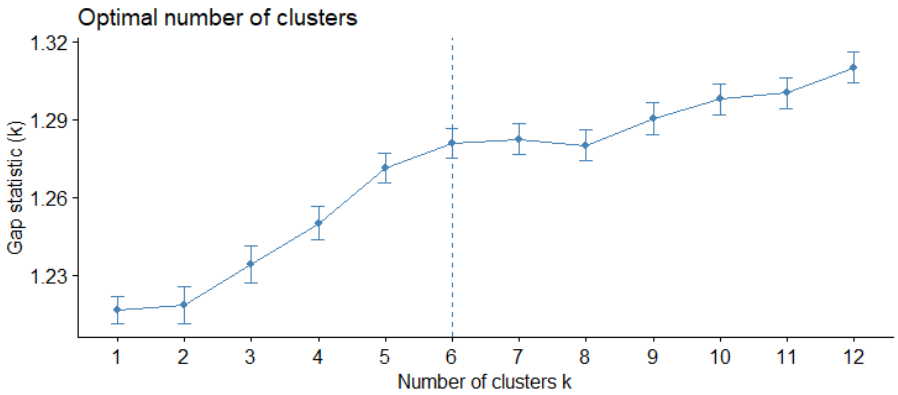


Figura 2.13: Determinação do Número de Agrupamentos (Retirado de TIBSHIRANI *et al.* [6])

2.5 Técnicas de Avaliação de Modelos

Devemos avaliar modelos de aprendizado supervisionado utilizando uma parte do conjunto de dados para treinamento do modelo, e outra para teste. É importante que cada amostra pertença a apenas um desses conjuntos, para evitar que o modelo

seja avaliado em amostras já conhecidas, mitigando o sobreajuste (*overfitting*)[12]. A forma mais popular de se avaliar esse tipo de modelo é através da validação cruzada, onde as amostras são retiradas de diversas formas diferentes formando os dois conjuntos em um processo i.i.d. Nesta seção avaliaremos algumas delas.

2.5.1 K-fold

O k-fold é a maneira mais comum de fazer validação cruzada [7]. Essa técnica dividirá o conjunto de dados em k partes, onde uma será usada para teste e as demais para treino, e assim sucessivamente k vezes até que todos os dados sejam utilizados ao menos uma vez para testes. A figura 2.14 mostra uma divisão com 4 dobras.

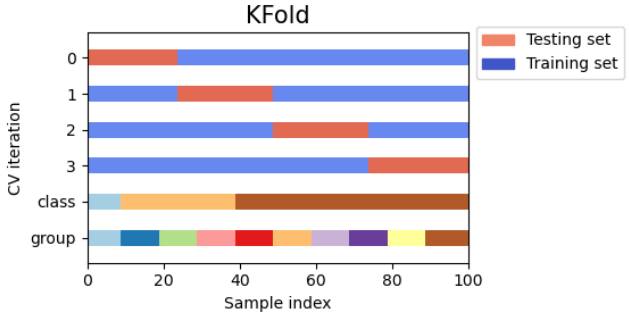


Figura 2.14: Validação Cruzada com K-fold (Retirado de PEDREGOSA *et al.* [7])

Essa estrutura também permite que os dados sejam embaralhados antes de serem divididos.

2.5.2 K-fold em Grupos

O Group k-fold é bastante similar ao k-fold original, com a diferença que passamos uma variável adicional indicando que determinadas amostras pertencem a um grupo específico. No caso do presente estudo, os grupos se referem as **eras**, que são recortes específicos de períodos no tempo.

Imagine o caso onde precisamos separar um quarto dos dados de uma série temporal mensal para testes, selecionando diferentes espaços de tempo, sem quebrar a estrutura temporal da série. Esse caso é descrido na figura 2.15, onde os grupos são selecionados de maneira aleatória ao longo da série.

Para o K-fold em blocos, a diferença é que um bloco de grupos contíguos é selecionado. Por exemplo, para uma série de preços com 30 meses, faremos o teste em cada bloco de 3 meses, e treinaremos nas demais eras, nesse caso configura-se um Block 10-fold.

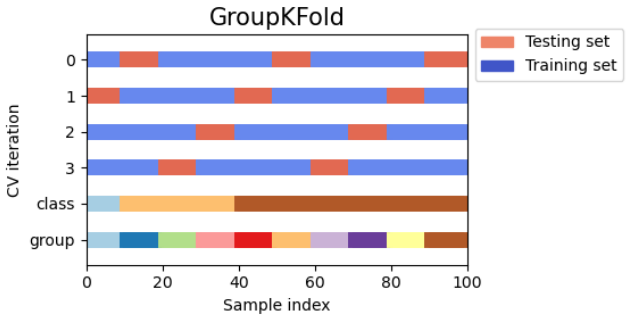


Figura 2.15: Validação Cruzada com K-fold em Grupos (Retirado de PEDREGOSA *et al.* [7])

2.5.3 K-fold para Séries Temporais

O k-fold usado em séries temporais, é um pouco diferente dos anteriores. Nessa técnica é necessário respeitar a sequência temporal dos dados. Como visto na figura 2.16, é feito uma seleção de um momento no tempo onde os dados anteriores serão usados para treino e os posteriores para teste [33].

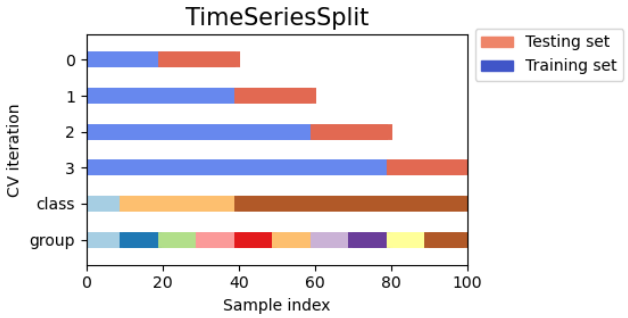


Figura 2.16: Validação Cruzada com K-fold para Séries Temporais (Retirado de PEDREGOSA *et al.* [7])

É importante observar que dessa forma, diferente de todas as anteriores, o tamanho do conjunto de treino varia e nem todos os dados podem ser usados para teste, então é necessário ter os devidos cuidados ao compararmos os resultados dessa técnica, com outras.

2.5.4 K-fold Aninhado

A validação cruzada aninhada *Nested k-fold* é uma abordagem para a otimização de hiperparâmetros que é mais robusta ao sobreajuste (*overfitting*). O validação k-fold reduz esse efeito, mas não o remove completamente [34].

O procedimento envolve tratar a otimização do hiperparâmetro do modelo como parte do próprio modelo e avaliá-lo dentro do procedimento de validação cruzada k-fold mais amplo para avaliar e comparar modelos. De tal forma, o k-fold é perfor-

mado em conjunto da otimização de hiperparâmetros, mas está aninhado dentro de outro procedimento de validação cruzada como mostra a figura 2.17.

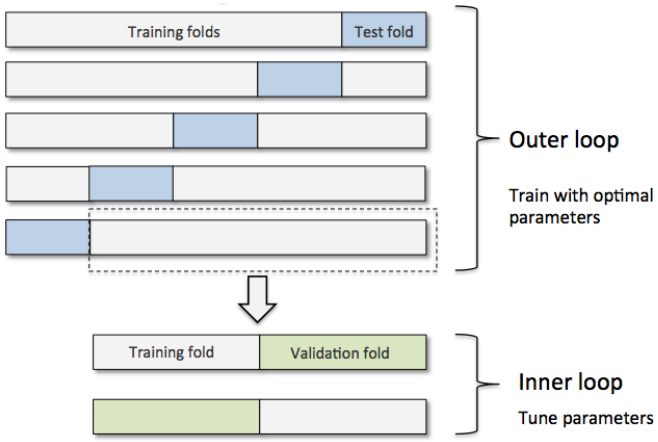


Figura 2.17: Validação Cruzada com K-fold Aninhado (Retirado de RASCHKA [8])

Sob este procedimento, a busca de hiperparâmetros é exposta apenas a um subconjunto do conjunto de dados fornecido pelo procedimento de validação cruzada externa. Isso reduz, senão elimina, o risco de sobreajuste e deve fornecer uma estimativa menos tendenciosa do desempenho de um modelo ajustado no conjunto de dados [34].

A grande desvantagem deste procedimento é o aumento exponencial de iterações necessárias para se concluir o treinamento. Por essa razão este procedimento só será utilizado em modelos combinados, apresentados na seção 2.2.3.

Em modelos de *stacking* usaremos o ciclo de validação cruzada interno para treinaremos os modelos de primeiro nível nos dados originais e faremos as previsões. No externo, treinaremos o modelo usando as previsões do primeiro passo como features.

2.6 Otimização de Hiperparâmetros

No aprendizado de máquina, diferentes modelos são testados e os hiperparâmetros são ajustados para obter melhores previsões. Hiperparâmetros são parâmetros que necessitam ser fornecidos ao modelos antes do treinamento, enquanto os parâmetros são valores que são aprendidos durante o treinamento [9].

Os hiperparâmetros são ajustados escolhendo os valores de parâmetro ideais de forma a otimizar uma métrica de usada para comparação entre os modelos. Este processo pode ser demorado, mas existem algumas técnica que podem melhorar esse procedimento.

2.6.1 Busca Exaustiva e Aleatória

Dois dos métodos de otimização de hiperparâmetros são a busca exaustiva (*grid search*) e a busca aleatória (*random search*), em ambos os casos, um conjunto de valores referentes a cada hiperparâmetro é passado para o modelo, e em seguida o modelo é treinado usando uma dessas combinações e avaliado de acordo com uma métrica de score definida.

A limitação fundamental da busca exaustiva, é que ela testa todas as combinações possíveis. Além do tempo computacional que essa operação consome, é necessário passar distribuições discretas de cada hiperparâmetro para possibilitar que o número de combinações seja finito.

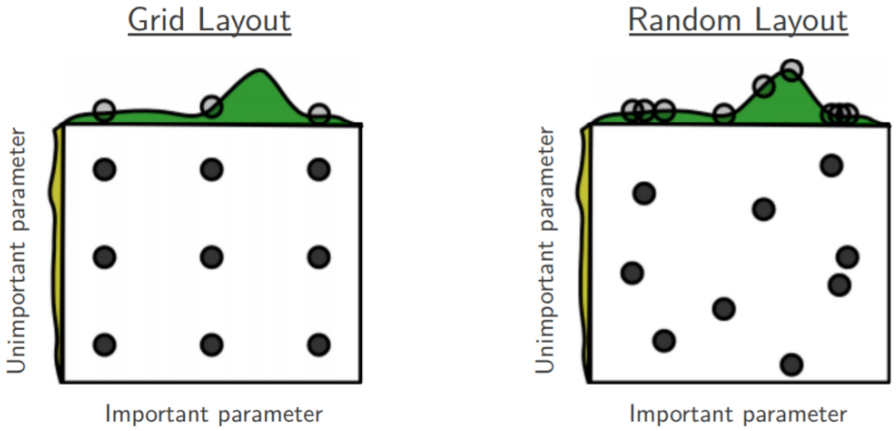


Figura 2.18: Comparativo Busca Exaustiva e Aleatória (Retirado de BERGSTRA e BENGIO [9])

A busca aleatória, além de permitir distribuições contínuos, é possível definir um número máximo de tentativas. A figura 2.18, mostra que de maneira geral a busca aleatória retorna valores melhores em relação a performance do modelo [9].

2.6.2 Busca Bayesiana

Observando novamente a figura 2.18, podemos interpretar o subespaço gerado pela combinação dos hiperparâmetros de um modelo de aprendizado de máquina, como um novo problema de otimização. Nesse contexto, utilizaremos uma abordagem probabilística, chamada Otimização Bayesiana. De acordo com SNOEK *et al.* [35] a Otimização Bayesiana é uma abordagem que utiliza o Teorema de Bayes para direcionar a busca de valores extremos de uma função objetivo. A principal vantagem desse método, é a sua capacidade de aprender a partir da amostragem do espaço, de forma que as amostras futuras sejam direcionadas às partes do espaço de busca que têm maior probabilidade de conter valores extremos.

A Otimização funciona construindo um modelo probabilístico da função objetivo, chamado de função substituta (*surrogate function*), que é uma aproximação da função objetivo, então uma função de aquisição selecionará pontos ótimos locais para serem testados na função objetivo, que nesse caso seria o próprio modelo de aprendizado de máquina retornando uma métrica pré-determinada.

Em termos probabilísticos, a função substituta é a probabilidade condicional $P(f|D)$ da função objetivo (f), dado as amostras (D) ou mais formalmente:

$$P(f|D) = P(D|f)P(f) \tag{2.22}$$

Onde a função substituta é a probabilidade a posteriori, a função objetivo é a probabilidade a priori e $P(D|f)$ a função de aquisição, que muda a cada vez que mais observações são coletadas. De acordo com FRAZIER [10], a busca de hiperparâmetros com Otimização Bayesiana pode ser resumida em quatro passos:

1. Selecionar uma amostra ao otimizar a função de aquisição.
2. Avaliar a amostra com a partir da função objetivo
3. Atualizar a função substituta
4. Retornar ao passo 1

Função Substituta

A forma mais comum de estimar a função substituta é tratando o problema como um problema de regressão com o subespaço hiperparâmetros representando a entrada e a saída é dada por uma métrica já definida (ex: erro médio quadrático). Para essa tarefa FRAZIER [10] recomenda o uso de *Random Forest* ou *Gaussian Process*.

Um Processo Gaussiano é um modelo que constrói uma distribuição de probabilidade conjunta sobre as variáveis, assumindo uma distribuição gaussiana multivariada. Como tal, é capaz de sumarizar de forma eficiente e eficaz um grande número de funções alterando à medida que mais observações são disponibilizadas para o modelo, como descrito no quadro superior da figura 2.19 (para mais detalhes [10]).

Função de Aquisição

Retornando a equação 2.19, a função de aquisição ($P(D|f)$) é responsável por indicar o próximo candidato a ser avaliado pela função objetivo. Observe no quadro inferior da figura 2.19, onde o candidato que apresenta maior melhoria esperada $EI(x)$(*Expected Improvement*) é selecionado para ser avaliado pela função objetivo,

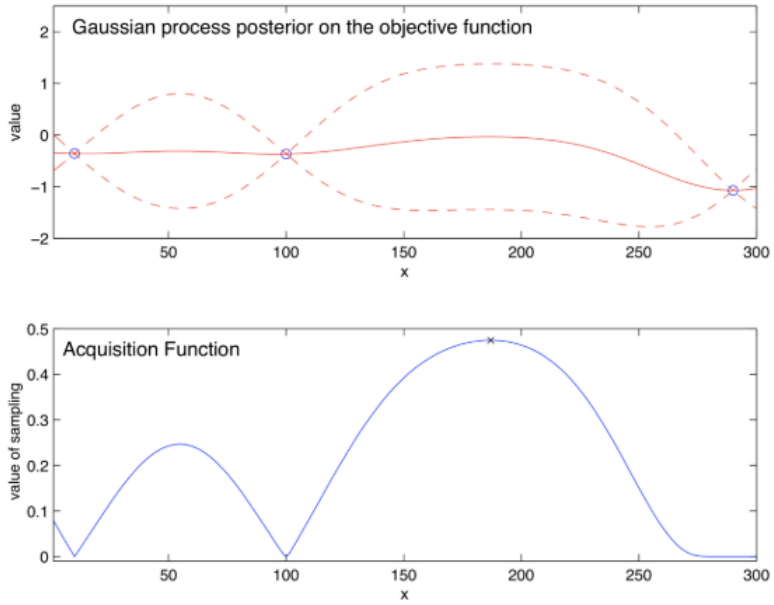


Figura 2.19: Processo Gaussiano (Retirado de FRAZIER [10])

ou na prática, o candidato é combinação de hiperparâmetros utilizado no treinamento do modelo de aprendizado de máquina. Definimos a melhoria esperada por:

$$EI_n(x) := E_n[[f(x) - f_n^*]^+] \tag{2.23}$$

A função de aquisição por melhoria esperada usa uma variedade de abordagens. Ao contrário da função objetivo (f) em nosso problema de otimização original, $EI_n(x)$ é barato de avaliar e permite fácil avaliação de derivadas de primeira e segunda ordem [10]. Por exemplo, uma técnica que funcionou bem para o autor é calcular as primeiras derivadas e usar o método quasi-Newton L-BFGS. A expressão fechada para o cálculo de $EI_n(x)$ pode ser encontrada em FRAZIER [10].

Capítulo 3

Dados do Problema

Ao longo deste capítulo será estruturado um problema de arbitragem estatística como um problema de aprendizado supervisionado a partir de dados públicos, em seguida vamos apresentar um torneio que fornece um conjunto de dados de boa qualidade para o mesmo problema e apresentar qual a proposta do mesmo além de detalhar esse novo conjunto de dados.

3.1 Estruturando o Problema

Nesta seção será estruturado um exemplo básico de aprendizado supervisionado para o problema de arbitragem estatística. Para tal, serão utilizados os títulos que compõem o índice Russell 3000[1]. Que representam as 3000 maiores ações negociadas no mercado norte-americano, além de uma cesta de aproximadamente 2000 ações das principais empresas de capital aberto negociadas pelo mundo.

Tabela 3.1: Lista de Ações Monitoradas

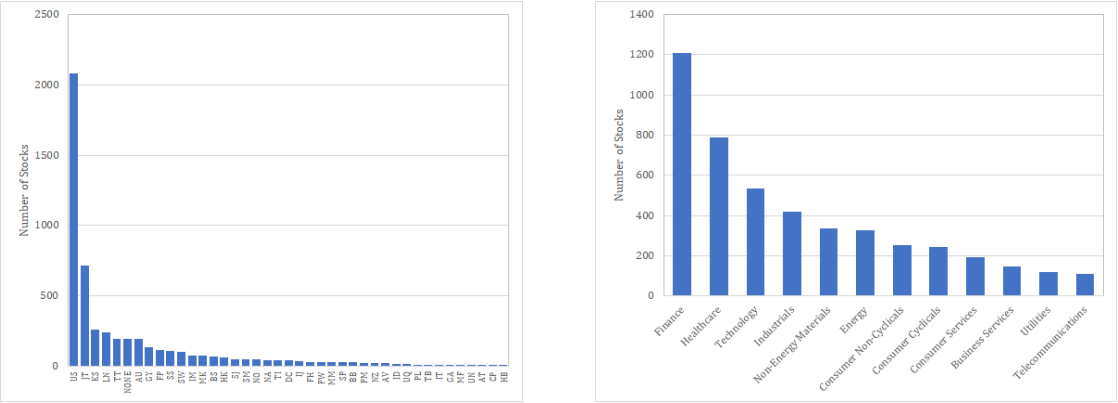
#	Ticket
0	316140 KS
1	000060 KS
2	000080 KS
...	...
4830	CG US
4831	GO US
4832	IMUX US

As ações da tabela 3.1 foram nomeadas de acordo com o ticket da bloomberg[2], onde o padrão é a sigla referente a ação seguida pela abreviação do país. Na figura

[1]Russell 3000. https://www.investopedia.com/terms/r/russell_3000.asp (Acessado em 25/10/2020)

[2]Bloomberg. https://www.bloomberg.com/markets/stocks (Acessado em 25/10/2020)

3.1, nota-se que a maioria das ações são dos Estados Unidos (US), seguidos por mercados asiáticos como Japão (JT) e Coreia do Sul (KS), seguido por países como Austrália (AU) e Alemanha (GY) e outros com uma pequena parcela[3]. Ainda na figura 3.1, sobre as indústrias, temos um predomínio de ações dos ramos financeiro, saúde e tecnologia, como é característico em bolsas de valores pelo mundo.



(a) Por País (b) Por indústria

Figura 3.1: Ações por País e Indústria (Adaptados de Numerai[3])

Em seguida, na tabela 3.2 agregamos o preço de fechamento diário do preço de cada uma dessas ações de dezembro de 2002 até a data presente (Dez-2020), usando um provedor de dados financeiros gratuito, resultando em uma série de aproximadamente 16 milhões de preços em suas respectivas moedas locais.

Tabela 3.2: Série de Preços

Data	Ticket	Preço
2002-12-02	1723 NA	20.359694
2002-12-02	2352 NA	32667.400391
2002-12-02	139480 KS	14.597309
2002-12-02	1316 HK	1.459269
2002-12-02	1605 JP	1.101638

Com a série de preços em mãos, é necessário incluir variáveis ao conjunto de dados que auxiliem a prever o comportamento futuro dessas ações. Nesse exemplo será utilizado o indicador de análise técnica, Índice de força relativa (RSI), que de maneira resumida, mede se o ativo alvo está excessivamente comprado, para valores acima de 70, ou excessivamente vendido, para valores abaixo de 30 (para mais detalhes consultar WILDER [36]). A formulação do RSI se dá na equação 3.1:

$$RSI = 100 - (100/(1 + RS)) \tag{3.1}$$

[3]Documetação Numerai https://www.numer.ai (Acessado em 22/01/2020)

RS = Média dos retornos nos períodos de alta / Média dos retornos nos períodos de baixa

O RSI é calculado levando em consideração uma janela de 14 dias, podendo utilizar outros intervalos. Em seguida, os valores de RSI obtidos serão distribuídos na forma de um histograma com cinco divisões (quintil), com valores inteiros entre [0..4], agora os dados estão como se segue na tabela 3.3.

Tabela 3.3: Série de Preços com RSI

Data	Ticket	Preço	RSI	Qtil.
2002-12-02	1723 NA	18.581551	18.748923	0
2002-12-02	2352 NA	33711.300781	70.729613	4
2002-12-02	139480 KS	14.736325	58.823245	4
2002-12-02	1316 HK	1.445166	45.453015	3
2002-12-02	1605 JP	1.038018	36.231858	2

Uma forma de adicionar mais informação em cada registro, é incluir uma série de combinações da variável quintil ao conjunto de dados, como o quintil dos dias anteriores (*lag*), a fim de aumentar o poder preditivo dos modelos, como mostra a tabela 3.4.

Tabela 3.4: Série de RSI com Atraso

Ticket	Qtil.	lag0	lag1	lag2	lag3	lag4	lag5
MNKD US	2	2	2	2	1	2	3
TXT US	3	3	3	2	1	1	0
SAF FP	2	2	2	3	3	3	3
CAST SS	0	0	0	0	0	0	0
TDG US	0	0	3	3	4	4	4

Observe que o indicador varia pouco de um dia para o outro, mesmo assim, conseguimos ver alguma variação ao longo da série de 6 dias. Então, possível levar essas combinações a exaustão e adicionar outras colunas que mostrem a diferença entre cada dia e outros, chegando facilmente a algumas dezenas de combinações de colunas (*features*).

3.1.1 A Variável Alvo

O problema de aprendizado supervisionado depende de uma variável alvo (*target*), que remete ao objetivo do problema, que é prever o retorno de uma cesta de ativos para um determinado período.

Nesse caso, deve-se considerar o retorno relativo de cada ação em um período de 20 dias úteis, que será chamada de **era**. Por padrão cada era começará as sextas-

feiras, compreendendo toda a série (2003-2020). Por fim os retornos também serão distribuídos em um histograma com 5 divisões, isto é (0,0.25,0.5,0.75,1), como visto na tabela 3.5.

Tabela 3.5: Série Com a Variável Alvo

Ticket	Data (sexta)	Returno (%)	Target
000270 KS	20030131	0.002	0.5
000810 KS	20030131	0.001	0.5
000830 KS	20030131	-0.001	0.5
002790 KS	20030131	-0.009	0.25
003450 KS	20030131	-0.011	0.25

Perceba que agora há uma série com aproximadamente 16 milhões de preços, para a cesta de 5000 ações e dividas em 851 eras individuais de 20 dias úteis, os outros 4 dias da semana agora podem ser descartados. Note que há uma sobreposição entre as datas de início e fim de cada era, configurando um vazamento de dados (*data leakage*), que será tratado posteriormente.

Finalmente, ao concatenar a tabela 3.3, com a tabela 3.5, o problema preditivo estará configurado, com uma série de preços para cada ação seguido de um conjunto de indicadores financeiros e algumas transformações desses indicadores como mostrado na tabela 3.4 e a última coluna será a variável alvo (target) da era subsequente, que é o valor precisamos prever, a tabela 3.6 ilustra um exemplo desse resultado. Nesse formato é possível treinar um modelo de aprendizado supervisionado para gerar predições.

3.1.2 Treinando um Modelo

Nessa seção será ajustado um modelo de árvore de decisão (regressão), como explicado na seção 2.2.2.1. Utilizou-se as eras entre 2003 e 2013 para treino e o restante para teste. Perceba na figura 3.2 que o modelo atribuiu diferentes importâncias para cada variável. Lembrando que seleção de variáveis é um dos principais temas a serem explorados ao longo deste trabalho.

A figura 3.3a remete as predições do modelo nos dados de teste, onde cada ticket em cada era, possui um valor atribuído entre 0 e 1, por se tratar de regressão. Na figura 3.3b, há a distribuição dos valores preditos pelo modelo. Note que a grande maioria se concentra em torno de 0.5, que é esperado, pois indica que a princípio, o modelo não encontrou uma oportunidade de arbitragem, as oportunidades em geral se encontram nas caudas da distribuição.

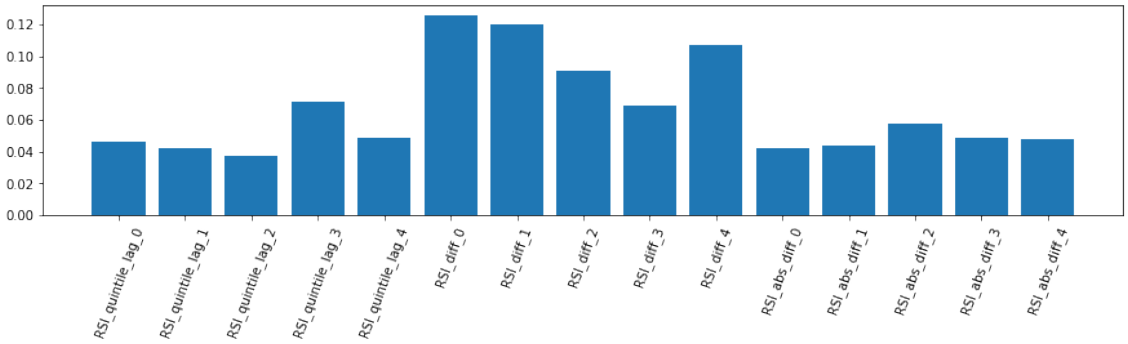
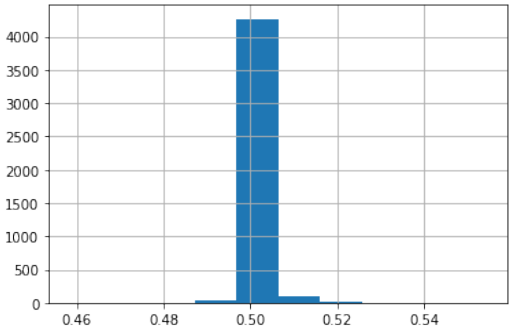


Figura 3.2: Importância por Variável

bloomberg_ticker	signal
AAPL US	0.5488135039273250
TSLA US	0.7151893663724200
EMG LN	0.6027633760716440
2638 HK	0.5448831829968970
MOH GA	0.4236547993389050

An example stock market signal

(a) Predições por Ticket



(b) Distribuição das Predições

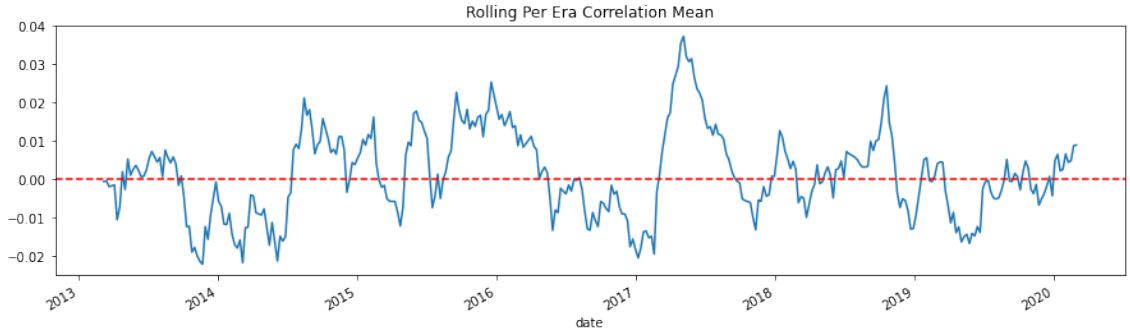
Figura 3.3: Predições do Modelo (Adaptados de Numerai[3])

3.1.3 Avaliando os Resultados

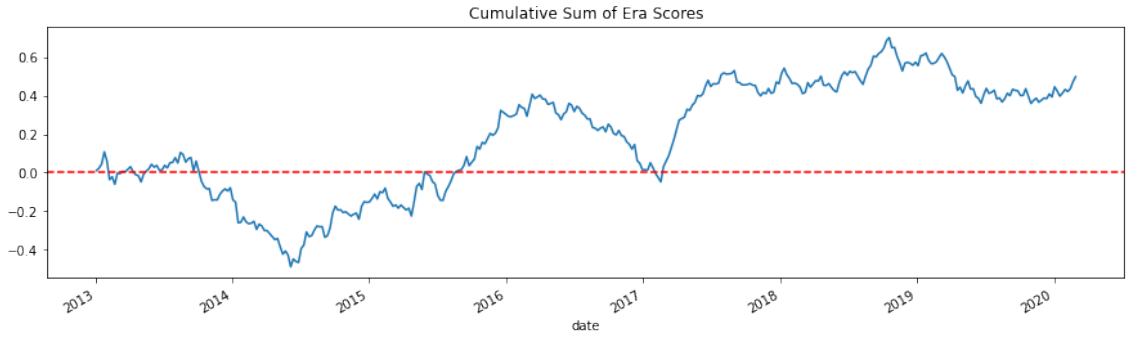
É possível simplificar o problema original, apenas comparando o ranking relativo das predições do modelo com a variável alvo. As melhores ações em uma determinada era devem receber um valor próximo a 1, e as piores 0, da mesma forma que o target. Pode-se obter essa relação através da correlação de spearman, que sera melhor detalhada na seção 4.3.1. Dessa forma o problema preditivo se transforma em um ranqueamento de ações **por era**.

A figura 3.4a ilustra os resultados obtidos pelo modelo. A correlação de spearman (ρ) obtida no período foi de 0.0014, sendo que 51.64% das eras obtiveram correlação positiva ($\rho > 0$). A princípio, pode-se concluir que é um valor extremamente baixo. Contudo analisando o resultado acumulado, é possível observar um bom resultado no longo prazo, como ilustra a figura 3.4b.

Porém, é importante salientar que esse é um resultado preliminar que desconsidera custos de transação, falta de liquidez dos ativos e até mesmo uma grande perda acumulada logo no início da série histórica que levariam a resultados práticos bem diferentes do ilustrado. Todavia, esse resultado foi obtido através de um conjunto



(a) Correlação Média 10 Eras



(b) Acumulado por Era

Figura 3.4: Resultados Base de Teste

de dados extremamente simples e com um modelo de aprendizado de máquina sem nenhuma otimização.

3.1.4 Aprimorando os Resultados

Certamente a adição de mais indicadores financeiros das empresas, melhoraria a performance do modelo, como:

- *Análise Técnica*: MACD, EWMA, RSI
- *Análise Fundamentalista*: P/E ratio, dividend yield, rating de agências
- *Indicadores mistos*: Barra Risk, Fama French
- *Dados Alternativos*: Transações de cartões, imagens de satélite, análise de rede social

Contudo conseguir dados de qualidade de maneira consistente, de forma a constituir uma vantagem competitiva no mercado é extremamente difícil e caro [22]. Dessa forma o *Numerai Tournament*, configurado exatamente no modelo descrito por DE PRADO e FABOZZI [22] se mostra bastante vantajoso para cientistas de dados interessados em desenvolver modelos de aprendizado de máquina para esse fim.

3.1.5 A Numerai

A Numerai nada mais é que um gestora de um fundo de risco neutro, como a Renaissance Technologies ou a AQR Capital Management citados no capítulo 1. A principal diferença é que ao invés de manter um time de pesquisa que constrói os modelos quantitativos, a Numerai compra dados de alta qualidade de fornecedores institucionais privados, o disponibiliza para a comunidade de maneira gratuita, porém encriptada e por fim mantém um time de operações que utilizará o sinal gerado pelos modelos.

A figura 3.5 ilustra a performance do fundo gerido pela Numerai em comparação a outros fundos de mesmo tipo no mercado. Apesar de 2020 se tratar de um ano bastante atípico em decorrência da COVID-19, é interessante notar que o fundo possui uma performance bastante descorrelacionada dos demais fundos do mercado. Acredita-se que isso é possível devido a enorme diversidade de modelos que são criados pela comunidade e enviados para Numerai de maneira recorrente. Dessa forma o gestor do fundo consegue identificar oportunidades de investimos que não são vistas pelos concorrentes.

2020 Quant Fund Returns	
Numerai	**+7.68%**
Aurum Quant Equity Market Neutral Hedge Fund Index [1]	-17.35%
Renaissance's RIDA [2]	-32%
Renaissance's RIDGE [3]	-31%
AQR Global Stock Selection [4]	-22%
Winton Fund [5]	-20%
Two Sigma Absolute Return [6]	-5%

Figura 3.5: Performance Fundo Numerai[4]

Na página do fundo[4] podemos encontrar informações sobre como aplicar no fundo diretamente, apesar de este ser declaradamente um produto financeiro mais adequado para investidores institucionais.

3.2 O Torneio

O artigo de DE PRADO e FABOZZI [22] se refere explicitamente ao torneio da Numerai[5]. Que se intitula como o torneio mais difícil do mundo onde os participantes estão ativamente construindo um fundo de investimentos e o organizador declara já ter pago mais de 85 milhões de dólares aos participantes.

[4]Performance Fundo Numerai. https://www.numer.ai/fund (Acessado em 11/11/2021)
[5]Numerai. https://www.numer.ai/ (Acessado em 11/11/2021)

O objetivo do torneio é informações para criar estratégias de investimento para o fundo. A competição é configurada exatamente nos moldes descritos na seção 1.2, usando os dados encriptados disponibilizados pela equipe da numerai, e as predições são feitas na mesma cesta de ações, porém de forma anonimizada.

Ao longo dessa serão explorados os dados disponibilizados pela Numerai. É declarado que este é um conjunto de dados limpo e de alta qualidade, porém encriptado. **Também é importante salientar que não há nenhum problema com o fato das linhas e colunas estarem encriptadas**, além da confidencialidade, também ajuda a evitar que o participante incorra em algum tipo de viés de seleção ao analisar os dados [12]. É notório que gestores de fundos quantitativos profissionais operam exatamente dessa forma, para coibir decisões de investimento discricionárias [23] e a encriptação dos rótulos das colunas e do ticket das ações de maneira nenhuma atrapalha o os modelos utilizados aqui.

3.2.1 Dados de Treinamento

A coluna ID na tabela 3.6 é referente ao ticket da ação, que aqui se encontra encriptado. O conjunto de dados de treinamento é identificado pela coluna *tipo*, ainda há o identificador de cada era, os indicadores da ação e o target, como vimos anteriormente, totalizando 314 colunas e aproximadamente 501 mil registros.

Tabela 3.6: Dados Treinamento

id	era	tipo	FI12	FW46	FCh86	FDx14	...	target
n000315175b67977	1	train	0	0.5	0.25	0	...	0.25
n0014af834a96cdd	1	train	0	0	0	0.25	...	0
n001c93979ac41d4	1	train	0.25	0.5	0.25	0.25	...	0.75
n0034e4143f22a13	1	train	1	0	0	0.5	...	0.25
n00679d1a636062f	1	train	0.25	0.25	0.25	0.25	...	1

Em relação as eras, a única diferença para o problema estruturado anteriormente é que não há a sobreposição. No conjunto de dados de treino anterior haviam 10 anos de dados (2003-2012) onde toda sexta-feira marcava o início de uma era de 20 dias. Nesse caso para evitar a sobreposição de eras 3/4 das eras foram descartadas, ou seja a era 1 remete a primeira semana, e a era 2 agora se refere a semana 5 e assim subsequentemente até a era 120.

3.2.1.1 Variáveis dos Dados

Totalizam 310 variáveis selecionadas e distribuídas na forma de quintil como visto na seção 3.1. O nome das colunas está ofuscado com nomes como "feature_-intelligence12", abreviado para FI12 na tabela 3.6 para fins de enquadramento.

O nome das colunas está dividido em 6 grupos como "feature_wisdom"e "feature_dexterity", como visto na tabela 3.7 onde cada um desses grupos pode ser perfeitamente entendido como indicadores de análise técnica, análise fundamentalista, indicadores macroeconômicos relacionados ao setor ou país.

Tabela 3.7: Grupos de Variáveis

Grupo	Qtd.
Intelligence	12
Wisdom	46
Charisma	86
Dexterity	14
Strength	38
Constitution	114

Note na figura 3.6 que os pares de variáveis mais correlacionadas são contíguas, como 2 e 3, indicando que podem pertencer a um mesmo indicador mas uma delas pertence a um período anterior (*lag*). Contudo a correlação entre essas variáveis pode ser inconsistente. A tabela 3.8 compara a correlação entre pares de variáveis nas primeiras 60 eras, depois nas 60 eras seguintes e por fim calcula a diferença entre as mesmas.

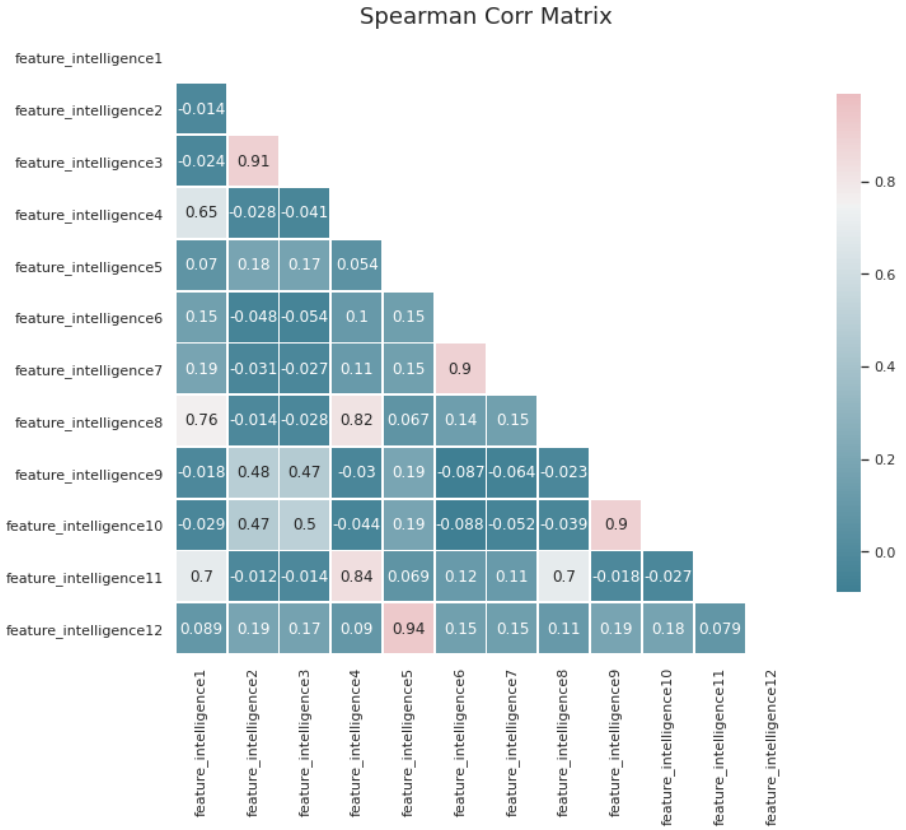


Figura 3.6: Correlação das Variáveis do Grupo Inteligência

Tabela 3.8: Diferença de Correlação por Era

Variável 1	Variável 2	Eras 1-60	Eras 61-120	Diferença
feature_intelligence11	feature_intelligence9	0.0914	-0.1289	-0.2202
feature_intelligence10	feature_intelligence11	0.075	-0.1305	-0.2055
feature_intelligence10	feature_intelligence4	0.0548	-0.1442	-0.199
feature_intelligence4	feature_intelligence9	0.0676	-0.1296	-0.1972
feature_intelligence10	feature_intelligence8	0.0549	-0.1354	-0.1903

3.2.1.2 A Variável Alvo

A variável alvo do conjunto de dados de treinamento é muito similar a apresentada na seção 3.1. Refere-se ao retorno obtido pela ação ao final da era, e então distribuída forma de um quintil (0, 0.25, 0.5, 0.75, 1). A principal diferença é que dentro de cada era, cada classe extrema possui 5% dos dados, as classes intermediárias 20% cada e a classe do meio 50%. A figura 3.7 ilustra a distribuição das classes ao longo das 120 eras. Note também que a quantidade de ações dentro de cada era varia, começando próximo de 2500 na era 1 e chegando a 4500 na era 120.

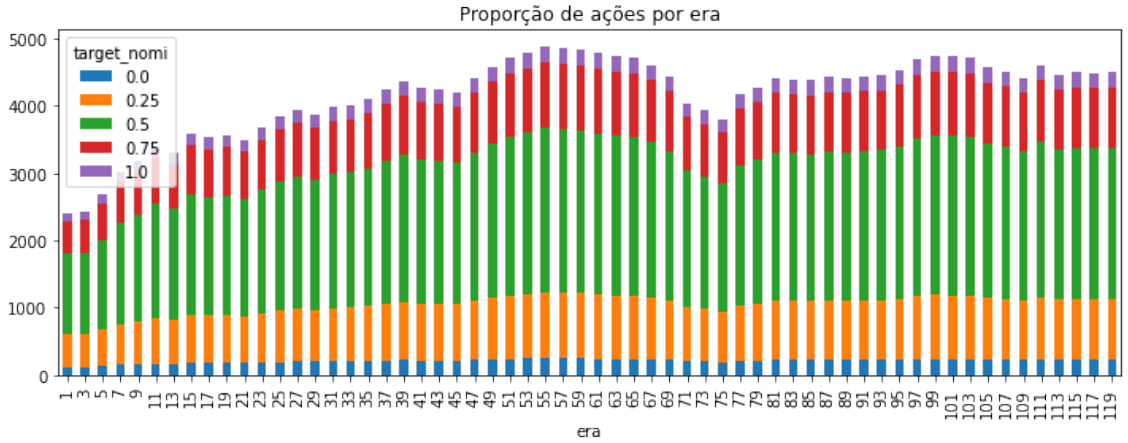


Figura 3.7: Variável Alvo no Conjunto de Treino

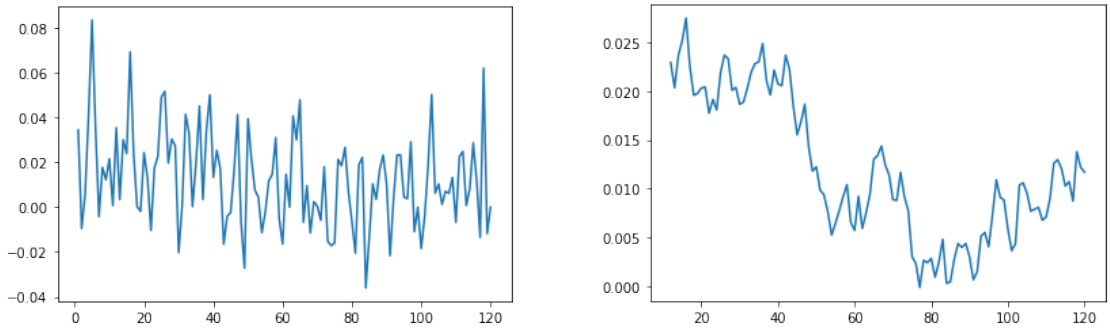
Assim como exposto na seção 3.1, onde havia apenas o indicador RSI e derivados, a correlação dos demais indicadores com o alvo é baixa (ver tabela 3.9), contudo ainda é possível extrair valor dessas variáveis.

Assim como foi observado na tabela 3.8, a correlação de uma variável é inconsistente. A figura 3.8a mostra a correlação da variável *strenght34* ao longo do tempo e a figura 3.8b com uma média móvel de 10 eras.

Lembrando que essas variáveis são indicadores financeiros, dessa forma é possível que se obtenha algum tipo de ganho financeiro observando apenas para uma variável, como mostra a figura 3.8b. Porém essa vantagem possui tempo limitado, podendo levar a perdas por longos períodos subsequentes.

Tabela 3.9: Correlação das Variáveis com o Alvo

Variável	Correlação
feature_strength34	0.0123
feature_strength14	0.0115
feature_charisma37	0.0110
...	...
feature_dexterity4	-0.0110
feature_dexterity6	-0.0111
feature_dexterity7	-0.0115



(a) Por Era

(b) Média Móvel 10 eras

Figura 3.8: Correlação de Uma Variável com o Alvo

3.2.1.3 Interpretando as Eras

Da mesma forma que na seção 3.1, as eras compreendem 20 dias de negociação. A razão pelos 20 dias não é divulgada, mas é sabido que um intervalo de predição mais curto, apesar de parecer mais fácil, também é mais sujeito a flutuações causadas por ruído do mercado.

Pode-se pensar em grupos de eras como regimes, ou seja, recortes no tempo onde o mercado tende a seguir diferentes direções. Grande parte do ano de 2020, período que compreende a COVID-19, representa uma ou mais mudanças de regime, tratando-se de um recorte temporal muito valioso para a avaliação de modelos. Esse período e alguns outros foram separados em um conjunto de dados de validação que serão apresentados na seção 3.2.2.

Para ilustrar as mudanças de regime, treinou-se um modelo de regressão linear em cada bloco de 10 eras (aproximadamente 10 meses) e o testou-se em cada um dos outros blocos, como mostra a figura 3.9.

Pode-se concluir que as primeiras eras são bastante aplicáveis entre si e também as eras entre 70 e 100. As eras do meio parecem descrever um regime mais complicado do comportamento dos mercados. De forma geral, seguindo a diagonal principal a direita, vemos a performance dos modelos decair com o decorrer do tempo.

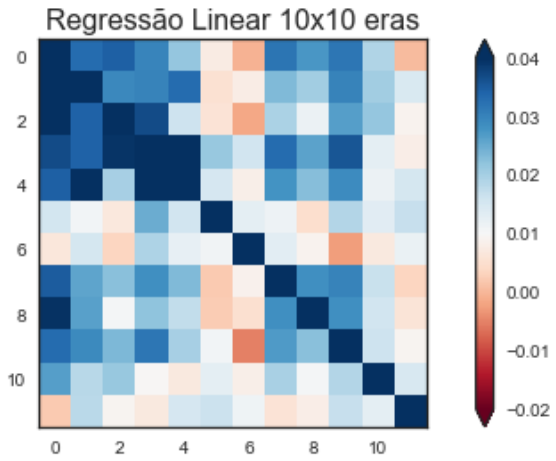


Figura 3.9: Regressão Linear 10 Eras

3.2.2 Dados de Validação

O conjunto de dados de validação é muito semelhante aos dados de treino (tabela 3.6), mudando apenas o valor da coluna "tipo", e as eras, que na presente data remetem as eras 121-132 e 197-206, sendo que a cada 6 meses um novo conjunto é incorporado, aumentando a quantidade de eras disponíveis. O conjunto de validação se encontra na tabela 3.10.

Tabela 3.10: Conjunto de Dados de Validação

id	era	tipo	FI1	FI2	FI3	FI4	...	target
n0003aa52cab36c2	121	validation	0.25	0.5	0.25	0	...	0.25
n000920ed083903f	121	validation	0.75	0	1	0.25	...	0
n0038e640522c4a6	121	validation	0.25	0.5	0.25	0.25	...	0.75
n004ac94a87dc54b	121	validation	1	0	0	0.5	...	0.25
n0052fe97ea0c05f	121	validation	0.25	0.25	0.25	0.25	...	0

O objetivo desse conjunto de dados é permitir testes em dados desconhecidos pelo modelo, contudo esta base é relativamente pequena e por se tratar de um problema não estacionário provavelmente não será representativa do presente. Dessa forma, essa base não será utilizada para comparação de modelos, mas será bastante utilizada ao longo do trabalho para ilustração de conceitos.

3.2.3 Dados de Teste

Além do conjunto de validação, há o conjunto de teste, composto de aproximadamente 850 eras *com sobreposição*, essa base é utilizada apenas para a Numerai avaliar se as predições deste modelo estão aptas a ser utilizadas em seu fundo de investimento. Nesse conjunto há também a *"live era"*, que é referente a semana

atual e é onde as previsões são remuneradas. É importante salientar que em todo o conjunto de teste, o valor referente a variável alvo não está disponível.

Tabela 3.11: Conjunto de Dados de Teste

id	era	tipo	FI1	FI2	FI3	FI4	...	target
nffd239464395ebf	eraX	live	0	0.5	0.25	0	...	NaN
nffd89078dbed0c8	eraX	live	0.5	0	1	0.25	...	NaN
nffe7b410a2e9866	eraX	live	0.25	1	0.75	0.25	...	NaN
nffef1ef7ea87c0b	eraX	live	1	1	0	0.5	...	NaN
nfffd2286c1361bd	eraX	live	0.25	0.75	0.25	0.25	...	NaN

Os três conjuntos de dados (treino, validação e teste), são dispostos de maneira ordenada por era. Em suma o período de 2003 a 2012 é usado para treino e o intervalo entre 2013 e a data presente compõem a base de teste, sendo que alguns intervalos dessa base são convertidos em dados de validação, isto é os verdadeiros valores dos target são liberados.

Capítulo 4

O Modelo Base

Na primeira parte deste capítulo serão explorados os tipos de aprendizado supervisionado para justificar a escolha por modelos de regressão, também será introduzido o modelo de referência (*baseline*) e alguns detalhes básicos do torneio. Por fim serão expostas algumas das métricas utilizadas para um diagnóstico preciso dos modelos. Na segunda parte será introduzido o conceito de diversidade entre os modelos e também uma técnica que permite decompor o sinal gerado pelos estimadores.

4.1 Tipos de Aprendizado Supervisionado

Ao longo da seção 3.1 um problema de aprendizado supervisionado foi configurado foram definidos os seus principais tipos, que são classificação e regressão na seção 2.1. Contudo, é sabido que a avaliação das predições é feita sobre a correlação de spearman, que mede se as predições possui uma relação co-monotônica com o alvo. Logo, pode-se afirmar que trata-se de um **problema de ranqueamento**, modelos de ranqueamento também se enquadram como um tipo de aprendizado supervisionado e também foram definidos na seção 2.1.

Observando a figura 4.1, é possível imaginar como seriam as predições para cada tipo de aprendizado. A figura mostra 4.1 casos onde as predições formariam uma correlação de spearman perfeita com o alvo. Dessa maneira conclui-se que é factível resolver o problema com os três tipos de aprendizado.

4.1.1 Comparando os Tipos de Aprendizado

Uma vez que os três tipos de aprendizado são factíveis, será feito um breve comparativo a fim de observar diferenças de performance e potenciais problemas em alguma das abordagens. Para tal, serão utilizadas três das melhores implementações de algoritmos de árvores de decisão, mantidos por empresas como Microsoft (*Xgboost* e *LightGBM*) citados na seção 2.2.2.1 e Yandex (*Catboost*).

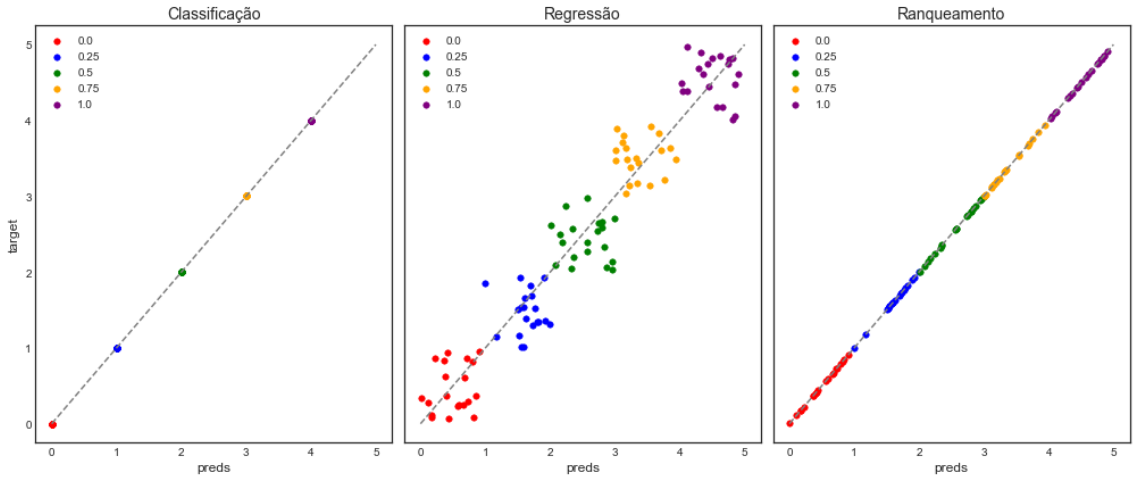


Figura 4.1: Relação Alvo x Predições

Os modelos foram treinados nas primeiras 60 eras do conjunto de dados de treinamento e testados nas últimas 60. Em todos os casos utilizamos 2000 estimadores (árvores) e apenas para fins ilustrativos calculamos o tempo computacional com e sem o auxílio de GPUs. Cabe a nota que todos os modelos treinados daqui em diante farão uso de GPU, com exceção aos modelos lineares onde não há esse suporte.

O objetivo da tabela 4.1 é fazer um comparativo entre os tipos de aprendizado, dessa forma, percebemos que os modelos de regressão tiveram uma melhor performance fora da amostra (*outsample*), seguido por um modelo de ranqueamento. Os modelos de classificação tiveram uma performance abaixo em relação a correlação fora da amostra (4.40% para 3.05%), dessa forma os modelos de classificação serão a princípio descartados.

Tabela 4.1: Comparativo Tipos de Aprendizado

Modelo	outsample	insample	time_elapsed_GPU(s)	time_elapsed(s)
LGBMRegressor	0.0447	0.2677	23.8324	62.3275
XGBRanker	0.0442	0.2021	67.5707	469.0444
XGBRegressor	0.0440	0.2667	37.2466	445.6100
LGBMClassifier	0.0305	0.2070	137.1966	301.8254
XGBClassifier	0.0302	0.1648	232.9470	2172.2911
CatBoostClassifier	0.0271	0.0959	19.0138	1193.6433
CatBoostRegressor	0.0267	0.0949	9.2403	209.8839
LGBMRanker	0.0150	0.0647	6092.7200	16133.8554

O Problema com os Modelos de Classificação

O melhor modelo de classificação no comparativo da tabela 4.1, foi o *LGBMClassifier*, assim as suas predições serão analisadas, de forma a entender qual o problema

com os modelos de classificação. A figura 4.2, mostra que esse modelo obteve uma acurácia de 26%, levando em conta que haviam 5 classes, estamos próximos a um modelo totalmente aleatório.

	precision	recall	f1-score	support
0	0.27	0.28	0.28	21352
1	0.23	0.15	0.18	21363
2	0.27	0.36	0.31	21474
3	0.22	0.15	0.18	21362
4	0.27	0.35	0.30	21344
accuracy			0.26	106895
macro avg	0.25	0.26	0.25	106895
weighted avg	0.25	0.26	0.25	106895

Figura 4.2: Performance Classificador

Contudo é importante lembrar que trata-se de problema com taxa de sinal-ruído muito baixa e uma mínima vantagem, mantida de forma consistente é capaz de trazer enormes ganhos. A questão está na forma como os modelos de classificação em questão cometem erros. É possível ver isso na figura 4.3, onde a matriz de confusão demonstra que o modelo frequentemente classifica uma ação que obteve o pior retorno possível (classe 0), como o melhor possível (classe 4) e vice-versa.

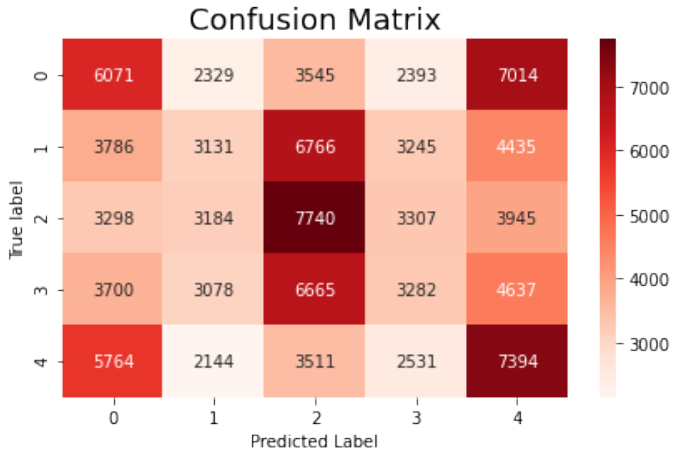


Figura 4.3: Matriz de Confusão

Considerando que o objetivo é otimizar uma métrica de ranqueamento (correlação de spearman), comete-se o pior erro possível, prejudicando bastante a métrica de avaliação. Isso acontece porque as predições das classes extremas (0 e 4), que na prática são, "totalmente vendido"e "totalmente comprado", são altamente correlacionadas (valores próximos a 1 na cor verde), e correlação inversa com as classes intermediárias (valores próximos a -1 na cor vermelha) na figura 4.4.

O mesmo comportamento é observado entre as classes intermediárias (1, 2 e 3), que são altamente correlacionadas entre si e o inverso com as classes extremas. De

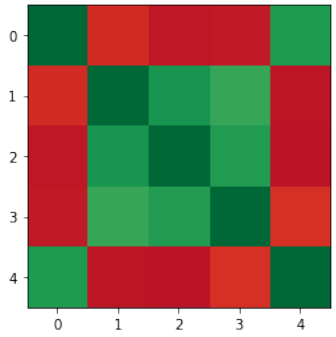


Figura 4.4: Correlação Entre as Classes Preditas

alguma maneira o modelo de classificação vê um comportamento similar entre as classes extremas.

4.2 Modelo de Referência

A Numerai disponibiliza um modelo público que serve como referência para os modelos criados pelos competidores, as predições dessa conta podem ser obtidas junto com os dados do problema e é divulgado que se trata de um modelo *XGBRegressor* com 2000 estimadores como fizemos na seção 4.1.1.

A figura 4.5, traz informações a respeito dessa conta. O modelo está relativamente bem ranqueado, na posição 561 (na data presente, aproximadamente 3000 predições são enviadas por rodada), esse ranking reflete a **correlação de spearman** (ver seção 4.3.1), das predições geradas pelo modelo, com o alvo nas últimas 20 semanas .

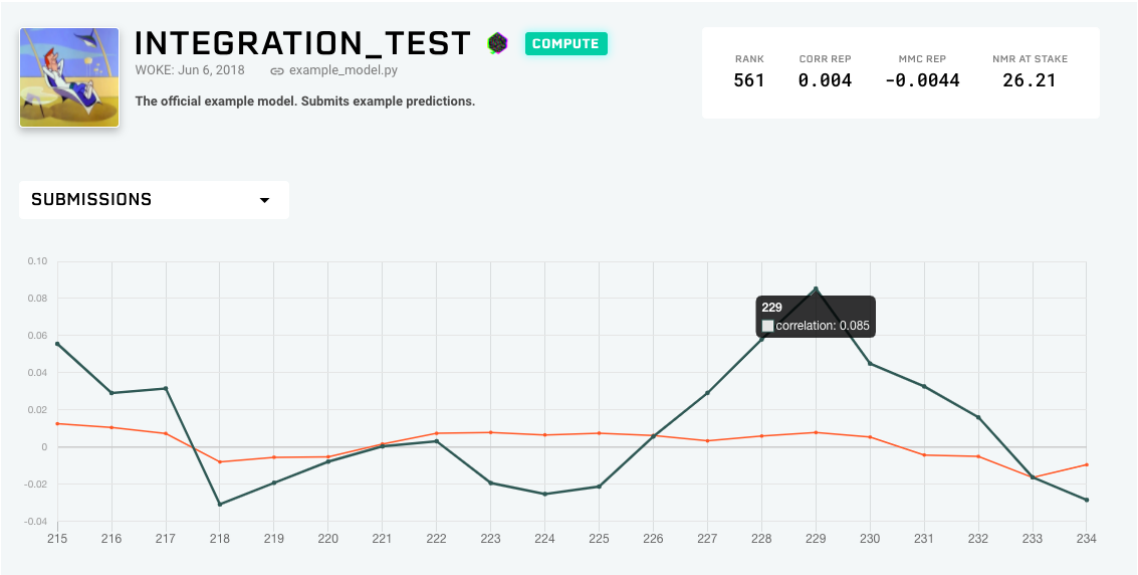


Figura 4.5: Modelo de Referência (Retirado de Numerai [1])

A linha azul da figura 4.5 ilustra a correlação de spearman da rodada 215 a 234,

onde destacamos a rodada 229, onde o modelo obteve uma excelente correlação de 8.5%. Porém o modelo é bastante instável, gerando perdas em várias rodadas no período de 20 semanas.

A linha laranja remete a contribuição (mmc) que as predições do modelo geraram para o fundo da Numerai. Relembre a seção 1.2, o objetivo da Numerai é utilizar as predições dos competidores em seu próprio fundo de investimentos. Podemos entender essa métrica, como uma medida de originalidade das predições (ver seção 4.3.3). Uma vez que diversos competidores encontram um determinado sinal contido nos dados, ele passa a ter um valor menor para a Numerai, pois é provável que já tenha sido absorvido pelo mercado. Por fim, note que o modelo de referência tem um resultado próximo a zero para essa métrica, pois além de se tratar de um modelo simples (divulgado pela própria numerai[1]), é provável que diversos competidores usem soluções similares.

A título de curiosidade o pagamento feito aos competidores a cada rodada se dá pela equação 4.1:

$$payout = stake\_value \cdot payout\_factor \cdot (corr + mmc) \tag{4.1}$$

O payout factor é um fator de desconto nos pagamentos que depende do total de competidores, quanto maior o volume investido pelos participantes maior o fator de desconto, na presente data está em torno de 50%. Além disso a figura 4.5 mostra que a conta possui 26.21 NMR investidos (1NMR = 45 USD). Ou seja para o modelo de referência na rodada 229 temos que:

$$payout = 26.21 \cdot 0.5 \cdot (0.085 + 0.005) = 1.1794 NMR \tag{4.2}$$

Obs: Mais detalhes específicos sobre a competição em si, estão fora do escopo desse trabalho, onde focaremos mais em questões teóricas.

4.3 Diagnóstico do Modelo

Nesta seção serão abordadas as métricas fornecidas pela Numerai para um diagnóstico das predições. A remuneração de fato é feita sobre métricas *Validation_Mean* e *MMC_Mean*, mas as outras métricas do quadro são muito úteis para interpretar o modelo. Essas métricas, como mencionado na seção 1.5, estão separadas em três grupos, performance, risco e originalidade.

É importante salientar que esse diagnóstico fornecido pela é referente ao conjunto de dados de **validação**. Além disso, note que a cor verde no quadro de diagnóstico significa que o modelo chegou a um resultado aceitável para determinada métrica.

[1]Numerai. https://www.numer.ai/integration_test (Acessado em 25/10/2020)

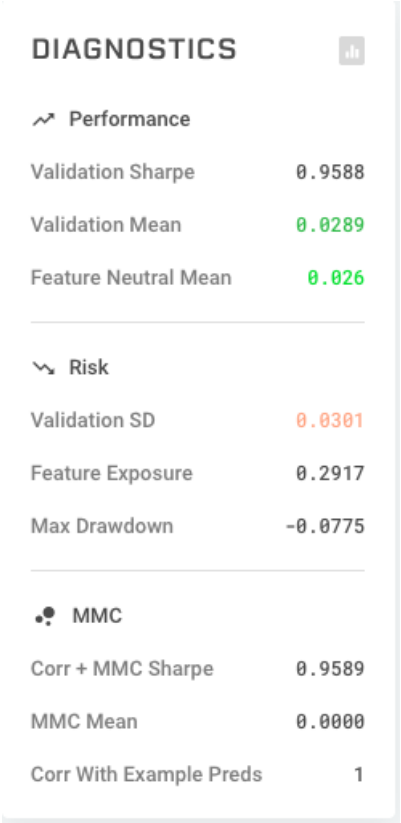


Figura 4.6: Diagnóstico das Predições do Modelo de Referência (Retirado de Numerai [1])

Na cor preta, significa que o resultado foi apenas razoável e a cor vermelha significa um resultado ruim para a métrica correspondente.

A tabela 4.2 mostra o intervalo dos valores considerados para atribuir cores no quadro de diagnóstico. Sendo que o verde mais brilhante, um valor acima do percentil 95, um verde mais escuro acima do percentil 75, cor preta para valores acima do percentil 35 e qualquer coisa abaixo na cor vermelha. Note também para as três métricas em negrito na tabela 4.2, valores menores são mais desejáveis, logo foram invertidos na tabela.

4.3.1 Métricas de Performance

As métricas de performance analisam de maneira direta o retorno das predições, sendo essas as principais métricas na hora de avaliar as predições dos modelos.

Validation Sharpe

DE PRADO [12] se refere ao Sharpe ratio como retorno descontado a uma taxa livre de risco, que no presente estudo se refere a média dos retornos (correlação de spearman) (μ) obtidos por era dividido pelo desvio padrão dos retornos (σ).

Tabela 4.2: Intervalos dos Valores Desejáveis para as Métricas de Validação

Métrica	v.min	v.max
Validation Sharpe	0.53	1.24
Validation Mean	0.013	0.028
Feat. Neutral Mean	0.006	0.022
Validation SD	**0.0303**	**0.0168**
Max Feat. Exp.	**0.4**	**0.0661**
Max Drawdown	-0.115	-0.025
Corr+MMC Sharpe	0.41	1.34
MMC Mean	-0.008	0.008
Corr w/ ex. Preds	**1**	**0.4**

$$Sharpe\_Ratio = \frac{\mu}{\sigma} \tag{4.3}$$

Validation Mean

A média de validação é referente a correlação de spearman obtida em cada era. A correlação de spearman entre duas variáveis é idêntica a correlação de pearson entre os valores de postos das duas variáveis. Posto ou rank, corresponde a ordem de classificação em uma série ordenada de valores, denotados pela subíndice rg [11].

$$r_s = \rho_{\mathrm{rg}_X, \mathrm{rg}_Y} = \frac{\mathrm{cov}(\mathrm{rg}_X, \mathrm{rg}_Y)}{\sigma_{\mathrm{rg}_X} \sigma_{\mathrm{rg}_Y}}, \tag{4.4}$$

A correlação de spearman avalia uma relação monotônica entre duas variáveis contínuas ou ordinais, lineares ou não (ver figura 4.7), em contraste a correlação de pearson, que avalia correlações lineares.

Feature Neutral Mean

A Neutralização (Φ) é um método onde subtraímos a contribuição de outro vetor numérico, enquanto mantém a **informação original**.

Deve-se neutralizar a própria previsão **normalizada** criada pelo modelo original, a partir de modelo linear treinado sobre as previsões. O resíduo resultante será **ortogonal** a uma regressão linear (OLS) como descrito na equação 4.5.

$$\Phi(scores, OLS) = scores - 1 \cdot exp \cdot (exp^{\dagger} \cdot scores) \tag{4.5}$$

Ao final aplica-se uma transformação **minimax** para voltarmos a escala [0,1]. Lembrando que $exp^{\dagger}$ é a pseudo inversa de exp e ainda temos que $exp^{\dagger} \cdot scores = \beta$ (OLS). Mais detalhes sobre nuetralização ao longo do capítulo 5.

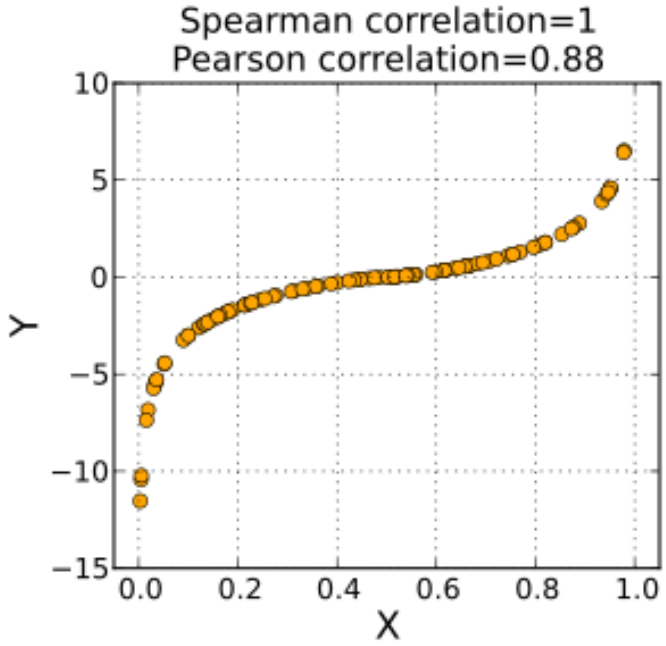


Figura 4.7: Correlação de Spearman (Retirado de DODGE [11])

4.3.2 Métricas de Risco

As métricas de risco analisam se o modelo em questão produz resultados estáveis. Deve-se nos lembrar que na ponta final há um fundo de investimentos onde uma volatilidade excessiva traz riscos ao gestor.

Validation SD

Representa o desvio padrão dos scores obtidos nos dados de validação.

$$Validation\_SD = \sqrt{\frac{1}{N}\sum_{s=1}^{S}(s_i - \mu)^2} \tag{4.6}$$

Onde μ é a média dos scores e N o tamanho do vetor de scores S.

Feature Exposure

É correlação de pearson (linear) de cada umas das 310 variáveis com as predições geradas pelo modelo, em cada era individualmente. Dessa forma é possível identificar se as predições estão confiando excessivamente em um único indicador. Ao final, calcula-se a média da maior exposição (fe_max) em cada uma das 120 eras como exposto na equação 4.7.

$$fe\_max = \frac{\sum_{era=1}^{120} fe\_max_{era}}{120} \tag{4.7}$$

Max Drawdown

DE PRADO [12] descreve como a perda máxima obtida entre dois pontos de máximo (picos) subsequentes. Descritos na equação 4.8 e na figura 4.8.

$$Max\_Drawdown(T) = max\left\{0, \underset{t\in(0,T)}{max} X(t) - X(T)\right\} \tag{4.8}$$

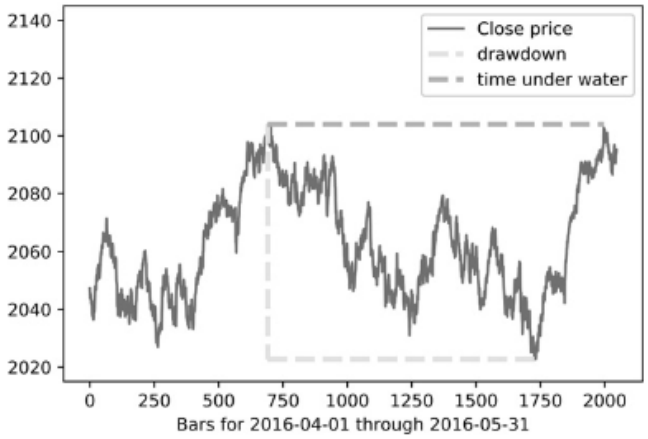


Figura 4.8: Ilustração Queda Máxima (Retirado de DE PRADO [12])

4.3.3 Métricas de Originalidade

O meta modelo da numerai, que monta as posições do fundo é uma média de todas as predições enviadas pelos participantes. Nesse cenário a fim de estimular a diversidade de modelos, faz sentido premiar a originalidade no sinal encontrado por diferentes soluções. Nesta seção serão tratadas as métricas específicas para medir a originalidade.

Corr + MMC Sharpe

De maneira análoga ao sharpe ratio dos retornos calculado na seção 4.3.1, aqui utiliza-se a média (μ) e o desvio padrão (σ) dos retornos somado ao MMC.

$$Corr\_MMCSharpe = \frac{(\mu + mmc)}{\sigma_{\mu,mmc}} \tag{4.9}$$

MMC Mean

A contribuição ao meta modelo (MMC), é obtida através da neutralização das predições originais do modelo, pelo meta modelo (que é a média dos outros competidores), de forma a resta apenas o sinal ortogonal como ilustrado na figura 4.9

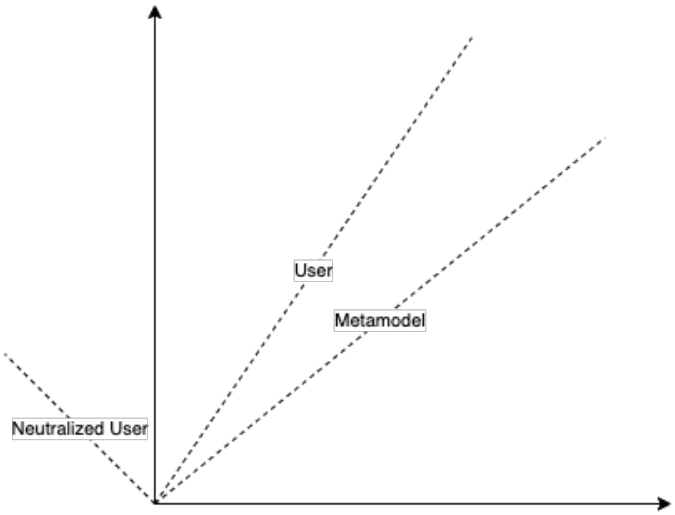


Figura 4.9: Contribuição ao Meta Modelo (Retirado de Numerai [1])

Para fins de remuneração, a numerai seleciona de maneira aleatória 67% das predições enviadas ponderadas pelo valor financeiro colocado pelo próprio participante nessas predições (entende-se que quanto maior o valor investido, maior a confiança no modelo que gerou as predições).

Para fins de diagnóstico, como as predições enviadas pelos outros participantes não são conhecidas, as predições serão neutralizadas pelo modelo de referencia. Note que na figura 4.6 temos o o *MMC Mean* igual a zero para o próprio modelo de referência.

Esse esquema é análogo ao *Feature Neutral Mean* onde descontava-se uma regressão linear das predições originais. Em seguida é preciso calcular a covariância entre as predições normalizadas e o target e então dividir pela variância da distribuição uniforme (i.e 0.0841) como visto na equação 4.10.

$$MMC_{scores} = \frac{\text{cov}(\Phi_{scores,meta}, \text{target})}{0.0841}, \tag{4.10}$$

Este último passo é feito para que o MMC fique na mesma magnitude dos scores (spearman) e torná-lo mais interpretável. Por fim, os scores calculados por era e retorna-se assim a sua média.

Corr w/ Example Preds

É a correlação de spearman medida na seção 4.3.1, das predições do modelo, com as predições do modelo de referencia da seção 4.2. A partir desta, obtêm-se uma métrica de similaridade com o meta modelo, mesmo que de maneira **aproximada**, note também que na figura 4.6 o valor da correlação é igual a 1 pois utilizou-se o próprio modelo de referência.

4.4 Diversidade de Modelos

Optou-se por razões de simplicidade utilizar apenas o modelo *Xgboost*, detalhado na seção 2.2.2.2 que é idêntico ao modelo de referência da seção 4.2, com os seguintes hiperparâmetros:

- **Estimadores** = 2000: Quantidade de árvores utilizadas
- **Taxa de aprendizado** = 0.01: Taxa de atualização do resíduo predito por cada estimador
- **Máxima profundidade** = 5: Profundidade máxima da árvore
- **Colunas por árvore** = 0.1: Percentual de colunas sorteadas para a construção de cada árvore.

De todo modo será feito um breve comparativo entre os modelos *Xgboost*, *LightGBM* (seção 2.2.2.2) e *XgbRanker* (seção 2.2.2.3). Utilizando 3 estratégias de busca de hiperparâmetros, validação cruzada (seção 2.5.2) e série temporal (seção 2.5.3) com busca aleatória (seção 2.6.1) além de busca bayesiana 2.6.2, totalizando 9 modelos.

A conclusão deste comparativo é que modelos de *Gradient Boosting* mesmo com diferentes conjuntos de hiperparâmetros geram resultados altamente correlacionados. A figura 4.10 mostra a correlação de spearman por era gerada por cada um dos 9 modelos na base de validação. Note que há pouca **diversidade** nos *scores* gerados entre esses modelos, todos ganham ou perdem exatamente nas mesmas eras.

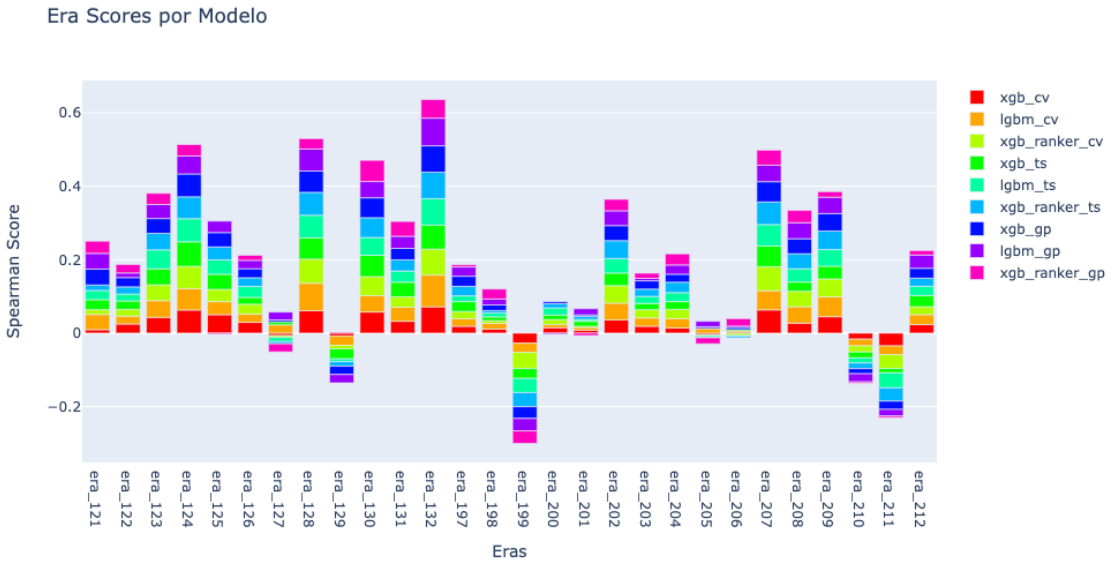


Figura 4.10: Comparativo Modelos Não Lineares

É importante ressaltar que o score acumulado nas 22 eras pode apresentar resultados discrepantes. Um modelo pode ter uma performance acumulada significativamente superior a outro, contudo essa base de 22 eras é relativamente pequena se comparada a base de treino (120 eras) e principalmente, pode não ser representativa das eras posteriores, logo é possível que haja uma correção na performance desses estimadores e suas performances voltem a convergir no período subsequente.

Esse comportamento também é percebido entre os modelos lineares citados na seção 2.2.1 e ilustrado na figura 4.11.

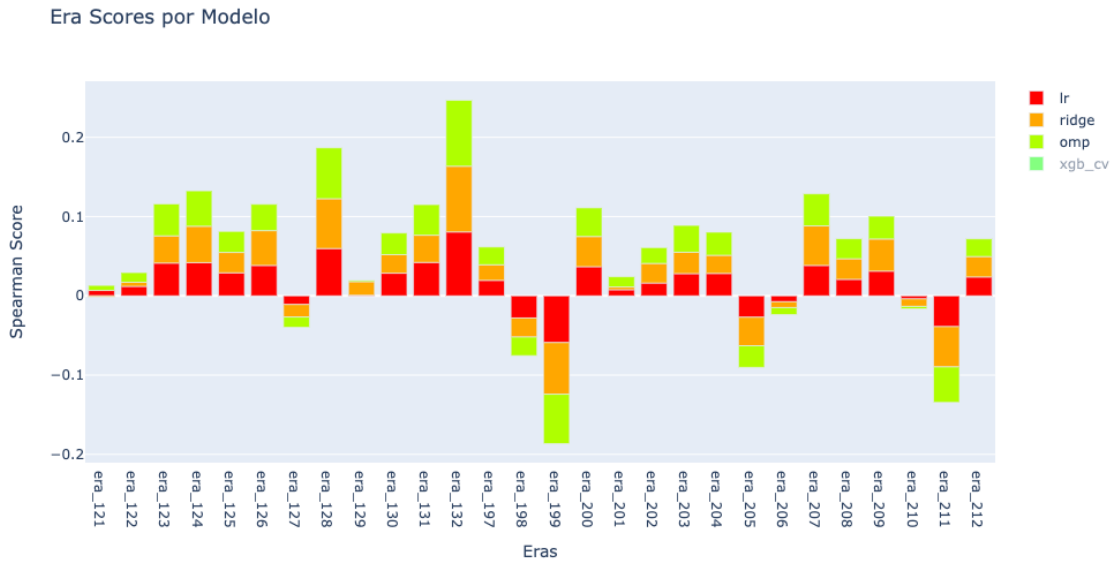


Figura 4.11: Comparativo Modelos Lineares

Por outro lado a correlação dos scores dos modelos lineares com os não lineares é menor se comparado a correlação entre eles, como mostra a figura 4.12.

Por se tratar de um modelo mais robusto, o modelo *Xgboost* possui uma performance superior a regressão linear mostrado no quadro de diagnóstico da figura 4.13. Note que o modelo não linear obteve melhor resultado em 6 das 9 variáveis, especialmente nos scores (*Validation Mean*). Note também a grande presença de variáveis na cor vermelha indicando um performance ruim, além da baixa contribuição ao meta modelo *MMC Mean* de ambos os modelos, indicando pouca originalidade nas predições.

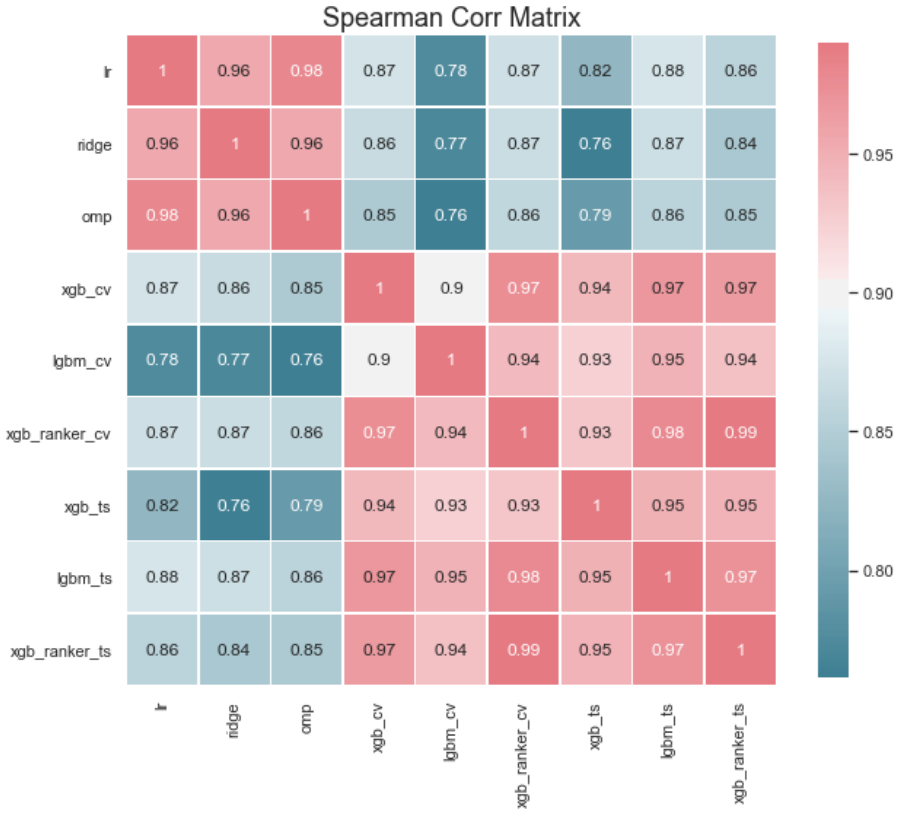


Figura 4.12: Correlação Entre Modelos Base

	lr	xgb_cv
Validation_Sharpe	0.5461	0.8126
Validation_Mean	0.0163	0.0224
Feat_neutral_mean	0.0030	0.0148
Validation_SD	0.0299	0.0276
Feat_exp_max	0.2784	0.2836
Max_Drawdown	-0.0853	-0.0506
corr_plus_mmc_sharpe	0.3629	0.6772
val_mmc_mean	-0.0008	-0.0012
corr_with_example_preds	0.6550	0.9065

Figura 4.13: Diagnóstico Modelo de Árvore e Regressão Linear

4.5 Decomposição das Predições

Com o estimador em mãos, será utilizado método proposto por LI *et al.* [13], para decompor as predições do modelo em porções linear, não linear e em pares. A metodologia proposta no artigo é uma extensão do conceito de função dependência parcial [37], que estima a contribuição marginal dada por uma variável isolada as predições do modelo. O procedimento para estimar a função de dependência parcial a partir dos dados empíricos é como se segue:

1. Selecione um valor único existente para a variável de entrada selecionada (x_k)
2. Combine esse valor com o vetor das variáveis de entrada restantes ($x_{/k}$) e gerar uma nova predição
3. Repetir o passo 2 para cada vetor de entrada ($x_{/k}$), mantendo o valor de x_k constante
4. Calcule a média de todas as predições do passo 3 para obter o valor da predição parcial para este ponto ($\hat{\gamma}_{x_k}$)
5. Repetir os passos anteriores para todos os valores únicos de x_k e gerar a função de dependência resultante

Observando o gráfico a esquerda da figura 4.14, a função de dependência parcial terá poucos desvios se a dada variável tiver pouca influência nas predições do estimador. Note também que caso o estimador seja uma regressão linear, a função de dependência se aproximará de uma reta com uma inclinação igual ao coeficiente de regressão da variável.

Em seguida, para decompor o efeito marginal da variável em uma componente linear e não linear, basta ajustar uma nova regressão linear sobre a função de dependência como visto no gráfico do meio da figura 4.14. O resíduo dessa operação pode ser entendido como a porção não linear, visto no gráfico da esquerda da figura 4.14.

É possível também podemos obter o sinal obtido pela interação em pares de cada par de variáveis, que por sua vez pode ser estimado repetindo o mesmo procedimento de 5 passos descrito anteriormente, porém levando em consideração os pares de valores únicos das duas variáveis selecionadas. Após esse cálculo, é necessário subtrair a contribuição obtida por cada variável de maneira isolada. É importante mencionar que o valor da contribuição não linear é obtido do resíduo da contribuição linear e da contribuição em pares. O valor total da contribuição é dado pela equação 4.11.

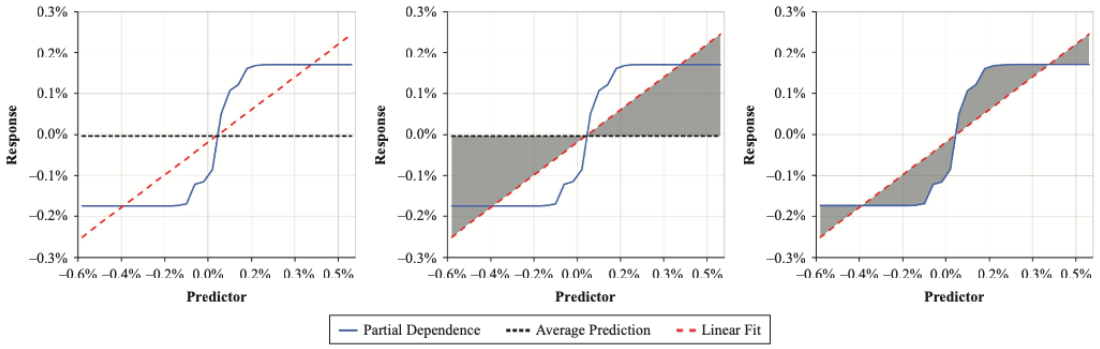
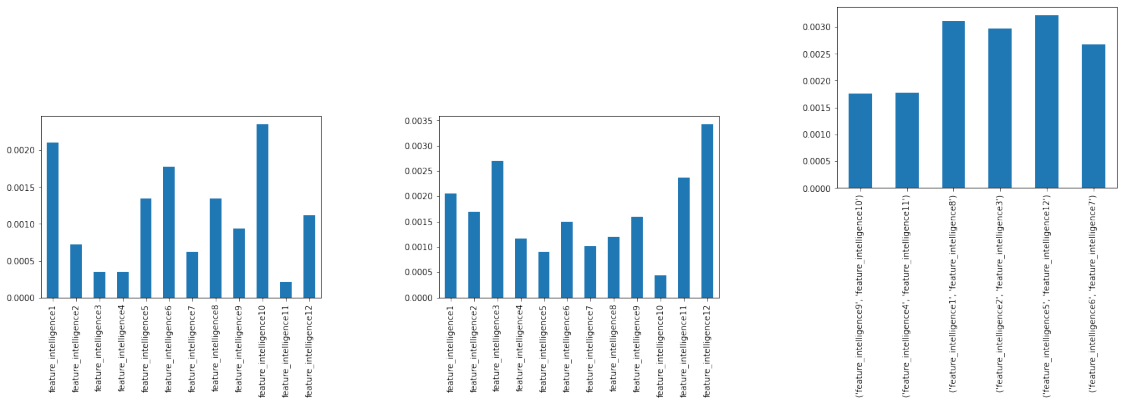


Figura 4.14: Função de Dependência Parcial (Retirado de LI *et al.* [13])

$$\hat{\gamma} = \sum_{k=1}^{M} Linear(x_k) + \sum_{k=1}^{M} Naolinear(x_k) + \sum_{(k,l)\in M, k\neq l} Pares(x_k, x_l) \tag{4.11}$$

Observando novamente a figura 4.11, perceba que a maior parte do sinal obtido por um modelo de *Gradient Boosting* é linear, além disso, LI *et al.* [13] afirmam que há pouca diferença no sinal linear obtido por um modelo desse tipo com um estimador linear.

Voltando ao presente estudo, desenvolveu-se o mesmo método debatido ao longo desta seção a fim de observar a composição do sinal obtido pelo estimador. Como o objetivo é apenas ilustrativo, consideremos apenas um dos grupos de variáveis a fim de facilitar a visualização. A figura 4.15 ilustra esse experimento e podemos notar que a maior parte do sinal é obtido pela interação em pares. Esse pode ser um indicativo que os indicadores contidos nesse grupo podem não funcionar bem de maneira isolada, mas possuem valor preditivo se analisados em conjunto de outras variáveis.



(a) Linear (b) Não Linear (c) Pairwise

Figura 4.15: Decomposição das Predições

Ao longo do capítulo 5 será demonstrada uma técnica para decompor a componente linear capturada pelo estimador do restante e assim gerar uma nova predição a partir desta nova componente.

Capítulo 5

Neutralização

Ao longo deste capítulo serpa introduzido o conceito de neutralização além de explorar diversas estratégias de neutralização de variáveis. Em seguida, serão observados os efeitos da neutralização sobre as predições e será desenvolvido um método para selecionar estratégias. Por fim será verificado se a neutralização é eficaz em reduzir a volatilidade dos modelos.

5.1 Introdução

A seção 4.5 demonstrou que um estimador não linear robusto é capaz de capturar diferentes tipos de relacionamentos entre as variáveis de entrada. Ao longo deste capítulo será feita uma decomposição das predições a fim de estudar seu comportamento entre as métricas de avaliação e se devemos mantê-las ou descartá-las.

Nesse contexto, retornando a seção 4.3.1, abordou-se a métrica *Feature Neutral Mean* que é semelhante a uma técnica da econometria chamada residualização, que é uma aplicação do teorema de Frisch-Waugh-Lovell desenvolvido em 1933 [38]. Na residualização, deve-se subtrair a predição **normalizada** gerada pelo estimador, por um modelo linear treinado sobre as previsões originais.

$$\Phi(scores, OLS) = scores - 1 \cdot exp \cdot (exp^{\dagger} \cdot scores) \tag{5.1}$$

O resíduo dessa operação corresponde a porção *não linear encontrada* pelo modelo, que no nosso caso é o modelo de referência demonstrado na seção 4.4.

5.1.1 Diagnóstico da Neutralização

Nesta seção o efeito da neutralização sobre as predições do modelo de referência serão observadas, na figura 5.1 observa-se o diagnóstico do modelo de referência (*ex_preds*), o modelo de referência neutralizado (*ex_FN100*) e uma regressão linear.

	ex_preds	ex_FN100	lr
Validation_Sharpe	0.8861	1.0475	0.5461
Validation_Mean	0.0241	0.0207	0.0163
Feat_neutral_mean	0.0182	0.0198	0.0030
Validation_SD	0.0272	0.0198	0.0299
Feat_exp_max	0.2666	0.0140	0.2784
Max_Drawdown	-0.0353	-0.0217	-0.0853
corr_plus_mmc_sharpe	0.8861	0.8111	0.3629
val_mmc_mean	0.0000	0.0046	-0.0008
corr_with_example_preds	1.0000	0.6063	0.6550

Figura 5.1: Diagnóstico Componentes

Veja que o modelo **neutralizado** apresentou uma melhora em diversas métricas onde destacamos:

- **Sharpe Ratio**: O modelo neutralizado é significativamente mais estável, mesmo descartando todo o sinal linear capturado pelo modelo (*validation mean*), houve uma redução significativa de volatilidade (*validation SD*).
- **Max_Feat_Exp**: Essa métrica mede a correlação linear de cada variável com as predições e uma vez que eliminou-se todas as correlações lineares, esse valor tende a ser próximo a zero. O fato do modelo não confiar em variáveis isoladas o torna mais estável.
- **Val_MMC_Mean**: A neutralização ajudou as predições do modelo a se tornarem menos correlacionadas com o modelo de referência, que é usado como estimativa para o cálculo do MMC.

Contudo é importante salientar que esse resultado é referente a um recorte específico no tempo (28 meses) compreendido por esse conjunto de validação, não há nenhuma garantia que esse ganho de performance se realizará em eras futuras.

5.2 Customizando a Neutralização

Nesta seção serão abordadas diversas maneiras de aplicar neutralização, a fim de obter melhor performance e diversidade de resultados.

5.2.1 Proporções de Neutralização

Modificando a equação 4.5, obtém-se a equação 5.2, onde o parâmetro *prop* nos permite aplicar diferentes níveis de neutralização sobre as predições.

$$\Phi(scores, OLS) = scores - prop \cdot exp \cdot (exp^{\dagger} \cdot scores) \tag{5.2}$$

Para uma $prop = 0.5$, apenas 50% da componente linear do sinal é removida. A figura 5.2 ilustra o efeito que múltiplas proporções de neutralização exercem sobre as predições. É fácil perceber que há uma mudança gradativa nas métricas entre os intervalos com e sem neutralização (FN_0.0 e FN_1.0) da figura 5.1. Além disso, é bastante tentador tentar calibrar a proporção de neutralização testando intervalos cada vez menores, mas assim serão criados estimadores muito correlacionados e a proporção ótima encontrada será apenas um sobreajuste deste conjunto de dados.

	FN_0.0	FN_0.25	FN_0.5	FN_0.75	FN_1.0	FN_1.5	FN_2.0	FN_-1.0
Validation_Sharpe	0.8861	0.9569	1.0382	1.1056	1.0475	0.4545	0.1040	0.7259
Validation_Mean	0.0241	0.0246	0.0245	0.0234	0.0207	0.0114	0.0031	0.0219
Feat_neutral_mean	0.0182	0.0187	0.0190	0.0192	0.0198	0.0201	0.0198	0.0161
Validation_SD	0.0272	0.0257	0.0236	0.0212	0.0198	0.0251	0.0302	0.0301
Feat_exp_max	0.2680	0.2324	0.1776	0.0982	0.0140	0.1876	0.2760	0.3244
Max_Drawdown	-0.0353	-0.0273	-0.0229	-0.0232	-0.0217	-0.0500	-0.1292	-0.0512
corr_plus_mmc_sharpe	0.8861	1.0038	1.0848	1.0195	0.8111	0.3973	0.1761	0.6142
val_mmc_mean	0.0000	0.0005	0.0014	0.0027	0.0046	0.0065	0.0058	-0.0009
corr_with_example_preds	1.0000	0.9897	0.9426	0.8225	0.6063	0.1124	-0.1951	0.9559

Figura 5.2: Aplicando Proporções de Neutralização

Uma outra forma interessante de gerar mais diversidade no diagnóstico é extrapolar a proporção de neutralização para valores além de 0 e 1, qualquer valor acima de 1 pode ser entendido como uma aposta contrária ao sinal linear encontrado pelo modelo, algo semelhante a operar vendido (*short*). Veja na figura 5.3, que os scores de um modelo 200% neutralizado se comportam de maneira quase oposta ao modelo de referência. Por fim, qualquer proporção abaixo de zero estamos adicionando mais sinal linear as predições do estimador (*alavancagem*).

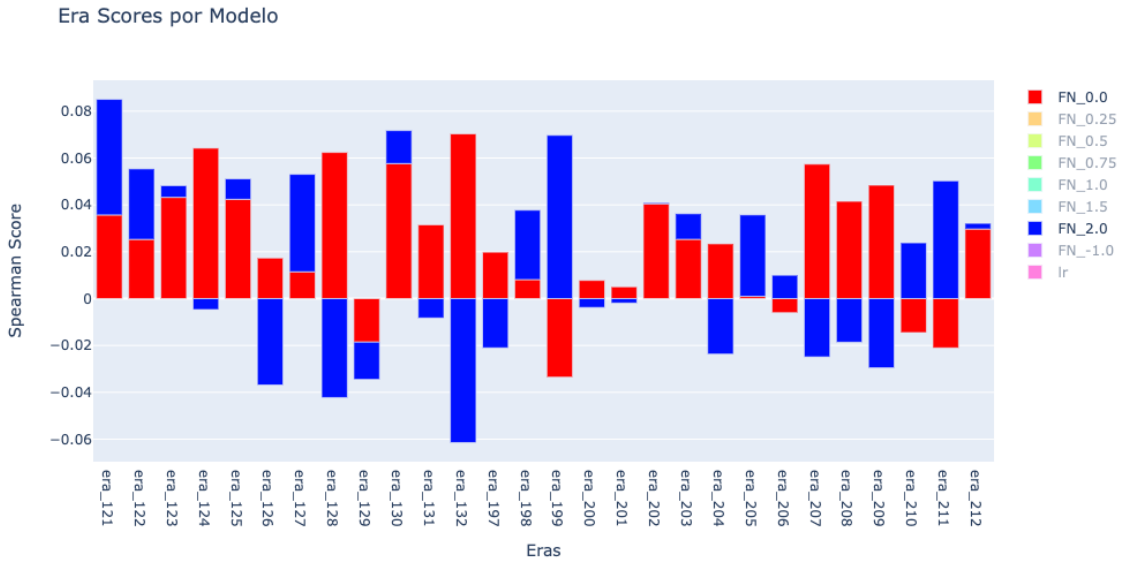


Figura 5.3: Era Scores Comparativo 200% Neutralizados

Apesar de em um primeiro momento essas estratégias não apresentarem bons resultados, veremos como utilizá-las mais adiante. Por fim, a figura 5.4 mostra a correlação entre os scores gerados utilizando múltiplas proporções de neutralização, além da regressão linear. Podemos constatar que atingiu-se uma correlação significativamente baixa para valores de neutralização acima de 100%.

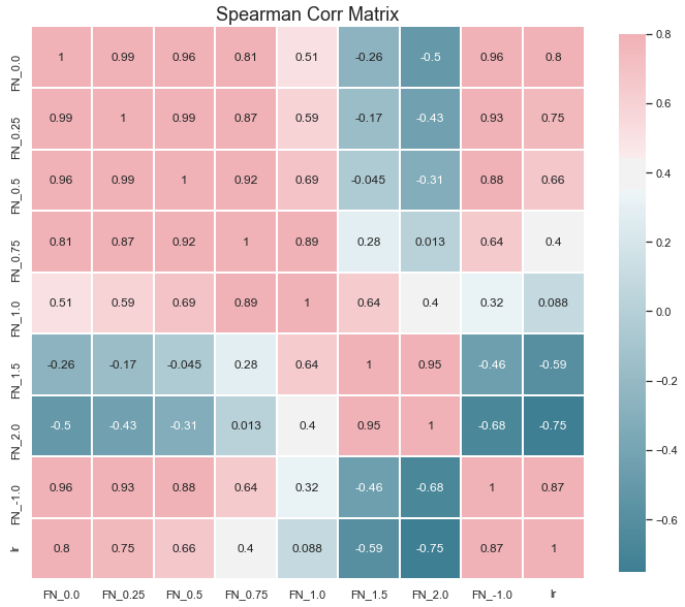


Figura 5.4: Correlação Era Scores por Proporção

5.2.2 Neutralizando Grupos de Variáveis

O método proposto pela equação 5.2 também permite através do parâmetro *exposures* selecionar quais variáveis serão neutralizadas, na prática será treinada uma regressão linear utilizando apenas as variáveis selecionadas e consequentemente apenas a exposição linear dessas variáveis serão eliminadas. A figura 5.5 ilustra as correlações lineares entre as variáveis e as predições geradas pelo modelo de referência. Repare que os diferentes grupos possuem comportamentos distintos, e alguns deles possuem sutis mudanças de comportamento ao longo das eras.

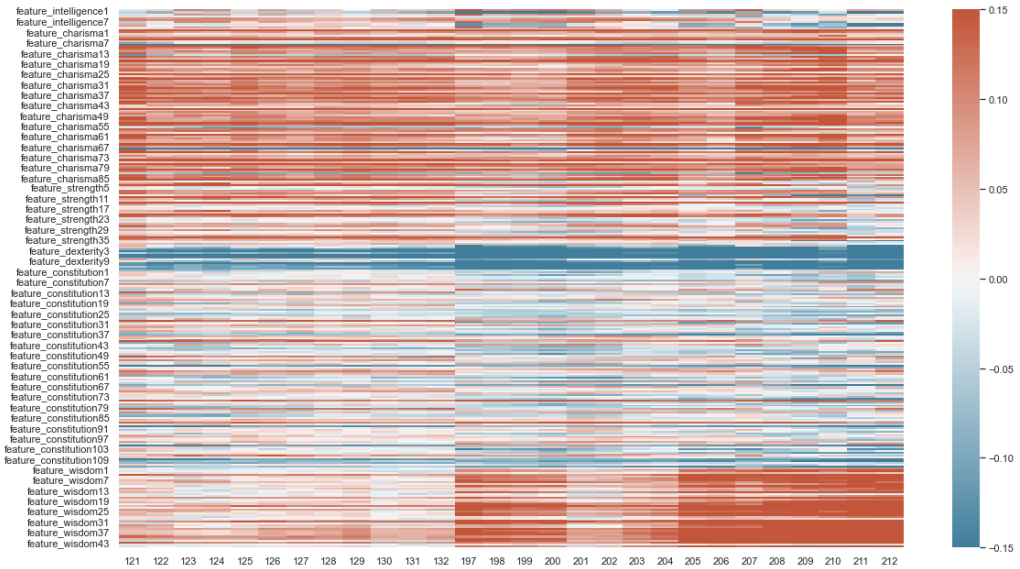


Figura 5.5: Exposições Modelo sem Neutralização

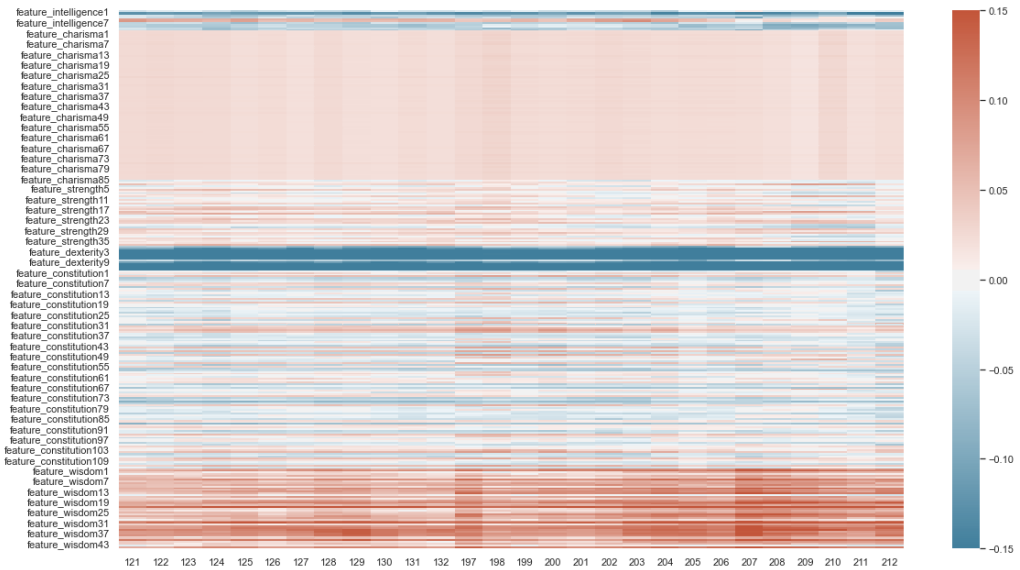


Figura 5.6: Exposições com Neutralização

A figura 5.6 ilustra o mesmo modelo porém performou-se uma neutralização nos grupos *Constitution*, *Strenght* e *Charisma*. Repare que as correlações nesses grupos

agora se aproximam de zero, restando apenas a **porção não linear** capturada por essas variáveis.

Voltando a tabela 3.7, será feita uma neutralização de 100% em todas as combinações com no mínimo 3 dos 6 grupos de variáveis . Isto é $\sum_{i=3}^{6} \binom{6}{i} = 42$ combinações. A figura 5.7 mostra que foi possível algum nível de diversidade nos scores se o grupos neutralizados forem suficientemente diferentes.

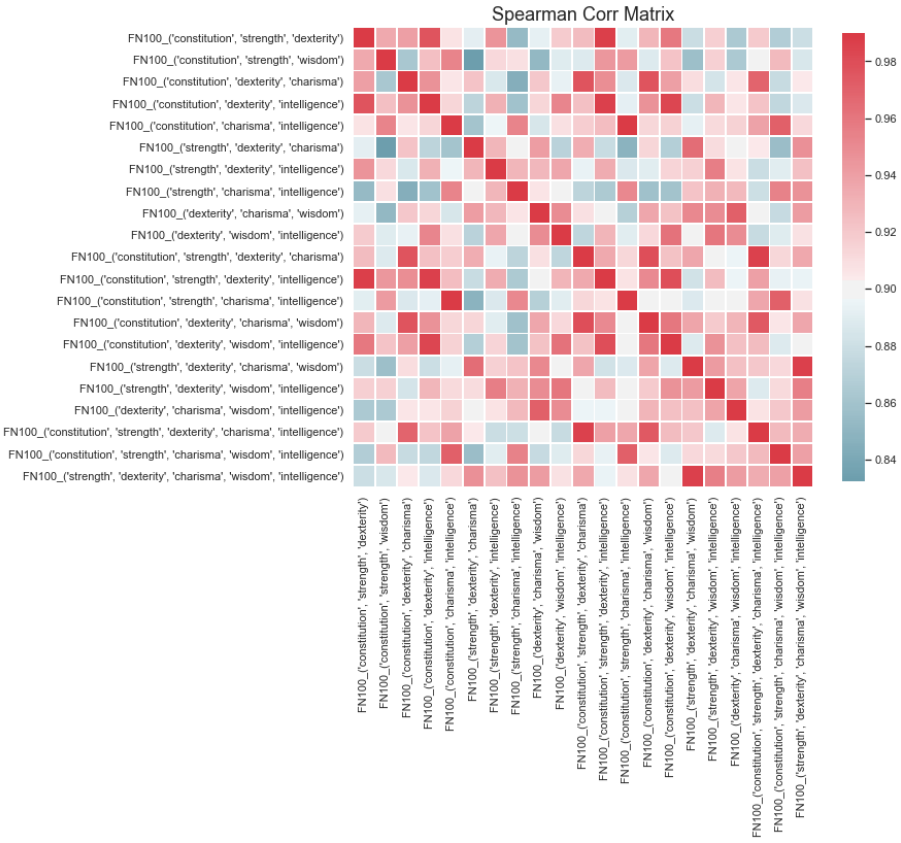


Figura 5.7: Correlação Era Scores por Grupo

Ao analisar o diagnóstico de todas as combinações geradas, percebe-se que determinados grupos aparecem com mais frequência nos melhores resultados, e que nenhum deles é superior em todas as métricas analisadas. Escolher um modelo entre estratégias similares não é exatamente relevante, uma vez que reflete o resultado em um recorte específico no tempo. De qualquer forma, selecionou-se a estratégia que neutraliza os grupos *Constitution*, *Strenght*, *Dexterity* e *Intelligence*.

FN100_('constitution', 'strength', 'dexterity', 'intelligence')	FN100_('constitution', 'strength', 'charisma', 'intelligence')	FN100_('constitution', 'dexterity', 'charisma', 'wisdom')	FN100_('constitution', 'dexterity', 'wisdom', 'intelligence')	FN100_('strength', 'dexterity', 'charisma', 'wisdom')	FN100_('strength', 'dexterity', 'wisdom', 'intelligence')	F
1.3026	1.1311	1.1589	1.1662	1.0657	1.1184	
0.0288	0.0279	0.0237	0.0267	0.0208	0.0237	
0.0186	0.0197	0.0186	0.0186	0.0196	0.0190	
0.0221	0.0246	0.0205	0.0229	0.0196	0.0212	
0.2222	0.2420	0.1680	0.2257	0.1777	0.2293	
-0.0162	-0.0381	-0.0203	-0.0094	-0.0188	-0.0107	
1.4638	1.1223	1.1113	1.2099	0.9567	1.1657	
0.0071	0.0071	0.0041	0.0051	0.0015	0.0024	
0.8071	0.7683	0.7563	0.8253	0.7831	0.8507	

Figura 5.8: Diagnóstico Neutralização por Grupos

Em seguida extrapolou-se a proporção de neutralização nesses mesmos grupos, como visto na figura 5.9. Pode-se ver que este último modelo supera com larga margem o modelo de referência, além disso a maior parte das métricas se encontra na cor verde, indicando boa performance, além de possuir uma baixa correlação com o modelo de referência (0.4677).

	ex_preds	FN100_('constitution', 'strength', 'dexterity', 'intelligence')	FN200_('constitution', 'strength', 'dexterity', 'intelligence')
Validation_Sharpe	0.8861	1.3026	1.4416
Validation_Mean	0.0241	0.0288	0.0254
Feat_neutral_mean	0.0182	0.0186	0.0177
Validation_SD	0.0272	0.0221	0.0176
Feat_exp_max	0.2666	0.2222	0.2077
Max_Drawdown	-0.0353	-0.0162	-0.0042
corr_plus_mmc_sharpe	0.8861	1.4638	1.3725
val_mmc_mean	0.0000	0.0071	0.0107
corr_with_example_preds	1.0000	0.8071	0.4677

Figura 5.9: Era Scores Modelo 200% Neutralizados nos Grupos

Contudo, apesar do resultado aparentemente satisfatório, é possível constatar diversos problemas na construção desta estratégia:

- A base de dados é relativamente pequena
- Houve um vazamento de informação ao escolhermos o melhor subconjunto de variáveis para a neutralização com 200%
- Desconsiderou a distribuição dos retornos gerada pela estratégia selecionada
- Desconsiderou os retornos gerados pelas outras tentativas

Esses problemas são um indicativo que esta boa performance pode não se repetir no futuro. Na seção 5.3 será demonstrado como identificar esses problemas e descartar estratégias potencialmente ruins.

5.3 Detectando Estratégias Falso Positivas

Na última seção foram identificadas quatro problemas na elaboração da estratégia de neutralização por grupos. Ao longo desta seção serão abordadas maneiras de mitigas esses problemas e ao final será aplicada uma metologia para descartar estratégias que contém indícios de serem falso positivas.

5.3.1 Validação Cruzada

O grande problema da base de validação usada anteriormente é o seu tamanho (28 meses), relativamente pequena, se comparada com a a base de treino (120 meses), essa base é capaz de compreender diferentes regimes de mercado e avaliar de maneira mais eficaz se a estratégia funcionará bem no longo prazo. Contudo, não se deve utilizar a base de treino diretamente, pois o modelo de referência já foi treinado nesta base, portanto para reaproveitarmos esses dados, as predições serão geradas a partir da mesma configuração do modelo de referência utilizando validação cruzada, ou mais especificamente a técnica *group k-fold* explicada na seção 2.5.2.

5.3.2 Viés de Seleção

Ao longo de toda seção 5.2 testou-se os resultado de diversas estratégias e a cada passo utilizou-se a informação gerada para gerar um novo subconjunto de estratégias a serem testadas. Isto é, incorreu-se em um viés de seleção a cada vez que se observou o resultado dos testes (*data snooping*).

5.3.3 Analisando a Distribuição dos Retornos

O *Sharpe Ratio* ($SR = \frac{\mu}{\sigma}$), apresentado na seção 4.3.1, é a métrica mais utilizada para medir a relação risco-retorno. Contudo os valores futuros de μ e σ são desconhecidos e obtém-se apenas uma estimativa desses valores uma vez que utilizamos o histórico dos retornos e cada estimativa possui uma variância e um nível de confiança.

Assumindo uma distribuição normal dos retornos, o Sharpe Ratio estimado ($\widehat{SR}$) também segue uma distribuição normal com desvio padrão ($\hat{\sigma}(\widehat{SR})$) [39]:

$$\hat{\sigma}(\widehat{SR}) = \sqrt{\frac{1}{n-1}\left(1 + \frac{1}{2}\widehat{SR}^2\right)} \tag{5.3}$$

Segundo BAILEY e DE PRADO [40] o viés (γ_3) e a curtose (γ_4) dos retornos não afetam diretamente o Sharpe Ratio, porém impactam no intervalo de confiança e na significância estatística. Então estendendo a equação 5.3 temos:

$$\hat{\sigma}(\widehat{SR}) = \sqrt{\frac{1}{n-1}\left(1 + \frac{1}{2}\widehat{SR}^2 - \gamma_3\widehat{SR} + \frac{\gamma_4 - 3}{4}\widehat{SR}^2\right)} \tag{5.4}$$

BAILEY e DE PRADO [40] ainda declaram que o Sharpe Ratio estimado sempre seguirá uma distribuição normal, mesmo que os retornos não sigam, ou seja, $(\widehat{SR} - SR) \rightarrow N\left(0, \hat{\sigma}(\widehat{SR})\right)$. Nesse contexto, dado um valor base (SR^*) o Sharpe Ratio estimado ($\widehat{SR}$) pode ser expresso em termos de probabilidade, onde Z é a distribuição cumulativa da distribuição normal:

$$\widehat{PSR}(SR^*) = Z\left[\frac{\left(\widehat{SR} - SR^*\right)}{\hat{\sigma}(\widehat{SR})}\right] = Z\left[\frac{\left(\widehat{SR} - SR^*\right)\sqrt{n-1}}{\sqrt{1 + \frac{1}{2}\widehat{SR}^2 - \gamma_3\widehat{SR} + \frac{\gamma_4 - 3}{4}\widehat{SR}^2}}\right] \tag{5.5}$$

A partir da equação 5.5 e um dado SR^*, conclui-se que o $\widehat{PSR}$ aumenta com o valor de $\widehat{SR}$ com o crescimento do histórico dos retornos com viés (γ_3) positivos, porém o PSR diminui com excesso de curtose (γ_4), sendo que as duas últimas penalizam diretamente retornos menos estáveis.

Sharpe Ratio Ajustado

Apesar de ser importante garantir que um modelo não apresente retorno médio negativo, em algumas ocasiões é difícil em distinguir qual modelo apresenta uma melhor performance ($\widehat{PSR} \approx 100\%$), então em algumas ocasiões o *Sharpe Ajustado*, proposto por ALEXANDER e SHEEDY [41], representado na equação 5.6 pode ser útil.

$$ASR = SR\left[1 + \left(\frac{\gamma_3}{6}\right)SR - \left(\frac{[\gamma_4 - 3]}{24}\right)SR^2\right] \tag{5.6}$$

5.3.4 O Problema dos Múltiplos Testes

BAILEY *et al.* [42] declaram que a maior parte dos testes publicados em artigos é falho devido ao viés de seleção em múltiplos testes. Os únicos testes reportados pelos pesquisados são aqueles que refletem estratégias supostamente vencedoras. Além disso de acordo com BAILEY *et al.* [42] lidar com o sobreajuste é possivelmente a questão mais fundamental das finanças quantitativas.

Nesse contexto, o máximo valor esperado do $\widehat{SR}$, para $N \gg 1$ tentativas independentes, pode ser aproximado como:

$$E\left[\max\left\{\widehat{SR}_n\right\}\right] \approx E\left[\left\{\widehat{SR}_n\right\}\right] + \sqrt{V\left[\left\{\widehat{SR}_n\right\}\right]}\left((1-\gamma)Z^{-1}\left[1-\frac{1}{N}\right]+\gamma Z^{-1}\left[1-\frac{1}{N}e^{-1}\right]\right) \tag{5.7}$$

Onde $V\left[\left\{\widehat{SR}_n\right\}\right]$ é a variância ao longo das tentativas do $\widehat{SR}$ e γ é a constante de Euler-Mascheroni (≈ 0.5772). Através da equação 5.7 e assumindo $E[SR] = 0$ e $V[SR] = 1$, verifica-se pela figura 5.10 que após apenas 1000 **tentativas independentes** $E\left[\max\left\{\widehat{SR}_n\right\}\right] = 3.26$, mesmo que o Sharpe Ratio real seja zero [43].

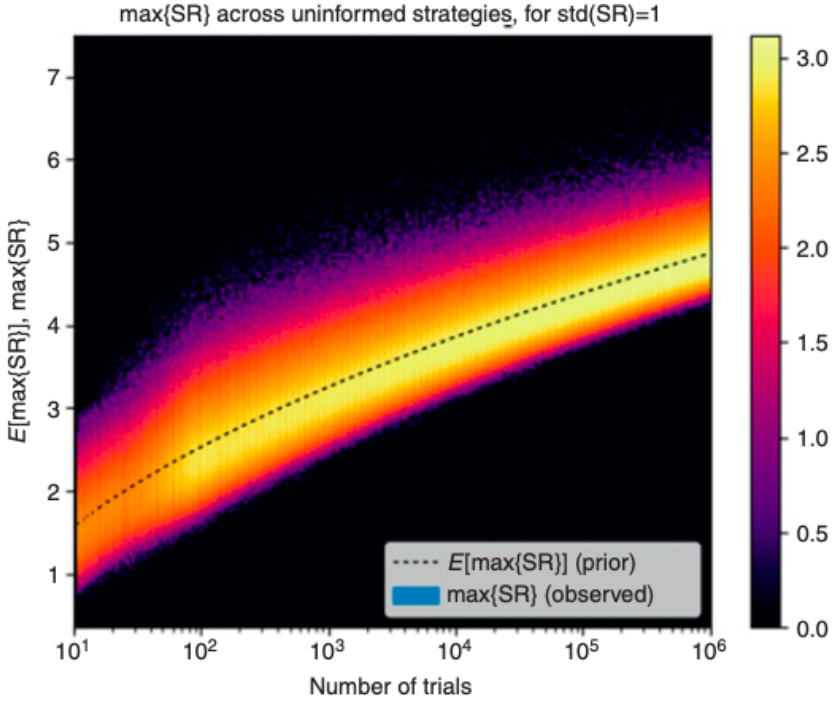


Figura 5.10: Fonteira Eficiente Sharpe Ratio Probabilístico (Retirado de DE PRADO [14])

A solução proposta por BAILEY e LÓPEZ DE PRADO [43] para o problema das múltiplas tentativas é o *Deflated Sharpe Ratio* (DSR), que estima a probabilidade do Sharpe Ratio estimado ($\hat{SR}$) ser estatisticamente significativo após controlar o efeito inflacionário das múltiplas tentativas, dos retornos não normais e do tamanho da amostra. Essencialmente, o DSR é um PSR em que o limite de rejeição (SR^*) é ajustado para refletir a multiplicidade de tentativas:

$$\widehat{DSR} \equiv \widehat{PSR}\left(\widehat{SR}_0\right) = Z\left[\frac{\left(\widehat{SR}-\widehat{SR}_0\right)}{\hat{\sigma}(\widehat{SR})}\right] = Z\left[\frac{\left(\widehat{SR}-\widehat{SR}_0\right)\sqrt{n-1}}{\sqrt{1+\frac{1}{2}\widehat{SR}^2-\gamma_3\widehat{SR}+\frac{\gamma_4-3}{4}\widehat{SR}^2}}\right] \tag{5.8}$$

Onde $\widehat{SR}_0 = E\left[\max\left\{\widehat{SR}_n\right\}\right]$ e assumindo que $E[SR] = 0$. Por fim, é necessário definir o que são tentativas independentes (N). Aqui será usada a formulação mais simples, onde M é o numero total de tentativas e $\widehat{\rho}$ é a correlação média entre essas tentativas (ver equação 5.9). Por fim, BAILEY e LÓPEZ DE PRADO [43] sugerem testar apenas $1/e$ (≈ 0.37) do total de estratégias.

$$N = \hat{\rho} + (1 - \hat{\rho})M \tag{5.9}$$

5.3.5 Configurando o Experimento

Nesta seção será feito um experimento de neutralização nos grupos de variáveis levando em consideração as premissas debatidas até aqui a fim de evitar certo tipos de erros e vieses. Como já comentado, o experimento será feito sobre a base de treino (120 eras).

Além disso, serão consideradas 4 proporções de neutralização (-0.5, 0, 1 e 1.5), em cada um dos 6 grupos, totalizando 4096 estratégias onde $\approx$ 37% delas serão selecionadas de maneira aleatória, totalizando 1506 tentativas.

Pela equação 4.1, os retornos financeiros são calculados pelo *corr_score* + *mmc_score*, logo o sharpe deve ser calculado com base na soma dessas duas métricas, representado por *corr+mmc_sharpe*. Para a base de treino, o sharpe obtido pelo modelo de referência é $SR_0 = 1.4042$ que será utilizado como valor base na equação 5.8.

O experimento obteve um $\hat{SR}^* = 1.6439$ que é maior que o valor base, o que gerou um $PSR(\hat{SR}^*) = 0.9650$. Porém de acordo com a equação 5.9, houveram 867 tentativas independentes (de um total de 1506) e consequentemente $E\left[\max\left\{\widehat{SR}_n\right\}\right] = 2.0577$ segundo a equação 5.7. Ao descontar o Sharpe pelas tentativas independentes, obteve-se um $DSR(\hat{SR}^*) = 0.00029$. Esse valor significa que há aproximadamente 0.03% de probabilidade de obter um Sharpe Ratio consistentemente superior ao modelo de referência no longo prazo.

De acordo com DE PRADO [12] era esperado um valor acima de 95%, logo deve-se **descartar todas as estratégias**, uma vez que é provável qualquer resultado excessivamente otimista seja fruto de algum tipo de sobreajuste, mesmo que $\widehat{SR}_n \gg SR_{baseline}$.

De toda forma as estratégias encontradas nesta seção serão armazenadas para uma avaliação posterior. A tabela 5.1 mostra que a estratégia selecionada na seção 5.2.2 ($SR^*(val)$), é muito diferente da estratégias selecionadas utilizando toda a base de dados (PSR^*), o que já era esperado, além disso, rejeitou estratégias de neutralização fora do intervalo $[0; 1]$, consideradas mais arriscadas.

Tabela 5.1: Estratégias Neutralização por Grupos

Grupo	SR* (val)	PSR*
intelligence	2	-0.5
wisdom	0	1
charisma	0	0
dexterity	2	1
strength	2	0
constitution	2	0

5.4 Mitigando a Volatilidade

A estratégia de neutralização por grupos falhou em encontrar um modelo que entregasse retornos positivos e suficientemente estáveis. Acredita-se que uma investigação por modelos menos voláteis pode trazer melhores resultados.

Nesta seção o foco estará em métricas que medem a volatilidade das variáveis, inspirada em algumas das métricas debatidas na seção 4.3.2. É esperado que após neutralizarmos essas métricas os retornos do modelo se tornem mais estáveis, otimizando o Sharpe Ratio.

5.4.1 Estacionariedade

Um modelo que apresenta performance estacionária é um indicativo de que a boa performance tende a continuar no futuro. A tabela 5.2 representa uma sequência de lançamentos de uma moeda honesta, onde cara (H) representa ganho e coroa (T), representa perda. Em ambos os casos há 17 períodos de ganho (+1), contra 8 de perda (-1).

Tabela 5.2: Comparativo Sequência de Lançamentos

#	Sequência
Modelo 1:	HTHHTHHHTHHHTHHHTTHTHHHHT
Modelo 2:	HHHHHTTTHHHHTTTTTHHHHHHHH

Dessa forma, ambos os modelos possuem o mesmo *Sharpe Ratio*. Contudo a sequência apresentada pelo modelo 1, com períodos de perda curtos, intercalados por períodos de ganho é preferível em relação ao modelo 2, que possui um período de perdas mais longo em sequência.

Tabela 5.3: Comparativo Sharpe

#	Sharpe
Modelo 1:	0.3858
Modelo 2:	0.3858

A autocorrelação entre as duas sequências é significativamente diferente, além disso a performance do modelo 2 não é estacionária. A confiança de uma boa performance futura pode ser visualizada na figura 5.11, onde o modelo 1 aparenta ser mais estável.

Tabela 5.4: Comparativo AR(1)

#	AR(1)
Modelo 1:	-0.2592
Modelo 2:	0.6250

Tabela 5.5: Comparativo Teste ADF

#	ADF Test (p-value)
Modelo 1:	2.2512 e-07
Modelo 2:	0.1875

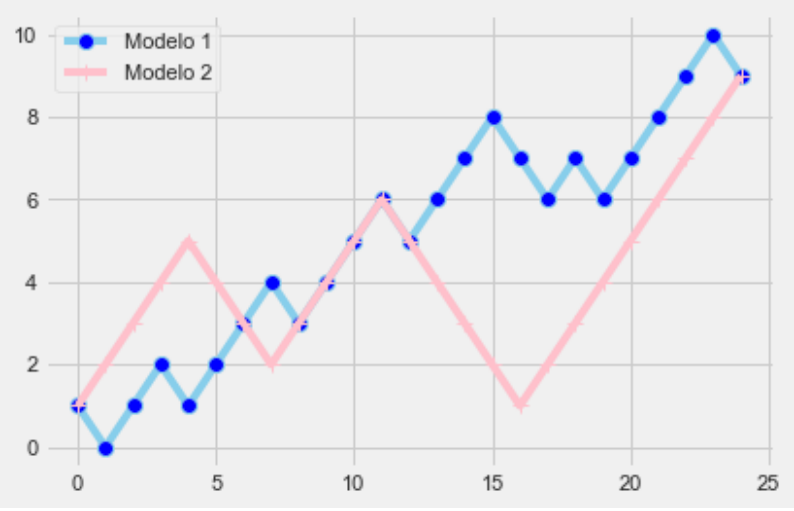


Figura 5.11: Comparativo Resultado dos Modelos

Esse resultado também pode ser medido pela métrica *max drawdown*, já comentada anteriormente, onde o modelo 1 possui um melhor resultado.

5.4.2 Métricas de Volatilidade

Nesta seção as variáveis do conjunto de dados serão observadas sobre esta perspectiva. Entende-se que modelos treinados em variáveis não estacionárias tendem

Tabela 5.6: Comparativo Drawdown

#	Max Drawdown
Modelo 1:	-2
Modelo 2:	-5

a apresentar uma performance não estacionária, isto é, funcionam bem apenas em regimes específicos. É sabido também que a componente linear do sinal tende a ser menos estável, então ao longo dessa seção essas variáveis serão neutralizadas.

310 estimadores de uma variável foram criados, onde a saída do estimador será apenas a própria variável e assim obtemos uma série temporal com a sua performance acumulada por era como visto na figura 5.12.

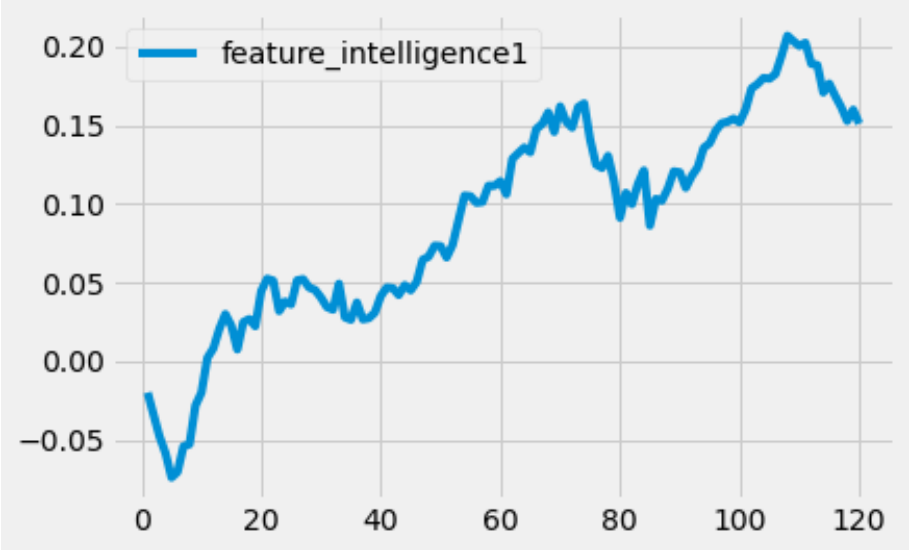


Figura 5.12: Estimador uma Variável

Inicialmente performou-se um teste de estacionariedade Dickey-Fuller Aumentado, sobre a performance desse modelos de uma variável, isto é, uma série temporal representando os scores das 120 eras. Por se tratar de um conjunto de dados normalizado, o teste ADF falhou em rejeitar a presença raiz unitária em apenas 7 das 310 variáveis (ver tabela 5.7), ou seja apenas 7 variáveis possuem performance não estacionária.

Tabela 5.7: Teste de Estacionariedade Features

#	ADF Test (p-value 5%)
Estacionárias:	303
Não estacionárias:	7

Por esse motivo, outros parâmetros relacionados a estacionariedade serão investigados para analisar as variáveis.

AR1

A autocorrelação, representada pela equação 5.10, mede a correlação da própria série com o valor imediatamente anterior. Observou-se anteriormente que valores absolutos próximos a zero, apresentam períodos de volatilidade mais curtos. A variável *Intelligence 3* apresentou AR(1)=0.2408.

$$y_i = c + \sum_{i=1}^{p} \varphi_i y_{t-1} + \varepsilon_t = \dots \tag{5.10}$$

Uma outra maneira interessante de calcular autocorrelação é substituindo a série, por uma sequência binária, onde 1 significa que está acima da média da série e zero abaixo.

Tabela 5.8: AR(1) Alternativo

era score	sign
-0.0194	0
-0.01	0
-0.0056	0
0.0243	1
0.021	1
0.0012	0
-0.0307	0
0.0484	1
-0.0273	0
-0.002	0

A série obtida pela variável *wisdom 27* possui média igual a 0.00437 e a tabela 5.8 demonstra como essa transformação foi feita, por fim chegamos a um valor de $AR(1)_{sign} = 0.2435$.

Smart Sharpe

O Smart Sharpe DE VIÉVILLE *et al.* [44], é equivalente ao sharpe, multiplicado por um termo que penaliza a autocorrelação.

$$AR\_penalty = \sqrt{1 + 2\sum_{i=1}^{n} \frac{(n-i)}{n} \rho^i} \tag{5.11}$$

Onde, $\rho = |AR(1)|$

5.4.3 Configurando o Experimento

Nesta seção será feito experimento similar ao da seção 5.3.5 com os seguintes hiperparâmetros, totalizando 100 estratégias distintas.

- **Métrica**: Desvio Padrão, Max Drawdown, AR(1), $AR(1)_{sign}$ e Smart Sharpe com $|AR(1)_{sign}|$
- **Percentual de variáveis**: [0.1, 0.2, 0.3, 0.4, 0.5], do total de variáveis, sendo a melhor ou pior dependendo da métrica selecionada
- **Proporção FN**: [0.25, 0.75. 1.0. 1.25]

O experimento obteve um $\hat{SR}^* = 1.8852$ que é superior que o valor base e ao Sharpe do experimento anterior, o que gerou um $PSR(\hat{SR}^*) = 0.9995$. Devido ao número relativamente baixo de combinações, obtivemos apenas 15 tentativas independentes, que refletiram em um $E\left[\max\left\{\widehat{SR}_n\right\}\right] = 1.9736$. e um $DSR(\hat{SR}^*) = 0.2711$, significativamente mais alto que o experimento anterior, porém ainda insuficiente para garantir que essa estratégia possui um Sharpe superior ao modelo de referência no longo prazo. A estratégia consiste em neutralizar em 100%, 40% das variáveis ordenadas pelo pior Smart Sharpe.

5.5 Resumo do Capítulo

Ao longo deste capítulo o conceito de neutralização foi introduzido e foi desenvolvida uma metodologia para avaliação de estratégias. Foram conduzidos dois experimentos para avaliarmos um conjunto de estratégias, um focado nos grupos de variáveis e outro que utiliza métricas de volatilidade como critério de seleção. Por fim além do modelo de referência, da regressão linear e do modelo de referência neutralizado, há 3 novos modelos que serão avaliados no capítulo 9. São eles:

1. **SR* (val)** criado na seção 5.2.2
2. **PSR* (grupos)** criado na seção 5.3.5
3. **PSR* (vol)** criado na seção 5.4.3

Por fim, através da figura 5.13, observe o diagnóstico dos 6 modelos desenvolvidos até aqui. Não é surpresa que o modelo *sr_val* tenha obtido o melhor resultado na maioria das métricas nesta base pelos motivos citados na seção 5.2.2. Por outro lado observe que os modelos *psr_group* e *psr_vol* obtiveram um sharpe superior ao modelo de referência, como desejado.

	ex_preds	lr	ex_FN100	sr_val	psr_group	psr_vol
Validation_Sharpe	0.8861	0.5461	1.0475	1.4416	0.9849	1.0437
Validation_Mean	0.0241	0.0163	0.0207	0.0254	0.0208	0.0245
Feat_neutral_mean	0.0182	0.0030	0.0198	0.0177	0.0179	0.0180
Validation_SD	0.0272	0.0299	0.0198	0.0176	0.0211	0.0234
Feat_exp_max	0.2666	0.2784	0.0140	0.2077	0.2692	0.2064
Max_Drawdown	-0.0353	-0.0853	-0.0217	-0.0042	-0.0236	-0.0199
corr_plus_mmc_sharpe	0.8861	0.3629	0.8111	1.3725	0.8656	1.0627
val_mmc_mean	0.0000	-0.0008	0.0046	0.0107	-0.0004	0.0021
corr_with_example_preds	1.0000	0.6550	0.6063	0.4677	0.8748	0.8904

Figura 5.13: Diagnóstico Final

Capítulo 6

Seleção de Variáveis

Ao longo deste capítulo serão implementadas diferentes técnicas de seleção e agrupamento de variáveis. Na sequência serão estudar maneiras de reduzir o ruído da matriz de correlação de forma a aprimorar os métodos de seleção e por fim será testada uma métrica robusta a relacionamentos não lineares.

6.1 Introdução

Ao longo deste capítulo, serão desenvolvidas a diversas técnicas de seleção de variáveis a fim de principalmente remover variáveis que de alguma maneira prejudicam o estimador (modelo de referência). É importante salientar que não serão realizados diagnósticos das técnicas citadas até que todo o experimento seja finalizado e assim mitigar o risco de sobreajuste. Dessa forma serão implementadas outras maneiras de comparar as técnicas apresentadas.

De acordo com DE PRADO [14] o método mais comum da estatística clássica se baseia em um teste de hipótese (teste de t-student), que busca confirmar a hipótese do coeficiente linear da referida váriável seja zero ($H0 : coef = 0$), caso contrário a variável possui algum sinal a ser utilizado pelo estimador [45]. A equação 6.1 como o teste é configurado.

$$\begin{cases} H0 : coef = 0 \\ H1 : coef \neq 0 \end{cases} \tag{6.1}$$

Para realizar este teste, utilizou-se apenas as 14 variáveis do grupo *dexterity*, além disso foram geradas 7 variáveis de ruído (*noise*) permutando as 7 primeiras variáveis, totalizando então 21 variáveis originais. A figura 6.1 mostra o resumo de uma regressão linear ajustada nestes dados. A coluna $Pr > |t|$ é referente ao p-valor, que quantifica a probabilidade do verdadeiro coeficiente ser zero.

Note que para ao menos três variáveis descartamos a hipótese nula, porém ne-

nhuma delas pertence ao grupo de variáveis ruidosas, o que mostra que este teste não possui boa aderência com o este conjunto de dados.

OLS Regression Results

Dep. Variable:	target	R-squared (uncentered):	0.788
Model:	OLS	Adj. R-squared (uncentered):	0.788
Method:	Least Squares	F-statistic:	8.903e+04
Date:	Wed, 15 Dec 2021	Prob (F-statistic):	0.00
Time:	04:24:59	Log-Likelihood:	-20156.
No. Observations:	501808	AIC:	4.035e+04
Df Residuals:	501787	BIC:	4.059e+04
Df Model:	21		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
feature_dexterity1	0.0739	0.003	26.743	0.000	0.069	0.079
feature_dexterity2	-0.0488	0.003	-15.015	0.000	-0.055	-0.042
feature_dexterity3	-0.0002	0.003	-0.067	0.946	-0.006	0.005
feature_dexterity4	0.0166	0.003	5.951	0.000	0.011	0.022
feature_dexterity5	-0.0118	0.003	-3.893	0.000	-0.018	-0.006
feature_dexterity6	0.0138	0.003	5.170	0.000	0.009	0.019
feature_dexterity7	0.0349	0.003	12.578	0.000	0.029	0.040
feature_dexterity8	0.0029	0.003	0.998	0.318	-0.003	0.009
feature_dexterity9	0.0659	0.002	30.839	0.000	0.062	0.070
feature_dexterity10	-0.0407	0.003	-13.141	0.000	-0.047	-0.035
feature_dexterity11	0.0381	0.003	14.458	0.000	0.033	0.043
feature_dexterity12	0.0354	0.003	13.042	0.000	0.030	0.041
feature_dexterity13	-0.0034	0.003	-1.107	0.268	-0.009	0.003
feature_dexterity14	0.0148	0.003	5.328	0.000	0.009	0.020
noise_1	0.1068	0.001	112.096	0.000	0.105	0.109
noise_2	0.1084	0.001	113.773	0.000	0.107	0.110
noise_3	0.1082	0.001	113.775	0.000	0.106	0.110
noise_4	0.1073	0.001	112.838	0.000	0.105	0.109
noise_5	0.1079	0.001	113.478	0.000	0.106	0.110
noise_6	0.1079	0.001	113.533	0.000	0.106	0.110
noise_7	0.1081	0.001	113.726	0.000	0.106	0.110

Omnibus:	228.776	Durbin-Watson:	1.978
Prob(Omnibus):	0.000	Jarque-Bera (JB):	244.874
Skew:	0.027	Prob(JB):	6.70e-54
Kurtosis:	3.094	Cond. No.	31.4

Figura 6.1: Teste de Hipótese

Deve-se lembrar que há uma grande quantidade de ruído nos dados e mesmo uma variável aleatória pode apresentar um comportamento similar ao de uma variável com sinal, como visto na figura 6.2.

Além disso DE PRADO [14] cita ao menos quatro limitações para esse tipo de teste:

1. **Suposições fortes** como independência e resíduos em ruído branco
2. Na presença de **multicolinearidade** o p-valor não pode ser precisamente estimado
3. Estima a **probabilidade** de ser igual ou mais extremo (**pouca utilidade**)
4. É uma estimativa **dentro da amostra** (*in-sample*), que pode não se repetir

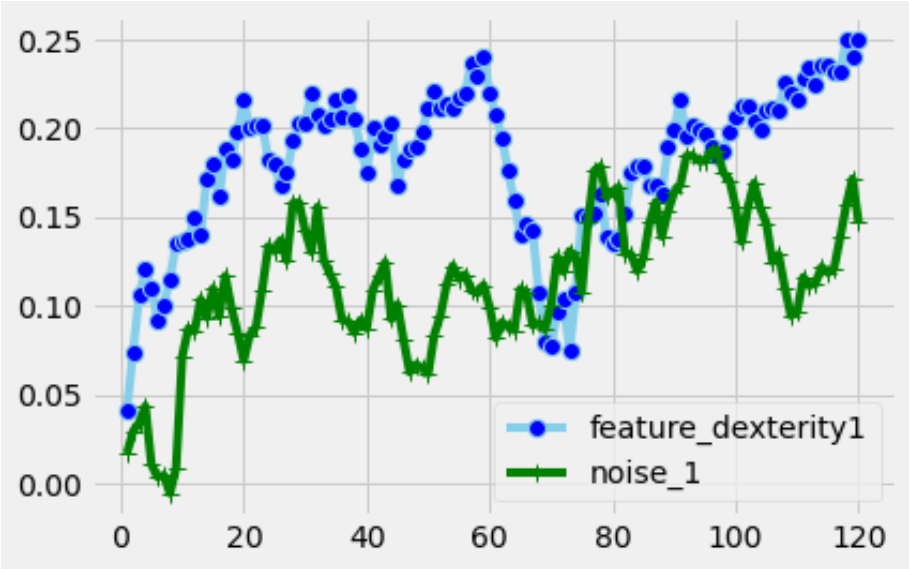


Figura 6.2: Performance Variável com Permutação

6.2 Métodos de Seleção de Variáveis

Nesta seção serão desenvolvidos alguns métodos básicos de seleção de variáveis além de uma comparação objetivas entre esses métodos.

6.2.1 Redução de Impureza

O *Mean Decrease Impurity* (MDI) é um método específico de modelos de árvore, como o *Random Forest* (ver seção 2.2.2.2). Em cada nó de cada árvore um subconjunto de variáveis é sorteada e aquela onde melhor ponto de corte produzir uma divisão mais pura é selecionada. Esse critério de pureza pode ser o Índice de Gini, Entropia, Ganho de informação, entre outros. O MDI representa a média do decréscimo de impureza atingido pelo ponto de corte a cada vez que a variável é selecionada [12].

É possível desta forma entender que a variável selecionada com maior frequência também é a que melhor reparte os dados. Contudo, segundo PEDREGOSA *et al.* [7], esse método possui alguns problemas, uma vez que comumente uma variável de maior **cardinalidade** possui maior probabilidade de dividir melhor os dados.

Felizmente não há esse problema no atual conjunto de dados, onde todas as variáveis possuem cinco valores únicos. De toda forma, seguiremos o método proposto por DE PRADO [12], que sugere permitir uma única variável por sorteio, garantindo assim que todas as variáveis sejam selecionadas igualmente. Além disso o mesmo autor algumas limitações para este método como:

1. É uma estimativa **dentro da amostra** (*in-sample*)

2. Não é robusto **multicolinearidade**, variáveis idênticas terão sua importância dividida pela metade

Experimento MDI

Note que as variáveis de ruído obtiveram resultado superior as demais variáveis, logo o experimento não funcionou e essa técnica será descartada.

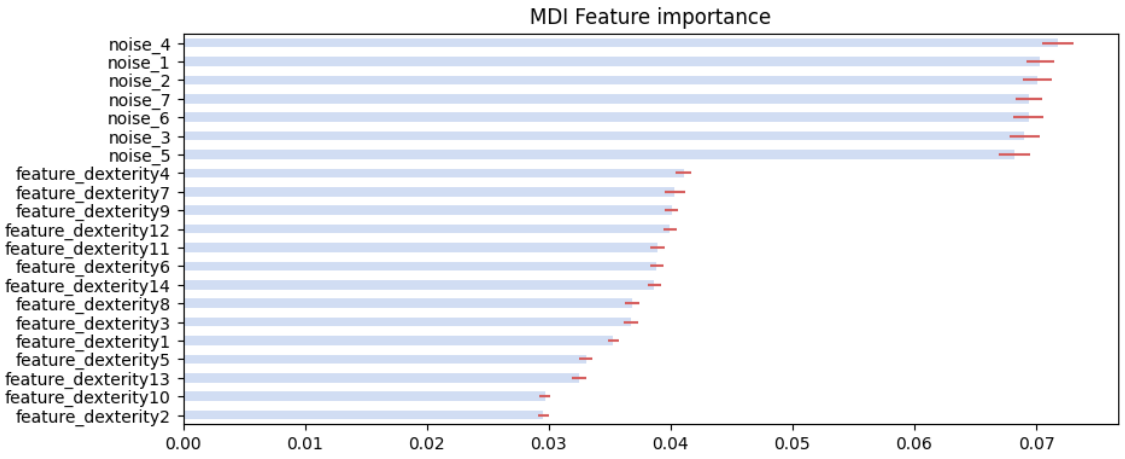


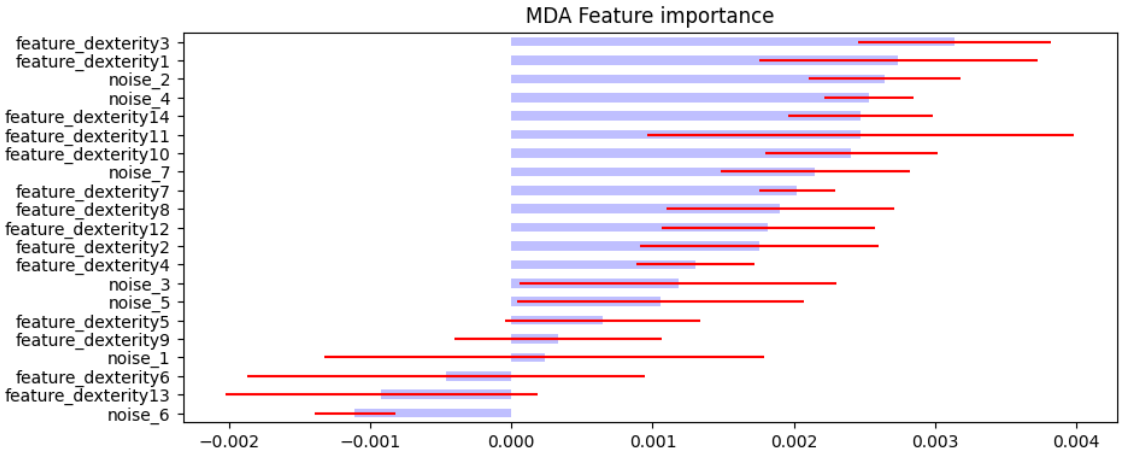
Figura 6.3: MDI - Random Forest

6.2.2 Permutação de Variáveis

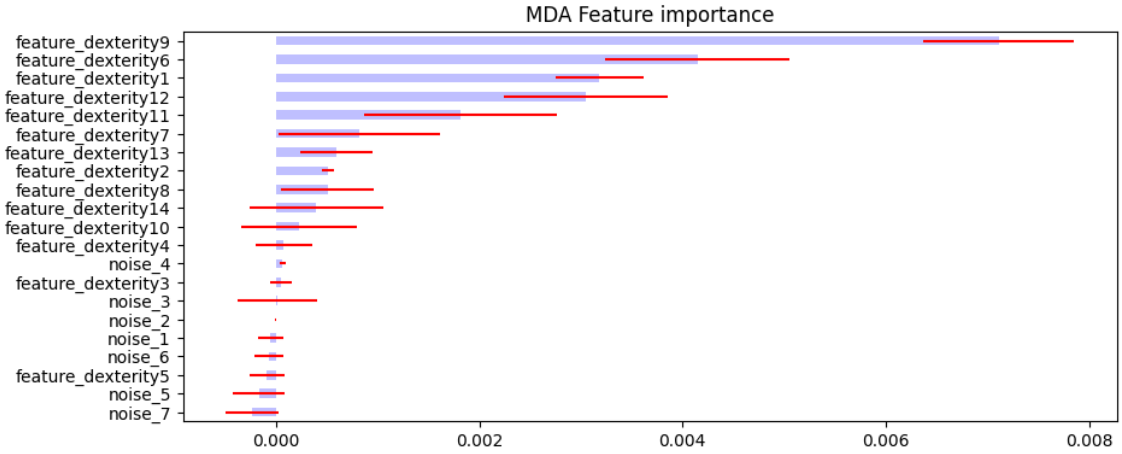
O *Mean Decrease Accuracy* (MDA), calcula a perda média de performance (podendo ser acurácia ou outra métrica) ao permutarmos cada variável em comparação a performance original. Este método é significativamente mais lento que o anterior porque é calculado usando validação cruzada (ver seção 2.5.1), porém é uma estimativa fora da amostra (*out-of-sample*). Contudo este método também não é robusto a multicolinearidade e caso haja duas variáveis idênticas ambas serão classificadas como sem importância [12].

Experimento MDA

O MDA pode ser utilizado em conjunto com qualquer modelo de aprendizado de máquina. Por essa razão utilizou-se o mesmo modelo *Random Forest* do experimento anterior e uma regressão linear. Perceba que o *Random Forest* na figura 6.4a fez um trabalho razoável ranqueando as variáveis originais acima das variáveis de ruído, contudo a regressão linear na figura 6.4b fez um trabalho ainda melhor.



(a) MDA - Random Forest



(b) MDA - Regressão Linear

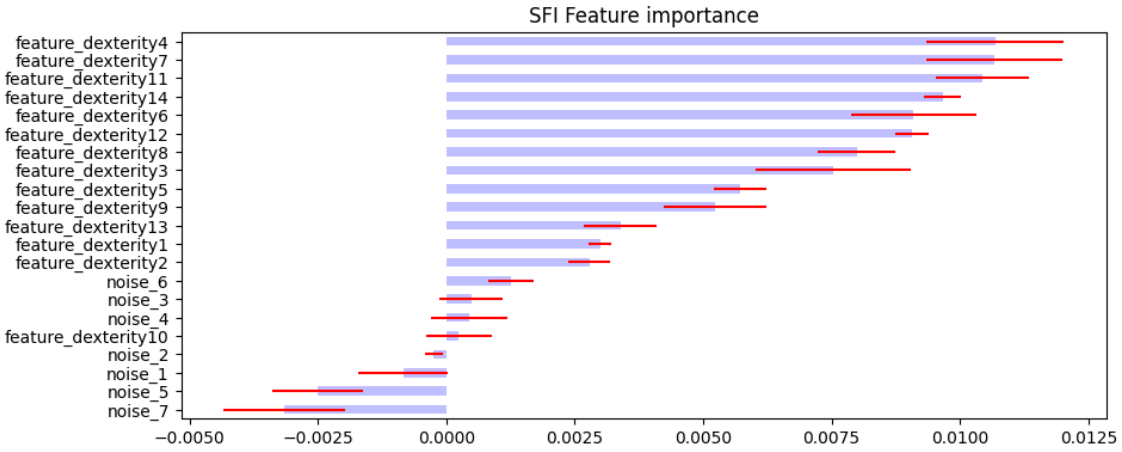
Figura 6.4: Resultados MDA

6.2.3 Modelos de Uma Variável

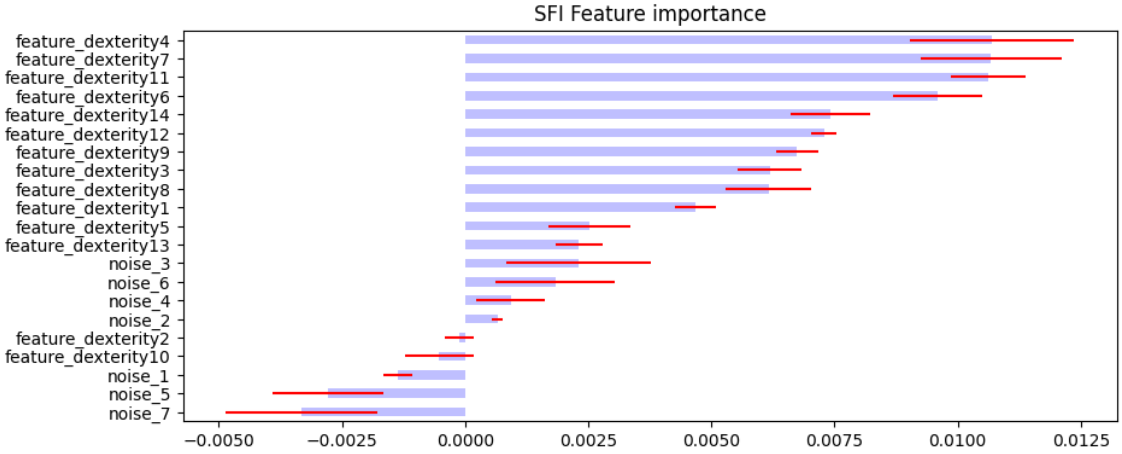
O método *Single Feature Importance* (SFI) é o mais simples de todos uma vez que cria um estimador com apenas uma variável e assim como o MDA, as variáveis com importância negativa devem ser descartados. Note que este método é robusto a multicolinearidade, porém desconsideramos as interações por pares, debatido na seção 4.5.

Experimento SFI

Repetindo o experimento do exemplo anterior, note que ambos os modelos obtiveram boa performance em relação as variáveis de ruído, além disso, note que ambos concordaram no ranqueamento da maior parte das variáveis originais.



(a) SFI - Random Forest



(b) SFI - Regressão Linear

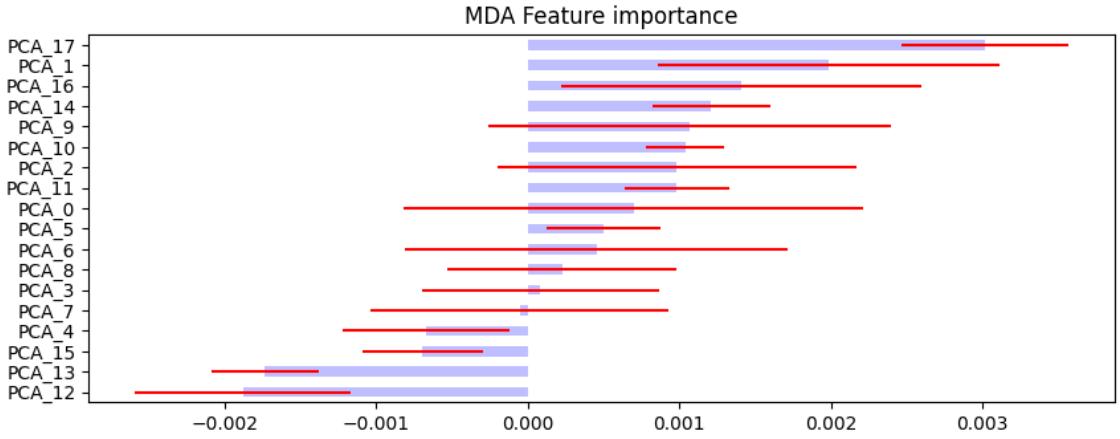
Figura 6.5: Resultados SFI

6.2.4 Ortogonalização

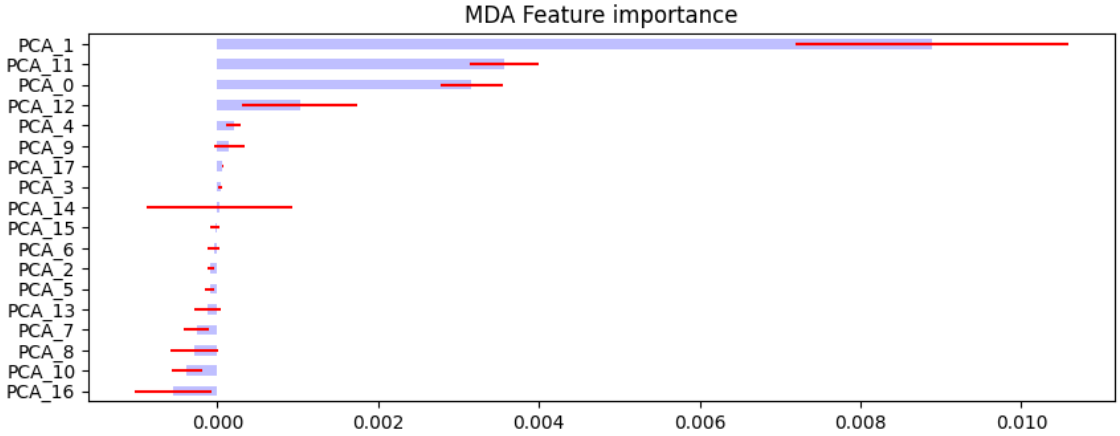
Ao ortogonalizarmos as variáveis utilizando análise dos componentes principais (PCA), descrito na seção 2.4.1. Lidamos parcialmente com a multicolinearidade, pois aliviamos os efeitos de substituição **lineares**.

Experimento PCA

Repetiremos o experimento realizado com o MDA, porém com as variáveis ortogonalizadas. Note que a regressão linear fez um trabalho melhor ranqueando as primeiras componentes mais acima.



(a) PCA+MDA - Random Forest



(b) PCA+MDA - Regressão Linear

Figura 6.6: Resultados PCA+MDA

6.2.5 Comparando Métodos

DE PRADO [12] cita que deve-se ser sempre cético a respeito de variáveis supostamente importantes identificada por qualquer método. Agora lembre-se que ao extrair as componentes principais, isso foi feito de maneira não supervisionada (ver seção 2.3), ao contrário dos métodos citados anteriormente que são supervisionados (ver seção 2.1). Quando algum método supervisionado (MDI, MDA, SFI), seleciona as mesmas variáveis que o PCA determinou como as principais de maneira não supervisionada, temos uma **evidência** que o padrão encontrado pelo método supervisionado não é sobreajuste.

Essa comparação será feita analisando a correlação das importâncias obtidas por um dos métodos supervisionados com os autovalores associados a essas variáveis. Para tal, multiplicaremos os N autovalores (Λ_N) pelos M autovetores (W_M) e obtém-se o módulo desta operação, ou seja $|(\Lambda \times W)|$, resultando em uma matrix P^{NxM}. A matrix P será então transformada em um vetor único Γ de comprimento $N \cdot M$. Este vetor será comparado com um outro vetor Υ contendo as importâncias de cada

uma das N variáveis encontradas pelo método de seleção repetidas M vezes. Por fim deve-se calcular a correlação entre os vetores Γ e Υ. A figura 6.7 ilustra o gráfico de dispersão em escala log do vetor com os autovalores (Γ) com o vetor das importâncias das variáveis (Υ) obtidas pelo método MDA com regressão linear.

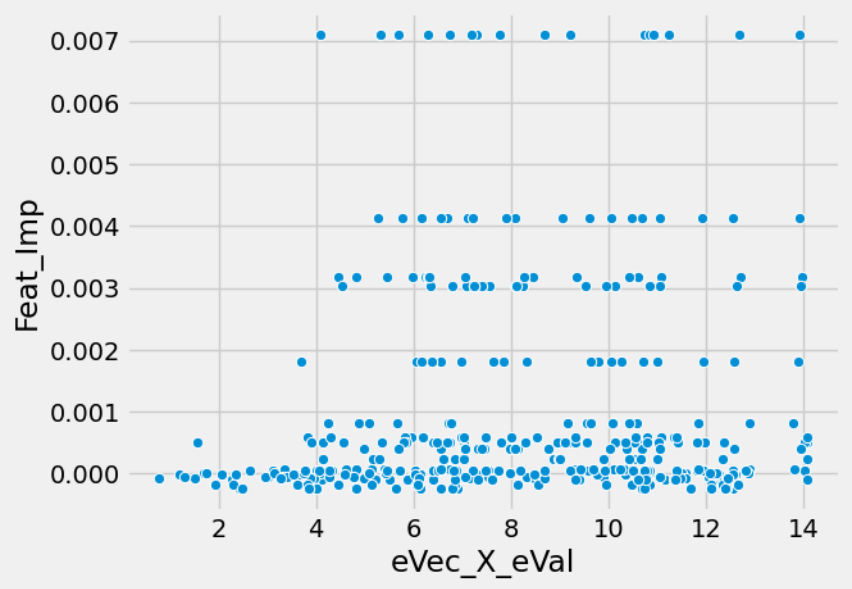


Figura 6.7: Importâncias x Autovalores

Perceba que o grande acúmulo de pontos na região inferior do gráfico se referem aos autovalores associados as variáveis de ruído. Certamente a presença desses pontos tornará qualquer medida de correlação pouco acurada, de fato obtivemos uma correlação de Spearman de apenas 0.1161. Uma forma de desconsiderar esses pontos provenientes de ruído é utilizarmos a correlação de postos de Kendall Tau, que é similar a correlação de Spearman, a diferença está no cálculo do coeficiente (τ), que leva em consideração o ranking relativo de cada par visto na equação 6.2.

$$\tau = \frac{(\#pares\_concordantes) - (\#pares\_disconcordantes)}{\binom{n}{2}} \tag{6.2}$$

A figura 6.8 auxilia no entendimento de pares concordantes e discordantes.

Nesse contexto, a vantagem de utilizar este coeficiente de correlação é a possibilidade de calcular o coeficiente (τ), de maneira ponderada, dando um peso para cada par avaliado, ou seja as variáveis de maior importância possuem maior peso, assim não precisamos nos preocupar com a ordenação de variáveis de menor importância. Use a equação 6.3 para encontrar o peso relativo para cada par de amostras, onde r e s, são os rankings relativos de cada variável.

$$Weight = \frac{1}{(r+1)} + \frac{1}{(s+1)} \tag{6.3}$$

Por fim, em conjunto das importâncias, deve-se utilizar o ranking da projeção

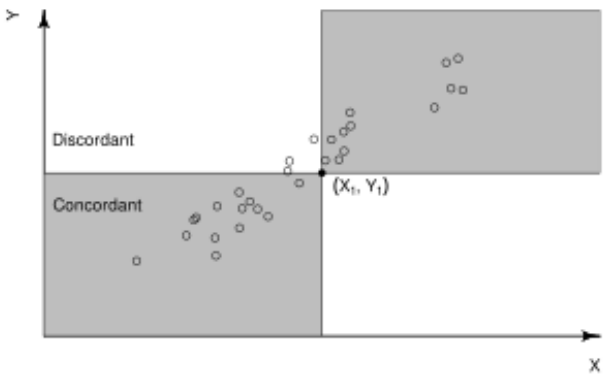


Figura 6.8: Correlação de Kendall (τ) (Retirado de SALKIND [15])

das importâncias obtidas pelos PCA. Esse ranking é obtido pela soma do módulo da multiplicação entre os autovalores e autovetores, que pode ser entendido com a soma das distâncias de cada componente na variável original como visto na equação 6.4.

$$PCA_{rank} = rank\left(\sum_{w} |(\Lambda \cdot W)|\right) \tag{6.4}$$

Resultados

A tabela 6.1 mostra o resultado do comparativo para o experimento que desenvolvemos ao longo desta seção. Perceba que em relação a correlação de Spearman, nenhum dos métodos obteve bom resultado, porém utilizando a correlação de Kendall ponderada o método MDA com regressão linear obteve bom resultado. Esse resultado vai de encontro com os bons resultados obtidos por essa técnica nas seções 6.2.2 e 6.2.4. Note também que o MDI obteve uma correlação altamente negativa, exatamente como o visto no experimento da seção 6.2.1.

Tabela 6.1: Resultado Experimento

#	Spearman	Weighted Kendall Rank
mda_linear_reg	0.1161	0.5890
sfi_rf	0.1125	0.2078
sfi_linear_reg	0.1033	0.1905
mda_rf	0.0162	-0.0148
mdi_rf	-0.1023	-0.5431

Agora repetindo o mesmo experimento desenvolvido até aqui considerando todas as 310 variáveis, além disso utilizou-se o modelo de referência (xgb) e a regressão

linear (*linear_reg*). Note que os valores obtidos são significativamente menores, o que é esperado, uma vez que é mais difícil obter uma correlação alta com um par de vetores maior. Note também que o MDA com regressão linear obteve novamente o melhor resultado, dando uma nova evidência de que essa é a melhor técnica entre as testadas até aqui para esse conjunto de dados.

Tabela 6.2: Resultado Todas as Variáveis

#	Spearman	Weighted Kendall Rank
mda_linear_reg	0.0087	0.1613
sfi_linear_reg	-0.0087	0.0823
mda_xgb	0.0081	0.0610
sfi_xgb	-0.0146	-0.0319

A figura 6.9 ilustra um agrupamento hierárquico (ver seção 2.4.3) da matriz de correlação das importâncias encontradas por cada método de seleção, note que o método escolhido (MDA com Regressão Linear) além de ser pouco correlacionado com os outros métodos, foi agrupado de maneira distinta dos outros métodos, indicando que este encontrou um sinal diferente dos outros métodos.

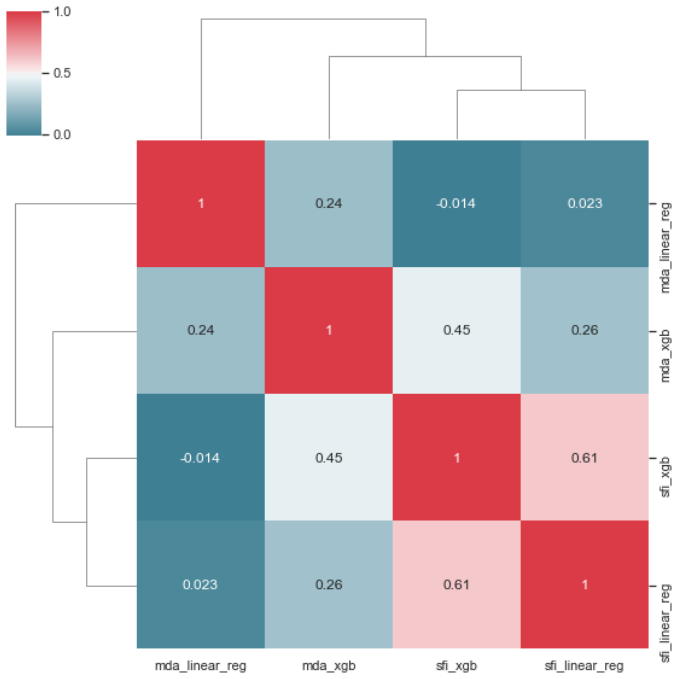


Figura 6.9: Agrupamento Hierárquico Métodos Seleção

Como se trata de uma regressão linear, optou-se por performar novamente uma neutralização das variáveis selecionadas. Por fim, observe que o método escolhido selecionou uma proporção razoavelmente parecida de variáveis em cada grupo.

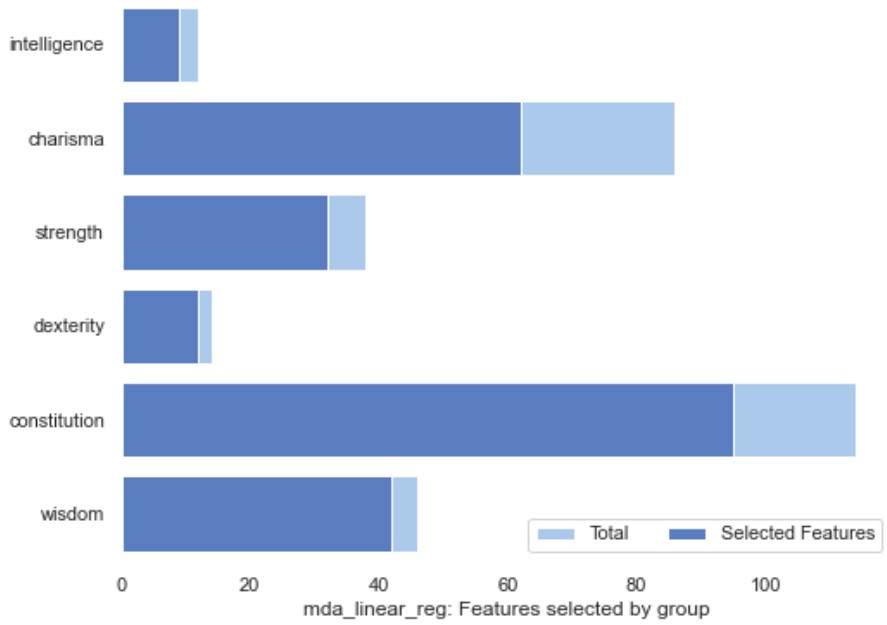


Figura 6.10: Variáveis Selecionadas

6.3 Agrupamento de Variáveis

Além dos métodos citados anteriormente, existem outros mais robustos como *SHAP* [46] e o *EBM* [47], porém ambos, assim como os outros analisam as variáveis individualmente. Nesta seção pretende-se aliviar os efeitos da multicolinearidade sem uma mudança de base (PCA), agrupando as variáveis antes de aplicarmos qualquer um dos métodos. Dessa maneira os efeitos das interações por pares serão mantidos.

6.3.1 Obtendo a Matriz de Distâncias

O primeiro passo é calcular a matriz de correlação (ρ) das variáveis de entrada, porém tenha em mente que correlação não é uma métrica, pois não satisfaz as condições de não negatividade e desigualdade triangular ($|a + b| \leq |a| + |b|$). Logo DE PRADO [14] sugere transformar a matriz de correlação em uma matriz de distâncias ($d_\rho[X, Y]$).

$$d_\rho[X, Y] = \sqrt{1/2(1 - \rho[X, Y])} \tag{6.5}$$

Observe a figura 6.11 e note que além das variáveis de ruído se encontrarem significativamente distante das outras variáveis, a matriz também indicou que as variáveis *dexterity_1* e *dexterity_2*, estão também distantes das demais variáveis do grupo, porém próximas entre si, indicando que podem sem agrupadas.

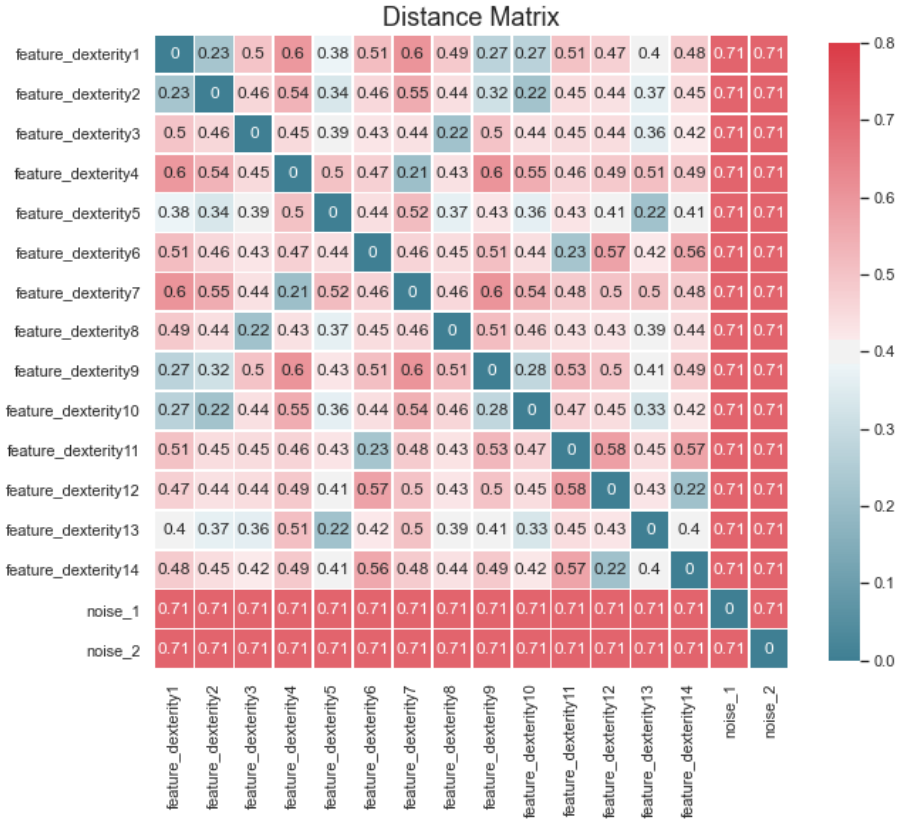


Figura 6.11: Matriz de Distâncias

6.3.2 Agrupando as Variáveis

Nesta seção será aplicado um algoritmo que irá determinar o número de agrupamentos existentes no conjunto de dados e organizar as variáveis nestes agrupamentos. O algoritmo k-médias, é o algoritmo de partição mais conhecido (ver seção 2.4.2). Porém esse algoritmo sofre de duas limitações, primeiro o usuário precisa indicar arbitrariamente o número de K agrupamentos desejado. Um método conhecido e bastante rudimentar para este fim é o método do cotovelo DE PRADO [14]. Outra limitação é a inicialização aleatória do centroide dos K agrupamentos, o que facilmente leva a soluções sub ótimas. [14] sugere utilizar o algoritmo *Optimal Number of Clusters* (ONC) que supera essas limitações.

O ONC nada mais é que uma extensão do próprio k-médias onde o número de agrupamentos é determinado por uma função objetivo (q) determinada pela métrica de silhueta (S), dada pela equação 6.6 e muito comum para avaliar algoritmos de partição [48].

$$S_i = \frac{b_i - a_i}{max\{a_i, b_i\}}; i = 1, ..., N \tag{6.6}$$

Onde a_i é a distância média da i-ésima amostra e todas as outras amostras do mesmo agrupamento e b_i é a distância média entre a i-ésima amostra e todos os

elementos do agrupamento mais próximo ao qual i não faz parte. $S_i \approx 1$ significa que a amostra está bem agrupada e $S_i \approx -1$ o oposto. Finalmente, a função de qualidade (q) dada pela equação 6.7 onde $E[\{S_i\}]$ é a média e $V[\{S_i\}]$ a variância da silhueta das amostras contidas no agrupamento.

$$q = \frac{E[\{S_i\}]}{\sqrt{V[\{S_i\}]}} \tag{6.7}$$

Com a função de qualidade em mãos deve-se performar a primeira etapa do ONC em duas etapas como se segue:

1. Crie um k-médias para K=2,...N. Armazenando a qualidade (q) a cada tentativa

2. Repita o passo 1 M vezes (DE PRADO [14] sugere 10 vezes)

Note que o primeiro passo testa todos os valores de K até o número N de amostras. O segundo passo é feito para superar os problemas causados pela inicialização aleatória, ao final desta etapa teremos o número (K^*) de agrupamentos bem como sua composição. A terceira e última modificação remete a qualidade dos agrupamentos encontrados, para tal deve-se calcular a qualidade (q_k) de cada agrupamento. Onde para cada agrupamento (q_k) com qualidade abaixo da média ($\bar{q}$) $[q_k < \bar{q}, \forall k \in K]$, é necessário performar um novo algoritmo k-médias (i.e. desde a primeira etapa) com todos os elementos destes agrupamentos rejeitados, até que hajam no máximo 2 agrupamentos.

Experimento ONC

Para este experimento serão utilizadas as variáveis do grupo *strenght*, além de seis variáveis de ruído. O primeiro passo do experimento é calcular a matriz de distâncias, em seguida aplica-se o algoritmo ONC. A matriz de correlação (figura 6.12) agora possui índices ordenados pelos agrupamentos encontrados, note que na maioria dos casos os agrupamentos possuem duas variáveis com silhueta ($S_i \geq 0.7$). Existe ainda um agrupamento com 4 variáveis e $S_i \approx 0.4$ e finalmente as variáveis de ruído num único agrupamento e $S_i \approx 0.1$.

No último passo performa-se a técnica MDA (ou outra) apresentada na seção 6.2.2 permutando um agrupamento por vez, note que as variáveis do mesmo agrupamento possuem a mesma importância. O ONC tem a vantagem de favorecer um grande número de agrupamentos, com poucos elementos em cada um. Esse comportamento é bastante desejável já que um agrupamento com um número maior de variáveis provavelmente seria classificado como grande de grande importância pelo MDA, exatamente como visto na figura 6.13. Por fim, note também que as variáveis de ruído foram classificadas como sem importância.

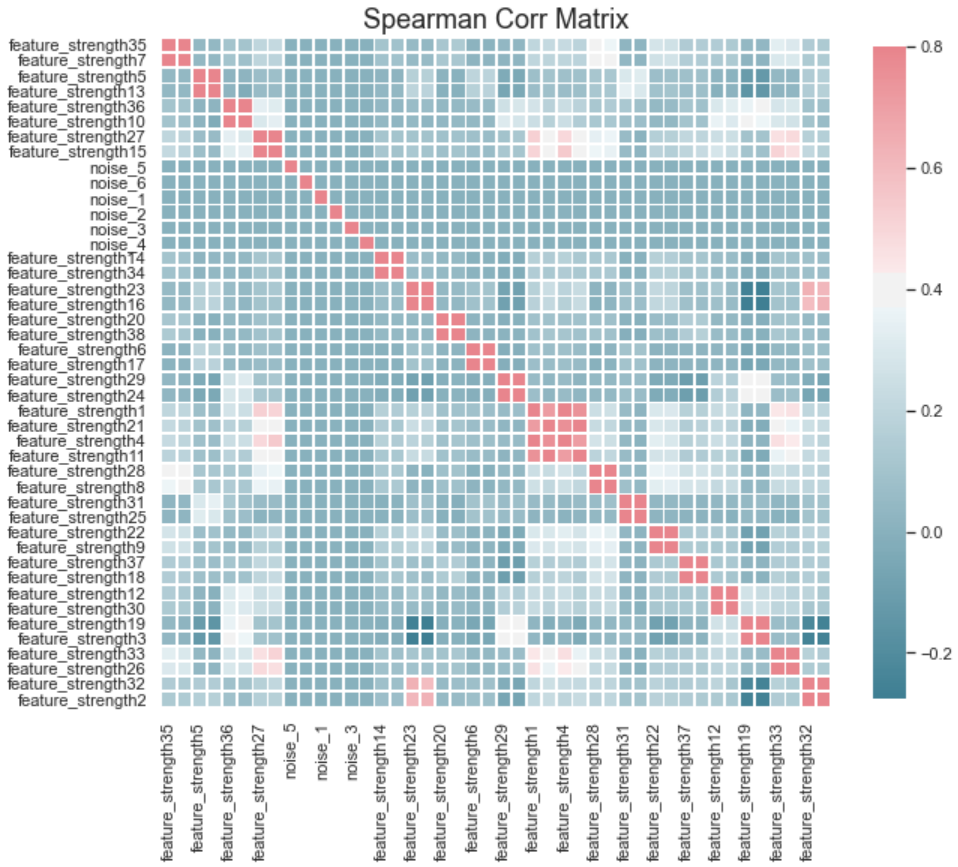


Figura 6.12: Matriz de Correlação Agrupada

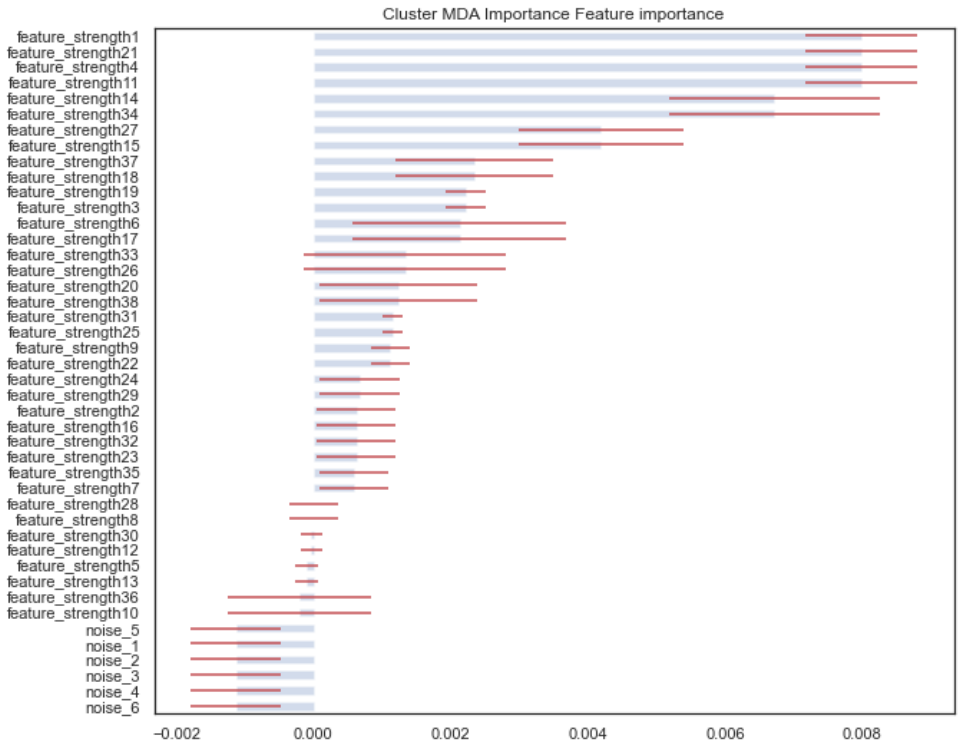
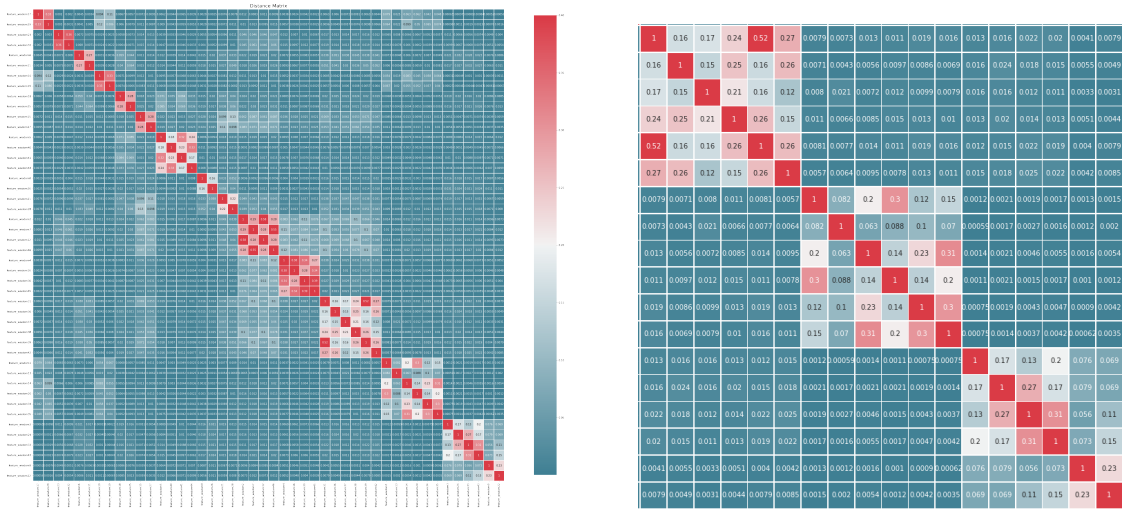


Figura 6.13: Seleção de Variáveis Agrupadas

6.3.2.1 Extensão Algoritmo ONC

A figura 6.14a mostra o resultado do mesmo experimento com as variáveis do grupo *wisdom* com 46 variáveis, um pouco maior que o anterior. Observe que os agrupamentos estão ordenados ao longo da diagonal principal na ordem que foram formados iterativamente pelo algoritmo ONC. Dessa forma os agrupamentos localizados na parte debaixo foram os últimos a serem formados e por isso foram destacados para melhor visualização na figura 6.14b. Esses agrupamentos possuem uma qualidade inferior aos encontrados no início e suspeita-se que estes foram formados devido a uma queda na qualidade média ($\bar{q}$) dos agrupamentos restantes e assim passaram a permitir agrupamentos de menor qualidade. Outro motivo é devido ao critério de parada, onde são necessários ao menos 2 agrupamentos de baixa qualidade para haver uma nova iteração.

Por esse motivo, o autor deste estudo propõe uma pequena extensão ao algoritmo ONC, permitindo ao usuário desagrupar os N últimos agrupamentos encontrados de tamanho maior que M. No exemplo da figura 6.14a, foram encontrados 15 agrupamentos ao todo e deseja-se desagrupar os últimos 4 agrupamentos ($\approx 30\%$ do total) maiores que 2 elementos. Dessa forma os agrupamentos 12, 13 e 14 de tamanhos 6, 5 e 4 respectivamente serão desagrupados, ou deixados em um agrupamento de apenas uma variável. O agrupamento 15 com apenas 2 variáveis será mantido intacto. Os parâmetros referentes a quantidade de agrupamentos a serem analisados, bem como o tamanho mínimo serão definidos pelo usuário.



(a) Todos os Agrupamentos (b) Últimos Agrupamentos

Figura 6.14: Matriz de Correlação Pós Agrupamentos

A razão principal de realizar esta extensão é que agrupamentos maiores tendem a inflacionar a sua respectiva importância ao realizarmos a permutação do agrupamento via algoritmo MDA, vide figura 6.13 onde o maior agrupamento recebeu a

maior importância entre as variáveis. Uma vez que verificou-se que estes agrupamentos não possuem boa qualidade, acredita-se que basta uma das variáveis pertencentes ao agrupamento seja boa para que esta superestime a importância das outras, sendo as remanescentes de boa qualidade ou não. Desta forma ao desagrupar todas elas ainda há a possibilidade de avaliar positivamente as boas variáveis e identificar se há de fato alguma variável de ruído.

6.3.3 Remoção de Ruído

Repetiu-se por diversas vezes ao longo deste estudo que há uma grande quantidade de ruído contida nos dados. Isso acontece devido ao próprio processo gerador dos dados, onde o mercado gera uma grande quantidade de movimentos aleatórios (*random walk*), além disso, precisaríamos confiar que não há falhas na captura e medição deste conjunto de dados. Esse ruído certamente estará presente na matriz de correlação e se não for tratado adequadamente, impactará nos resultados que obtermos a partir da mesma [14]. Logo nesta seção iremos tratar de como remover o ruído da matriz de correlação.

Considere uma matriz de entrada i.i.d. com **observações aleatórias** (X) com média igual a zero, variância (σ^2) de dimensões TxN. Os autovalores (λ) da matriz $C = T^{-1}X'X$ convergem de maneira assintótica com o crescimento da matriz de observações ($N \to \infty, T \to \infty, 1 < T/N < +\infty$) para a função densidade de probabilidade (PDF) da distibuição de Marcenko-Pastur [14]. A função de distribuição dos autovalores ($f[\lambda]$) e dada pela equação 6.8.

$$\begin{cases} \frac{T}{N}\frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi\lambda\sigma^2} & se \lambda \in [\lambda_-, \lambda_+] \\ 0 & se \lambda \notin [\lambda_-, \lambda_+] \end{cases} \tag{6.8}$$

Onde o intervalo dos valores esperados da distribuição dos autovalores $[\lambda_-, \lambda_+]$ é igual a $\sigma^2(1 \pm \sqrt{N/T})^2$. Quando $\sigma^2 = 1$, C é a matriz de correlação da matriz de entrada X [49]. Autovalores (λ) contidos no intervalo $[0, \lambda_+]$ são compatíveis com um comportamento aleatório e o contrário para autovalores fora deste intervalo.

A figura 6.15 mostra o histograma dos autovalores observados além da distribuição de Marcenko-Pastur ajustada a partir da matriz empírica, onde nem todos os autovalores possuem comportamento aleatório, o que explica a presença de valores fora do intervalo. Para que possamos distinguir os autovalores não aleatórios, precisamos determinar o ponto de corte (λ_+) que limita superiormente o intervalo.

De acordo com LALOUX *et al.* [50] uma vez que parte da variância é explicada por autovetores aleatórios, é possível estimar σ^2 de acordo com $\sigma^2(1 \pm \sqrt{N/T})^2$. Assumindo que o autovetor associado ao maior autovalor não é aleatório, deve então substituir σ^2 por $\sigma^2\left(1 - \frac{\lambda_+}{N}\right)$ o que responderá pela parcela da variância que é

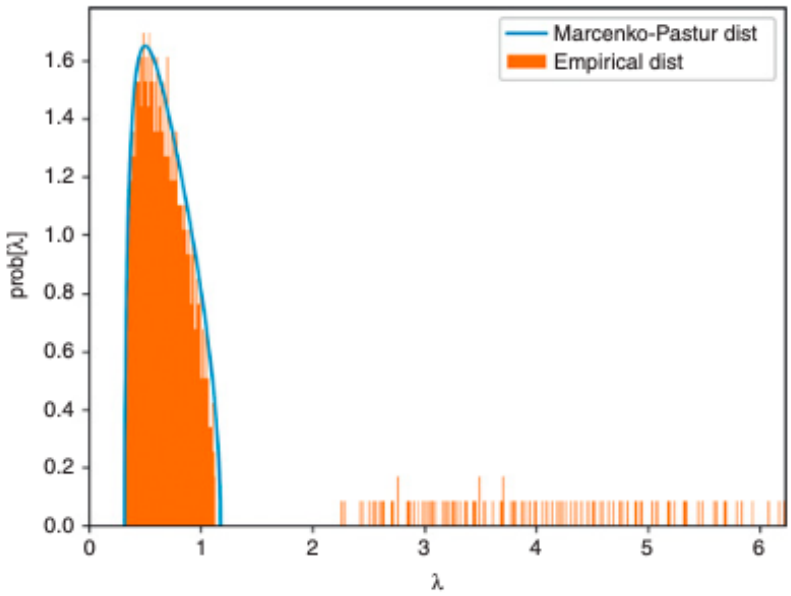


Figura 6.15: Distribuição de Marcenko-Pastur (Retirado de DE PRADO [14])

explicada por autovalores aleatórios contidos na matriz de correlação além de determinar o ponto de corte (λ_+) ajustado para a presença de autovetores não aleatórios. Na prática deve-se estimar o valor de σ^2 que minimiza a soma do quadrado (RMSE) da diferença entre a função densidade de probabilidade (PDF) analítica e a estimada. A função estimada pode ser obtida pela estimativa de densidade por Kernel (KDE), que é um método não paramétrico para estimar a PDF (mais detalhes em DEVROYE e GYÖRFI [51]).

Finalmente, para que seja possível o remover ruído da matriz de correlação, devemos atribuir um valor constante a todos os autovalores associados a um comportamento aleatório $\lambda < \lambda_+$. Esse valor é simplesmente a média de todos os autovalores contidos no intervalo $[0, \lambda_+]$, enquanto os demais serão mantidos de forma a manter o traço da matriz de correlação. Dada a decomposição dos autovetores $VW = W\Lambda$, obtém-se a matriz de correção C_1 sem a presença de ruído.

$$C_1 = \tilde{C_1} \left[\left(diag[\tilde{C_1}] \right)^{\frac{1}{2}} \left(diag[\tilde{C_1}] \right)^{\frac{1'}{2}} \right]^{-1} \tag{6.9}$$

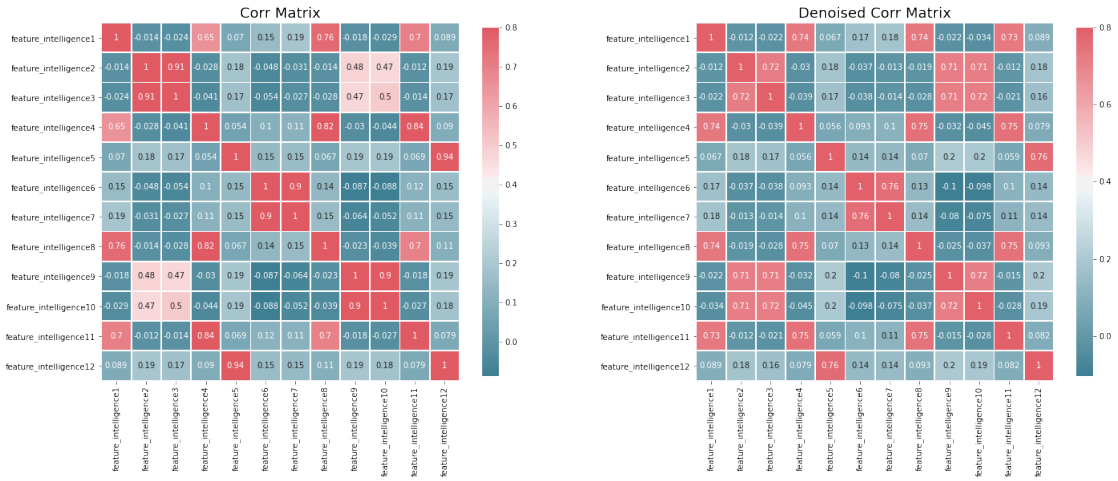
Onde $\tilde{C_1} = W\tilde{\Lambda}W'$ e $\tilde{\Lambda}$ é a matriz diagonal contendo os autovalores corrigidos e ordenados de maneira decrescente. A partir da equação 6.9 podemos normalizar a matriz $\tilde{C_1}$ de tal forma que C_1 possua um vetor de valores iguais a 1 na diagonal principal. DE PRADO [14] sugere também remover o maior autovalor, também conhecido como a "componente do mercado", pois é comum que exista alguma correlação entre todos os ativos devido a flutuação geral do próprio mercado. Tendo em mãos a matriz C_1, obtemos $\tilde{C_2}$ através da equação 6.10.

$$\tilde{C_2} = C_1 - W_M \Lambda_M W_M^{'} = W_D \Lambda_D W_D^{'} \tag{6.10}$$

Em seguida, obtém-se a matriz C_2 aplicando novamente a equação 6.9. É importante salientar que devido a remoção do autovalor associado a "componente de mercado"(Λ_M), a matriz C_2 é singular limitando seu uso em aplicações que se utilizam da matriz inversa, o que não é o caso em problemas de agrupamento (*clustering*). Porém como não lidamos diretamente com uma série de preços de ativos financeiros optou-se por não realizar essa segunda transformação, mas voltaremos a utilizá-la na seção 8.2.1.

Experimento Remoção de Ruído

Nesta seção será observada a matriz de correlação antes e depois da remoção do ruído nas variáveis do grupo *intelligence*. Note pela figura 6.16 que a diferença entre ambas é pequena, as maiores variações são percebidas em pares de variáveis onde a correlação já era significativamente alta, dessa forma é provável que tenhamos a formação de agrupamentos maiores, o que pode ser indesejável.



(a) Sem Remoção (b) Com Remoção

Figura 6.16: Matriz de Correlação - Remoção de Ruído

A figura 6.17 mostra a matriz sem ruído agrupada pelo algoritmo ONC. Note que apesar dos agrupamentos serem maiores que dois elementos, pela sua disposição na diagonal principal, o algoritmo ONC classificou esses agrupamentos como os de maior qualidade. Porém não é possível afirmar se a remoção de ruído levará a melhores resultados em termos de performance.

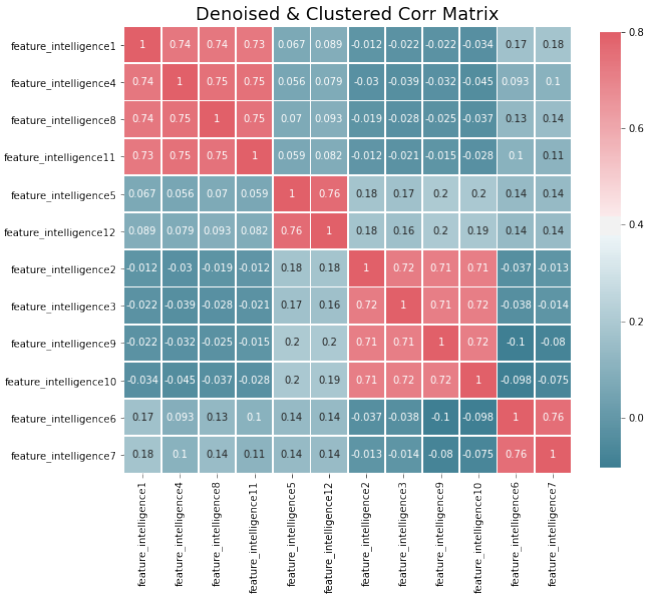


Figura 6.17: Matriz de Correlação sem Ruído Agrupada

6.3.4 Variação da Informação

A matriz de correlação utilizada até aqui, mesmo com a remoção de ruído ainda possui algumas ressalvas. Primeiro, quantifica apenas relações lineares, segundo é altamente influenciada por *outliers* e terceiro, sua aplicação em relações diferentes de gaussianas multivariadas é questionável. Uma maneira de superar essas barreiras é utilizando conceitos de teoria da informação [14].

Para uma variável aleatória discreta X, onde $x \in S_X$ com probabilidade $p[x]$. A entropia de X é definida como:

$$H[X] = -\sum_{x \in S_X} p[x] log[p[x]] \tag{6.11}$$

Entropia pode ser interpretada como o montante de incerteza associada a X. A entropia é zero quando toda a probabilidade foi concentra em um único elemento de S_X [52]. De maneira análoga a entropia conjunta $H[X, Y]$ onde Y é também uma variável aleatória discreta que não necessariamente precisa ser definida em um mesmo subespaço de probabilidade [14].

$$H[X, Y] = -\sum_{x,y \in S_X \times S_X} p[x, y] log[p[x, y]] \tag{6.12}$$

Seguindo essa linha a entropia condicional $H[X|Y]$ é definida pela equação 6.13, que quantifica a incerteza esperada em X explicada pelo valor de $Y = y$ [52].

$$H[X|Y] = H[X, Y] - H[Y] \tag{6.13}$$

A informação mútua $I[X, Y]$ é definida como o decréscimo na incerteza em X resultante do valor de Y [52].

$$I[X, Y] = H[X] - H[X|Y] \tag{6.14}$$

Contudo, a informação mútua não é uma métrica, porque não satisfaz a condição de desigualdade triangular citada na seção 6.3.1. Nesse contexto, a variação da informação ($VI[X, Y]$) pode ser interpretada como a incerteza esperada em uma variável se soubermos o valor de outra.

$$VI[X, Y] = H[X|Y] + H[Y|X] \tag{6.15}$$

Relacionando ao presente estudo, a Variação da Informação é uma medida de distância entre dois agrupamentos, sendo que esta é uma métrica de fato, pois obedece a desigualdade triangular [53]. A figura 6.18, ilustra a correspondência entre as medidas citadas nesta seção em um Diagrama de Venn. Podemos ver claramente que a entropia conjunta representa a união entre a entropia das duas variáveis ($H[X] \cup H[Y]$), a Informação Mútua é a interseção ($H[X] \cap H[Y]$) e a Variação da Informação corresponde a diferença simétrica entra ambas ($H[X] \triangle H[Y]$).

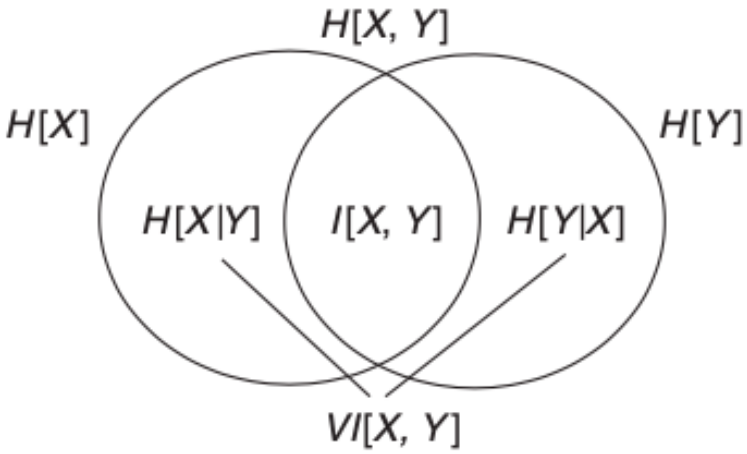
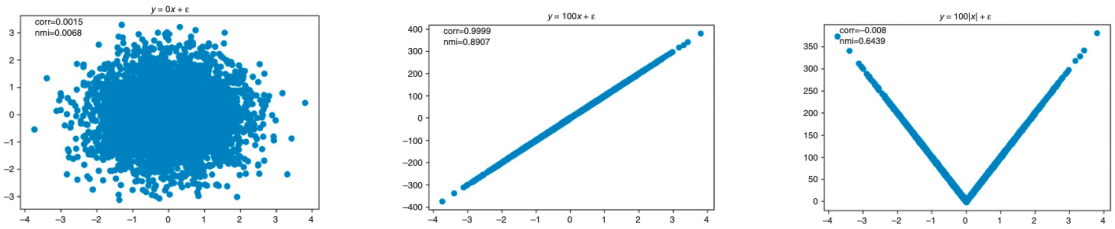


Figura 6.18: Correspondência entre Medidas (Retirado de DE PRADO [14])

Por fim, a figura 6.19 ilustra um experimento realizado por DE PRADO [14], para quantificar a correlação entre duas variáveis X e Y utilizando a correlação de pearson (corr) e a informação mútua normalizada (nmi), de forma que ambas fiquem na mesma escala [0,1]. Note que nos experimentos realizados nas figura 6.19a e 6.19b, ambas possuem valores próximos a 0 e 1 respectivamente, contudo na figura 6.19c, a correlação de pearson é próxima a zero, por outro lado o $nmi = 0.64$, que indica que a informação mútua foi capaz de capturar parte do relacionamento não linear entre X e Y.



(a) Aleatório (b) Linear (c) Não Linear

Figura 6.19: Experimento Informação Mútua (Retirado de DE PRADO [14])

Experimento Variação da Informação

O teste realizado nesta seção utilizou as variáveis do grupo *constitution*, com 114 variáveis e o maior dos grupos. As técnicas testadas anteriormente tiveram alguma dificuldade em apresentar um agrupamento satisfatório via algoritmo ONC, porém ao utilizar a Variação da Informação como matriz de dependência, o algoritmo ONC conseguiu agrupar as variáveis de maneira satisfatória. Note pela figura 6.20 que os agrupamentos com mais de 2 duas variáveis se encontram mais acima na diagonal principal, e portando possuem maior qualidade, por outro lado todos os agrupamentos na parte inferior e de menor qualidade possuem apenas duas variáveis.

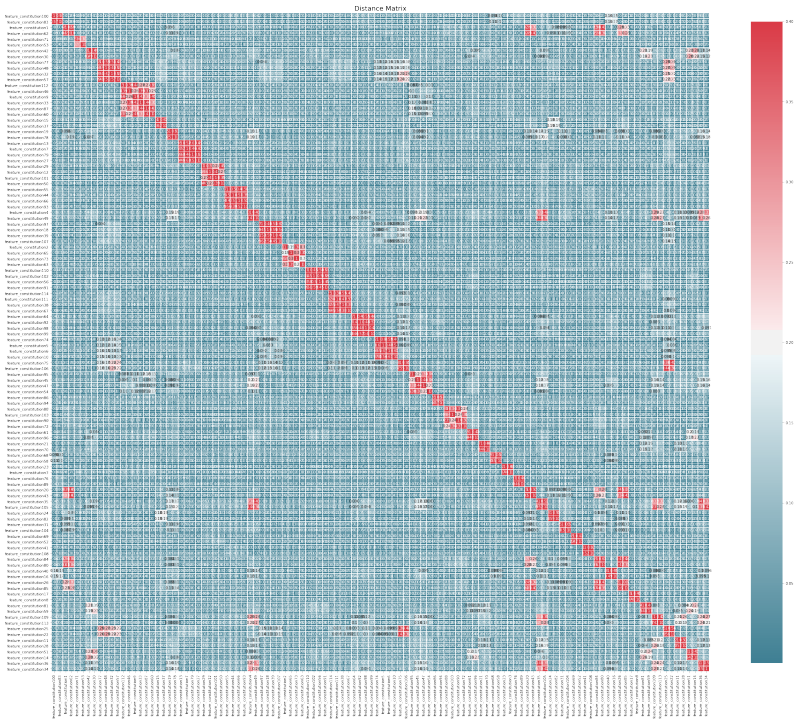


Figura 6.20: Variação da Informação Agrupada

6.3.5 Comparando Métodos

Nesta seção serão comparados os métodos de seleção de variáveis com agrupamento desenvolvidos ao longo deste capítulo. Para realizar esse comparativo utilizou-se no primeiro passo para obter a matriz de dependência:

1. Matriz de correlação
2. Matriz de correlação com remoção de ruído
3. Variação da informação
4. Misto: Utilizamos uma técnica em cada um dos 6 grupos individualmente

No segundo passo utilizou-se o algoritmo ONC, com a extensão proposta na seção 6.3.2.1, para a obtenção dos agrupamentos e por fim, performou-se uma permutação das variáveis (MDA) de cada agrupamento utilizando uma regressão linear ou um modelo *XGBoost*, totalizando 8 relações de importância de variáveis diferentes.

Observe a figura 6.21 que ilustra a correlação das importâncias de cada modelo e note que os modelos que utilizaram a regressão linear para permutação obtiveram um resultado significativamente diferentes do que utilizaram *Xgboost*.

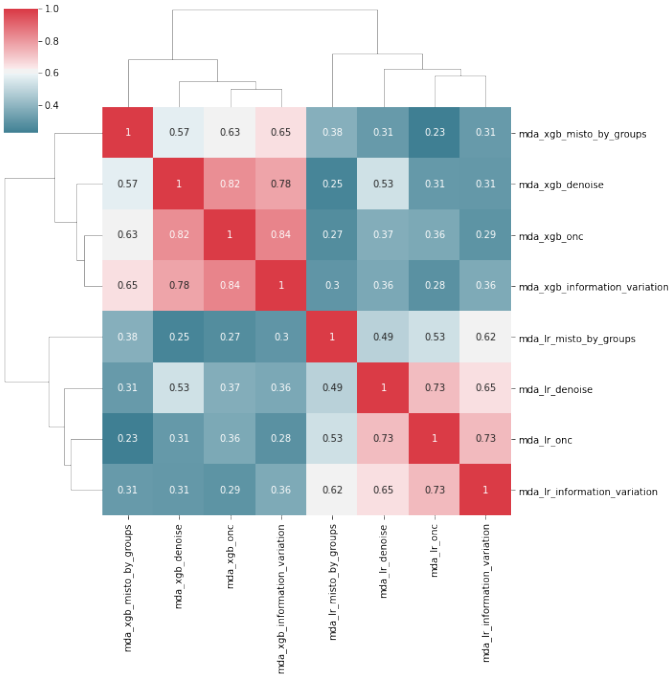


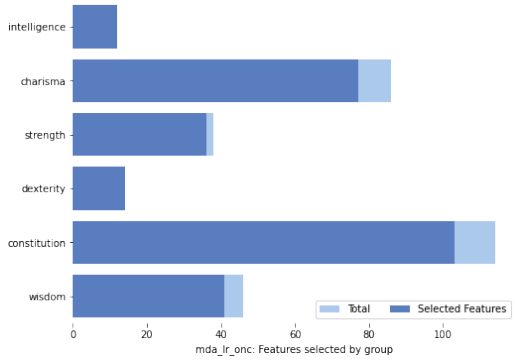
Figura 6.21: Correlação Importâncias - Métodos de Agrupamento

Em seguida, perceba que dentro desses 2 agrupamentos a correlação é significativamente alta o que leva a concluir que não há uma grande diferença entre escolher qualquer uma das 4 opções em cada agrupamento.

Agora perceba pela figura 6.23, que o modelo *XGBoost* excluiu uma quantidade significativamente maior de variáveis ($MDA < 0$) que a regressão linear. Neste caso, para os modelos que utilizaram a regressão linear performaremos uma neutralização de variáveis novamente e para os modelos que utilizaram *XGBoost* na etapa de permutação, a variável será completamente removida.

Figura 6.22: Variáveis Selecionadas por Grupo

(a) LR + Corr + ONC



(b) XGB + Misto + ONC

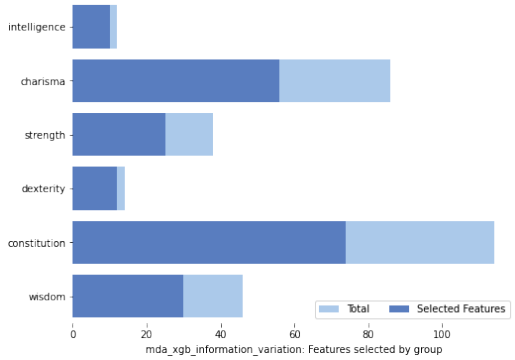


Figura 6.23: Variáveis Selecionadas por Grupo

6.3.6 Comparando Métricas

Agora que há modelos o suficiente para uma análise comparativa mais profunda, deve se ter em mente que o objetivo final é otimizar a performance futura ($payout = era\_score + mmc\_score$). Contudo é fácil perceber que performance passada não é o melhor indicativo de performance futura. A figura 6.24 mostra a correlação das métricas indicadas entre todos os modelos desenvolvidos nos capítulos 4, 5 e 6 na base de treino e suas respectivas performances na base de validação (val_payout).

Note que a performance na base de treino ($train\_payout$) possui uma correlação de apenas 0.4167 com a performance futura, sendo apenas o quinto colocado entre as nove métricas testadas.

A métrica *Sortino Ratio* obteve o melhor resultado neste comparativo, esta métrica é similar ao *Sharpe Ratio*, $\mu/\sigma_{Downside}$, a principal diferença é que apenas a volatilidade negativa é contabilizada ($\sigma_{Downside}$), este termo é calculado pela equação 6.16.

$$\sigma_{Downside} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} min[(Rt - Payout), 0]^2} \tag{6.16}$$

	val_payout
train_sortino	0.5167
train_asr	0.4667
train_psr(%)	0.4500
train_smart_sortino	0.4500
train_payout	0.4167
train_Max_DD	0.4167
train_sharpe	0.3667
train_smart_sharpe	0.2667
train_Std_Dev	-0.2167

Figura 6.24: Correlação Métrica com Retornos Futuros

Resultados - Com e Sem ONC

Agora, compare o modelo desenvolvido na seção 6.2.2 que permutou as variáveis individualmente com o seu equivalente com as variáveis agrupadas (seção 6.3.2). A figura 6.25a ilustra o resultado de ambos os modelos na base de treino, perceba que o modelo sem ONC performou melhor em todas as métricas analisadas, exceto a performance (val_payout).

	ex_preds	lr	ex_FN100	mda_linear_reg	mda_lr_onc
train_psr(%)	48.4559	0.2867	0.0002	95.4766	73.0074
train_asr	1.2328	1.0274	0.9010	1.7427	1.5621
train_sharpe	1.3984	1.0659	0.9028	1.6137	1.4786
train_payout	0.0471	0.0423	0.0273	0.0449	0.0475
train_sortino	3.4356	2.1855	1.2809	7.2184	5.4996

(a) ONC - Base Treino

	ex_preds	lr	ex_FN100	mda_linear_reg	mda_lr_onc
val_psr(%)	47.2640	0.6586	33.9126	92.0389	71.3613
val_asr	0.8582	0.3464	0.8040	1.0439	1.0086
val_sharpe	0.8701	0.3564	0.7965	1.3900	1.0240
val_payout	0.0241	0.0155	0.0253	0.0317	0.0281
val_sortino	1.0299	0.1659	0.9992	2.2417	1.5185

(b) ONC - Base Validação

Figura 6.25: Comparativo com e sem ONC

Contudo, analisando a figura 6.25b, perceba que o modelo sem ONC obteve performance superior na base de validação, dando mais um indicativo de que a performance passada não é o melhor indicativo para performance futura, além disso

obteve-se alguma evidência que utilizar o algoritmo ONC não melhorou os resultados.

Resultados - Neutralização ou Remoção

Nesta serão comparados alguns dos modelos obtidos nas seções 6.3.3 e 6.3.4. Ao aplicar o algoritmo ONC lida-se com a multicolinearidade linear e através da Variação da Informação conseguimos medir relacionamentos não lineares. Agora compare esses métodos removendo as variáveis completamente (realizando um novo treinamento do algoritmo *Xgboost*) ou apenas neutralizando. As três primeiras colunas da figura 6.26a se referem aos resultados com neutralização e as seguintes com remoção. Os 3 melhores resultados em cada métrica foram marcados na cor verde, perceba que 2 dos modelos com neutralização possuem a cor verde na maior parte das métricas e apenas um deles entre os modelos com remoção. A figura 6.26b, dá ainda mais evidência de que a neutralização apresenta resultados superiores, uma vez que os 3 modelos neutralizados apresentam os melhores resultados em praticamente todas as métricas.

	mda_xgb_denoise	mda_xgb_iv	mda_xgb_misto_by_groups	remove_xgb_denoise	remove_xgb_iv	remove_xgb_misto_by_groups
train_psr(%)	24.4827	99.5930	87.0839	89.7688	78.3057	86.3201
train_asr	1.1500	1.7961	1.6208	1.4359	1.4435	1.4405
train_sharpe	1.3030	1.7965	1.5475	1.6025	1.5162	1.5699
train_payout	0.0436	0.0470	0.0483	0.0580	0.0560	0.0589
train_sortino	2.9734	7.2460	6.2926	5.4935	4.9916	5.4425

(a) FN x Remoção - Base Treino

	mda_xgb_denoise	mda_xgb_iv	mda_xgb_misto_by_groups	remove_xgb_denoise	remove_xgb_iv	remove_xgb_misto_by_groups
val_psr(%)	23.2172	19.8999	1.5213	1.3395	0.8714	1.2321
val_asr	0.7613	1.2058	0.8974	0.8627	0.8442	0.8964
val_sharpe	1.1361	1.1947	0.9032	0.8806	0.8550	0.8946
val_payout	0.0312	0.0316	0.0256	0.0267	0.0286	0.0275
val_sortino	1.4552	2.3414	1.1802	1.1675	1.2156	1.2528

(b) FN x Remoção - Base Validação

Figura 6.26: Comparativo Neutralização x Remoção

Resultados - Todos os Modelos

Finalmente, pode se comparar os resultados obtidos pelos modelos desenvolvidos ao longo deste capítulo. A figura 6.27a mostra os comparativo na base de treino, note que o modelo *mda_xgb_information_variation* obteve os melhores resultados na maior parte das métricas, contudo o modelo equivalente com remoção completa das variáveis *remove_xgb_information_variation*, obteve uma performance

(*payout*) significativamente acima, porém há possibilidade deste resultado ser apenas sobreajuste, esta hipótese é corroborada pela figura 6.27b, onde o modelo com remoção obteve uma performance significativamente abaixo, além disso o modelo *mda_xgb_information_variation* não obteve a melhor performance na base de validação entre todos os modelos por uma diferença muito pequena, dando indício que este pode ser de fato o melhor modelo desta seção.

	ex_preds	lr	ex_FN100	mda_linear_reg	mda_lr_onc	mda_xgb_denoise	mda_xgb_information_variation	remove_xgb_information_variation
train_psr(%)	48.4559	0.2867	0.0002	95.4766	73.0074	24.4827	99.5930	78.3057
train_asr	1.2328	1.0274	0.9010	1.7427	1.5621	1.1500	1.7961	1.4435
train_sharpe	1.3984	1.0659	0.9028	1.6137	1.4786	1.3030	1.7965	1.5162
train_payout	0.0471	0.0423	0.0273	0.0449	0.0475	0.0436	0.0470	0.0560
train_sortino	3.4356	2.1855	1.2809	7.2184	5.4996	2.9734	7.2460	4.9916

(a) Comparativo - Base Treino

	ex_preds	lr	ex_FN100	mda_linear_reg	mda_lr_onc	mda_xgb_denoise	mda_xgb_information_variation	remove_xgb_information_variation
val_psr(%)	1.0844	0.0000	0.2458	48.4235	6.0025	23.2172	19.8999	0.8714
val_asr	0.8582	0.3464	0.8040	1.0439	1.0086	0.7613	1.2058	0.8442
val_sharpe	0.8701	0.3564	0.7965	1.3900	1.0240	1.1361	1.1947	0.8550
val_payout	0.0241	0.0155	0.0253	0.0317	0.0281	0.0312	0.0316	0.0286
val_sortino	1.0299	0.1659	0.9992	2.2417	1.5185	1.4552	2.3414	1.2156

(b) Comparativo - Base Validação

Figura 6.27: Comparativo Todos os Modelos

6.4 Resumo do Capítulo

Ao longo deste capítulo foram desenvolvidos métodos básicos de seleção de variáveis, em seguida um método de agrupamento de variáveis a partir da matriz de correlação com uma extensão para este algoritmo não prevista originalmente. Performou-se um método de remoção de ruído sobre a matriz de correlação e utilizou-se um método baseado em teoria da informação que é robusto a relacionamentos não lineares, por fim os modelos selecionados para análise posterior são:

1. **MDA Linear Reg** criado na seção 6.2.2
2. **MDA Linear Reg ONC** criado na seção 6.3.2
3. **MDA XGB Denoise** criado na seção 6.3.3
4. **MDA XGB Information Variation** criado na seção 6.3.4
5. **Feature Remove MDA XGB Information Variation** criado na seção 6.3.4

Através da figura 6.28, observe o diagnóstico dos 5 modelos desenvolvidos neste capítulo, além de alguns dos modelos desenvolvidos no capítulo 5 e o modelo de referência. Observe que os modelos deste capítulo obtiveram um resultado superior aos modelos desenvolvidos no capítulo 5.

	ex_preds	psr_group	psr_vol	mda_linear_reg	mda_lr_onc	mda_xgb_denoise	mda_xgb_information_variation	remove_xgb_information_variation
Validation_Sharpe	0.8861	0.9849	1.0437	1.2932	0.9961	1.1604	1.1629	0.8889
Validation_Mean	0.0241	0.0208	0.0245	0.0268	0.0259	0.0265	0.0270	0.0259
Feat_neutral_mean	0.0182	0.0179	0.0180	0.0187	0.0188	0.0190	0.0196	0.0202
Validation_SD	0.0272	0.0211	0.0234	0.0207	0.0260	0.0228	0.0232	0.0291
Feat_exp_max	0.2666	0.2692	0.2064	0.2023	0.2781	0.2005	0.1944	0.2699
Max_Drawdown	-0.0353	-0.0236	-0.0199	-0.0278	-0.0290	-0.0384	-0.0224	-0.0369
_plus_mmc_sharpe	0.8861	0.8656	1.0627	1.4156	1.0428	1.1569	1.2167	0.8707
val_mmc_mean	0.0000	-0.0004	0.0021	0.0049	0.0022	0.0047	0.0045	0.0027
ith_example_preds	1.0000	0.8748	0.8904	0.8374	0.9525	0.8321	0.8663	0.9273

Figura 6.28: Diagnóstico Final

Por fim a última análise deste capítulo refere-se a diversidade da performance dos modelos. O agrupamento hierárquico encontrou basicamente 3 agrupamentos, note também que os 3 modelos com melhor diagnóstico (**MDA Linear Reg**, **MDA XGB Denoise**, **MDA XGB Information Variation**) encontram-se no mesmo agrupamento, podendo ser necessário atingir um nível ainda maior de diversidade.

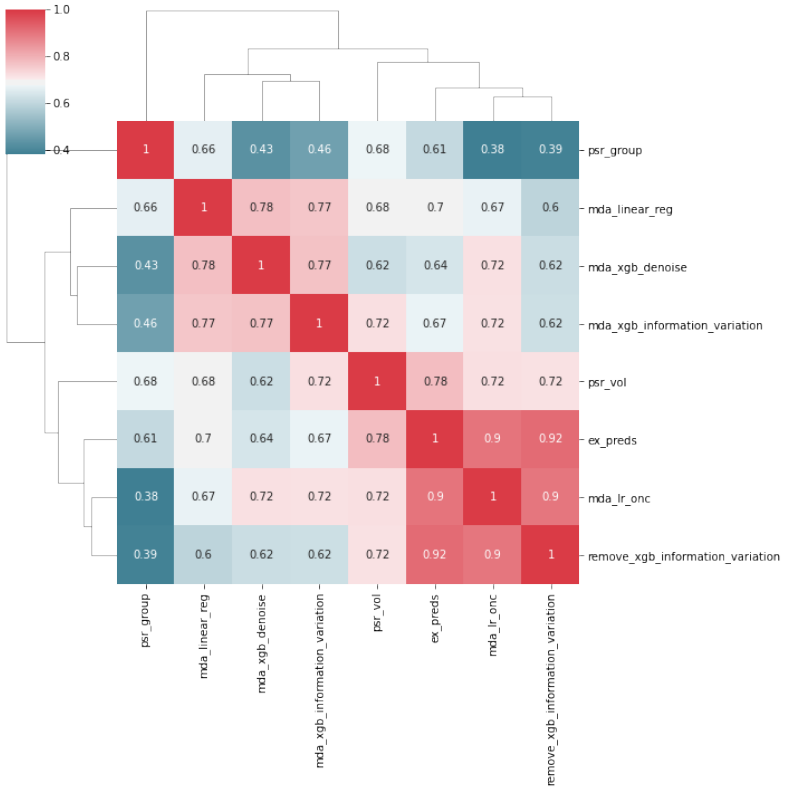


Figura 6.29: Correlação dos Retornos

Capítulo 7

Detecção de Regimes

O objetivo deste capítulo é identificar diferentes regimes que direcionam o comportamento dos estimadores ao longo do tempo e desenvolver modelos focados nesses regimes. Em seguida deve-se selecionar quais variáveis funcionam melhor em cada regime de acordo com as técnicas estudadas no capítulo 6 e por fim será feita uma tentativa de prever mudanças de regime de maneira eficaz.

7.1 Introdução

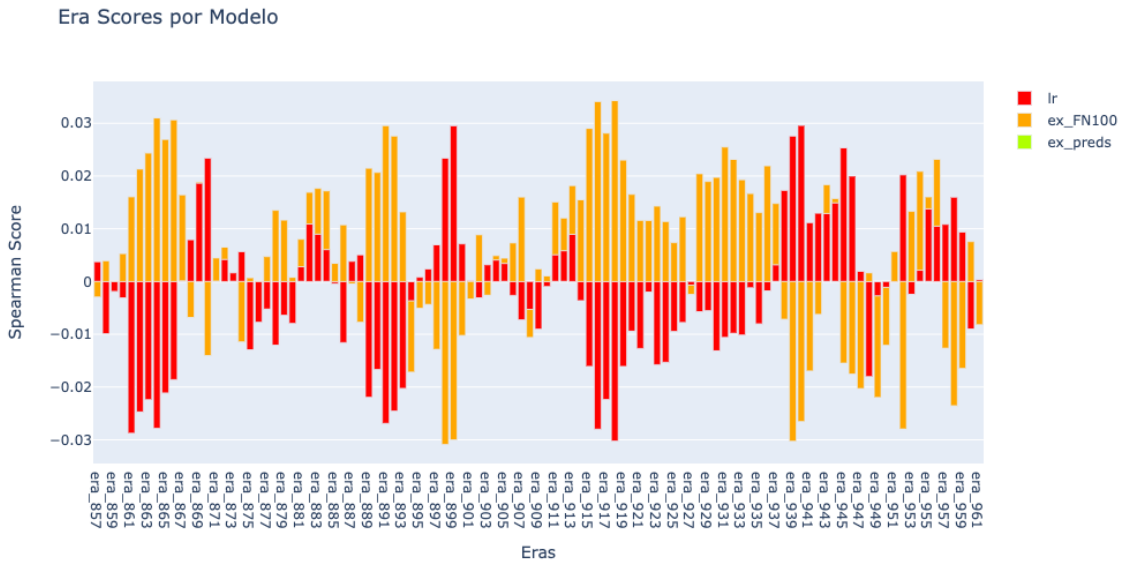


Figura 7.1: MMC Linear x Não Linear

Voltando a seção 4.3.3, discuti-se que a contribuição ao Meta Modelo (MMC_Scores) é o resíduo ortogonal entre as predições de um estimador neutralizadas pelo modelo de referência. Logo $\hat{MMC}_{baseline} \approx 0$. Por outro lado, a figura 7.1 mostra o comportamento do MMC_Score para a Regressão Linear e o

modelo de referência neutralizado (ex_FN100). Observando o gráfico de barras é possível perceber um comportamento espelhado entre as duas séries que possuem uma correlação negativa ($\rho_{LR \times FN} = -0.75$), dando indício de se tratar de dois regimes antagônicos.

Outro ponto interessante e fácil de observar pelo gráfico é que uma vez estabelecido, um regime tende a perdurar por algum tempo. Ao formar uma série a partir da função indicadora $I_{Dom} = max(I_{LR}, I_{FN})$, com o maior valor para cada era, haverá uma autocorrelação significativamente alta $AR1_{Sign} \approx 0.44$.

Logo, destaca-se a presença de dois regimes, sendo um de dominância linear onde ($MMC_{LR} > MMC_{FN}$), onde geralmente o retorno financeiro é maior e está presente em 77 das 120 eras da base de treino e um outro regime dominância não linear onde ($MMC_{LR} < MMC_{FN}$) presente nas demais 43 eras. Essas eras geralmente apresentam retorno baixo ou negativo. É fácil perceber que esse arranjo além de ser mutuamente excludente se assemelha ao conceito dos touros e ursos utilizados comumente no mercado.

7.1.1 Modelos por Regime

Seguindo o conceito introduzido na seção 7.1, podemos treinar novos modelos utilizando apenas as eras contidas em cada regime e obter suas predições utilizando validação cruzada. Ao concatenarmos os resultados dos 2 modelos (um para cada regime) e assumindo que utilizamos o melhor regime para cada era, obtém-se um novo conjunto de modelos com o sufixo "dom", derivados dos três modelos originais.

	ex_preds	xgb_dom	lr	lr_dom	ex_FN100	fn_dom
full_payout	0.0471	0.0614	0.0423	0.0483	0.0273	0.0279
full_asr	1.2328	1.3966	1.0274	1.0079	0.9010	0.7565
full_sharpe	1.3984	1.4722	1.0659	1.0643	0.9028	0.7506
full_sortino	3.4356	4.9523	2.1855	2.2347	1.2809	1.0066
full_Std_Dev	0.0335	0.0415	0.0395	0.0452	0.0301	0.0370
full_Max_DD	-0.1100	-0.0699	-0.0799	-0.0792	-0.1005	-0.0828
full_smart_sharpe	1.2301	1.4378	1.0601	1.0385	0.8527	0.7452

Figura 7.2: Comparativo Todas as Eras

A figura 7.2 mostra os resultados dos 3 modelos originais além dos seus equivalentes treinados apenas nas eras de cada regime. Claramente o modelo xgb_dom apresentou os melhores resultados na base de treino, destacando-se a grande diferença no *payout* entre seu equivalente treinado em todas as eras (6.14% vs 4.71%).

Por outro lado pode ser interessante observar os resultados obtidos em cada regime separadamente. Note que para as eras tidas como fáceis (dominância linear),

a regressão linear treinada em todas as eras (lr) apresenta os melhores resultados na maior parte das métricas.

	ex_preds	xgb_dom	lr	lr_dom	ex_FN100	fn_dom
easy_payout	0.0570	0.0689	0.0636	0.0693	0.0135	0.0172
easy_asr	2.4177	1.9463	2.6243	2.2291	0.5305	0.4983
easy_sharpe	1.9726	1.7419	2.3345	2.0774	0.5364	0.4902
easy_sortino	57.6736	19.3728	148.3390	34.6301	0.1883	0.3341
easy_Std_Dev	0.0287	0.0393	0.0271	0.0332	0.0250	0.0349
easy_Max_DD	-0.0000	-0.0088	-0.0000	-0.0008	-0.1032	-0.1052
easy_smart_sharpe	1.6650	1.5850	2.2932	2.0166	0.5173	0.4462

Figura 7.3: Comparativo Eras Fáceis

No caso das eras difíceis (dominância não linear) o modelo neutralizado treinado em todas as eras é superior aos demais em todas as métricas.

	ex_preds	xgb_dom	lr	lr_dom	ex_FN100	fn_dom
hard_payout	0.0293	0.0480	0.0041	0.0105	0.0520	0.0470
hard_asr	0.7695	0.9958	0.1446	0.2671	2.5293	1.2802
hard_sharpe	0.8473	1.1277	0.1460	0.2680	2.3973	1.4189
hard_sortino	1.0532	2.2295	-0.2570	0.0029	33.3150	3.5928
hard_Std_Dev	0.0342	0.0420	0.0278	0.0387	0.0214	0.0327
hard_Max_DD	-0.1100	-0.0699	-0.1365	-0.1013	-0.0000	-0.0384
hard_smart_sharpe	0.7501	1.0139	0.1359	0.2480	2.3080	1.2924

Figura 7.4: Comparativo Eras Difíceis

Por fim, estes resultados ainda são bastante inconclusivos para afirmar se é válido treinar um modelo em um subconjunto de eras ou se é melhor utilizar determinado modelo no regime específico. Por outro lado, os resultados mostraram que modelos simples como a regressão linear ou o modelo de referência neutralizado atingem excelentes resultados em regimes específicos.

7.2 Selecionando Variáveis por Regime

Este arranjo com eras fáceis e difíceis, pode ser utilizado para recriar as estratégias de seleção de variáveis. Nesta seção as estratégias desenvolvidas nos capítulos 5 e 6 serão refeitas, sendo uma cada regime. Para as estratégias de neutralização por grupos, serão utilizados os mesmos resultados do experimento da seção 5.3.5. A principal diferença é obter o Sharpe do modelo de referência para os 2 regimes.

Esses valores podem ser verificados nas figuras 7.3 e 7.4, sendo $SR_{LR} = 1.9726$ e $SR_{FN} = 0.8473$ respectivamente.

Nas eras fáceis, o experimento obteve um $S\hat{R}^* = 1.6741$, que é abaixo do Sharpe do modelo de referência, levando a um $PSR(S\hat{R}^*) = 0.0510$ e consequentemente um $DSR(S\hat{R}^*) \approx 0$. Por outro lado nas eras difíceis chegou-se a um $S\hat{R}^* = 1.8887$ bastante superior ao valor base e assim obteve-se um $DSR(S\hat{R}^*) = 0.9103$, que ainda é um pouco inferior aos 95% exigidos, mas é um bom resultado se comparado ao experimento anterior. As estratégias obtidas em cada experimento podem ser vistas na tabela 7.1.

Tabela 7.1: Estratégias Neutralização por Grupos e Regime

Grupo	PSR_{LR}	PSR_{FN}
intelligence	0	-0.5
wisdom	0	1
charisma	-0.5	0
dexterity	0	1
strength	1	0
constitution	1.5	0

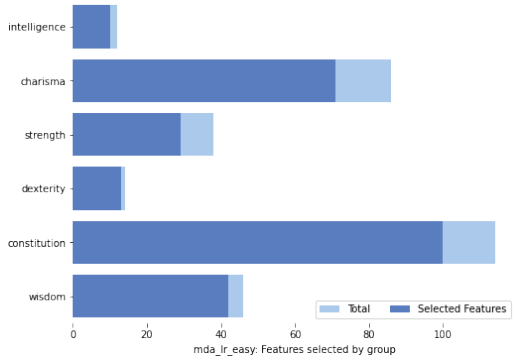
Note que a estratégia PSR_{FN} é idêntica a estratégia obtida seção 5.3.5. Além disso, não surpreende o fato de ter obtido sucesso apenas nas eras difíceis, pois em cada estratégia apenas partes da porção linear de cada variável é removida.

No experimento das métricas de volatilidade desenvolvido na seção 5.4.3, nas eras fáceis obteve-se $S\hat{R}^* = 1.9316$ também inferior ao Sharpe do modelo de referência, a se destacar apenas que esta é a mesma estratégia do experimento anterior com todas as eras, onde neutralizamos em 100%, 40% das variáveis ordenadas pelo pior Smart Sharpe.

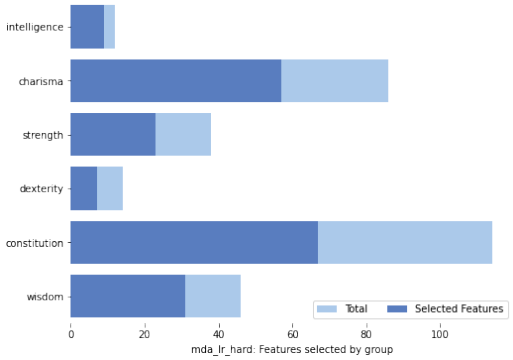
Para as eras difíceis $S\hat{R}^* = 2.2103$ e considerando as 17 tentativas independentes era esperado um Sharpe $E\left[\max\left\{\widehat{SR}_n\right\}\right] = 1.5032$, levando a um $DSR(S\hat{R}^*) = 0.9985$ bastante satisfatório. A estratégia consiste em neutralizar em 75%, 50% das variáveis ordenadas pelo pior *Max Drawdown*.

Agora é necessário repetir os experimentos desenvolvidos ao longo do capítulo 6. Serão utilizadas quatro estratégias para cada regime, utilizando o MDA, SFI apresentadas na primeira parte do capítulo, além do algoritmo ONC e o ONC com Varição da Informação, vistos na segunda parte. Observe pela figura 7.5, que o método MDA removeu uma parcela significativamente maior de variáveis nas eras difíceis ($\approx 30\%$), que nas eras fáceis ($\approx 10\%$), como o esperado.

Por outro lado, ao observar a matriz de correlação entre as variáveis, por regime (figura 7.6), não há uma diferença significativa de forma a mudar a composição dos agrupamentos obtidos pelo algoritmo ONC.

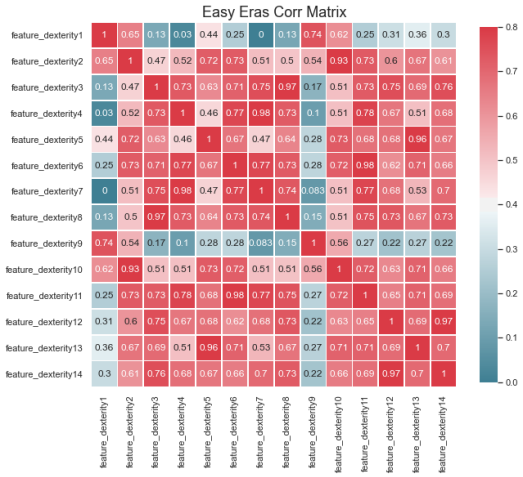


(a) Eras Fáceis

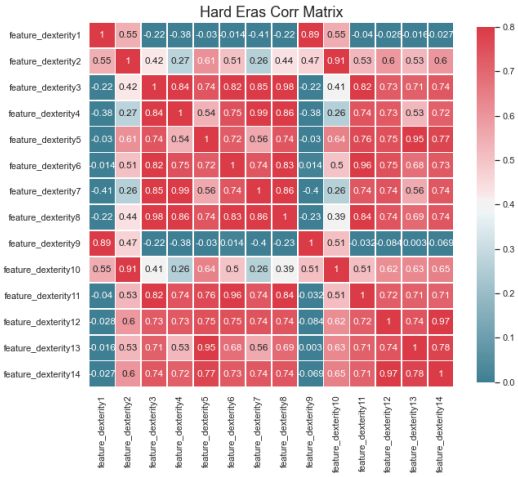


(b) Eras Fáceis

Figura 7.5: Variáveis Selecionadas por Regime com MDA



(a) Eras Difíceis



(b) Eras Difíceis

Figura 7.6: Correlação Variáveis por Regime

O que deve fazer diferença nesse caso é a diferença na correlação com o alvo, como ilustrado na figura 7.7. A maior parte das variáveis deste grupo possui uma correlação significativa com o algo (mesmo que negativa) nas eras fáceis, mas possuem correlação próxima a zero nas eras difíceis, se assemelhando a ruído.

	lr_eras	fn_eras	diff
feature_dexterity4	-0.0223	0.0046	-0.0269
feature_dexterity7	-0.0226	0.0042	-0.0267
feature_dexterity6	-0.0176	-0.0009	-0.0167
feature_dexterity11	-0.0165	-0.0001	-0.0164
feature_dexterity3	-0.0119	-0.0014	-0.0105

Figura 7.7: Correlação com Alvo por Regime

A figura 7.8 mostra os resultados obtidos pelas técnicas selecionadas para ambos os regimes. Apesar de em ambos os casos superar o modelo de referência com alguma margem, não foi possível superar a regressão linear (eras fáceis) e o modelo neutralizado (eras difíceis). Levando a entender que esses dois modelos, apesar de simples dificilmente podem ser superados com consistência em seus respectivos regimes.

	ex_preds	xgb_dom	lr	lr_dom	xgb_dom_mda_lr_easy	xgb_dom_sfi_xgb_easy	xgb_dom_mda_lr_information_variation_easy
easy_payout	0.0570	0.0689	0.0636	0.0693	0.0531	0.0686	0.0600
easy_asr	2.4388	1.9615	2.6422	2.2457	1.4132	1.9229	1.6272
easy_sharpe	1.9856	1.7533	2.3498	2.0910	1.2820	1.7810	1.5678
easy_sortino	57.6736	19.3728	148.3390	34.6301	5.3722	15.2770	8.8868
easy_Std_Dev	0.0287	0.0393	0.0271	0.0332	0.0414	0.0385	0.0383
easy_Max_DD	-0.0000	-0.0088	-0.0000	-0.0008	-0.0242	-0.0106	-0.0204
easy_smart_sharpe	1.6650	1.5850	2.2932	2.0166	1.2393	1.6937	1.5000

(a) Comparativo - Eras Fáceis

	ex_preds	ex_FN100	fn_dom	ex_preds_mda_lr_hard	ex_preds_sfi_xgb_hard	ex_preds_mda_xgb_information_variation_hard
hard_payout	0.0293	0.0520	0.0470	0.0508	0.0471	0.0438
hard_asr	0.7774	2.5638	1.2933	2.4321	1.8726	1.5275
hard_sharpe	0.8573	2.4257	1.4357	2.2103	1.7898	1.4240
hard_sortino	1.0532	33.3150	3.5928	19.7601	9.4440	6.0189
hard_Std_Dev	0.0342	0.0214	0.0327	0.0230	0.0263	0.0308
hard_Max_DD	-0.1100	-0.0000	-0.0384	-0.0028	-0.0143	-0.0172
hard_smart_sharpe	0.7501	2.3080	1.2924	2.1242	1.7610	1.3705

(b) Comparativo - Eras Difíceis

Figura 7.8: Comparativo Seleção de Variáveis por Regime

7.3 Relacionamento das Variáveis

Esta seção é dedicada a tecer alguns comentários sobre o relacionamento das variáveis com as predições dos estimadores e também como elas são utilizadas. Deve-se explicar o que faz uma variável ser boa ou ruim e o motivo de performarem bem com determinados modelos e com outros não. A figura 7.9, mostra a importância atribuída a cada variável por um modelo de *Gradient Boosting* de **regressão** (mais detalhes sobre este modelo em NORI *et al.* [47]).

Note que na grande maioria as variáveis pertencem ao grupo *dexterity* e a figura 7.10 ilustra uma análise mais profunda da variável com maior importância para este modelo. Em primeiro lugar, através da figura 7.10a, note que a presença de apenas 5 valores únicos para a variável no eixo x forma o gráfico na forma de uma escada descendente. Observando o eixo y, note que a medida que o valor da variável cresce, há um influência negativa no valor das predições geradas pelo estimador. A figura 7.10b mostra a correlação linear (pearson), das predições do estimador com

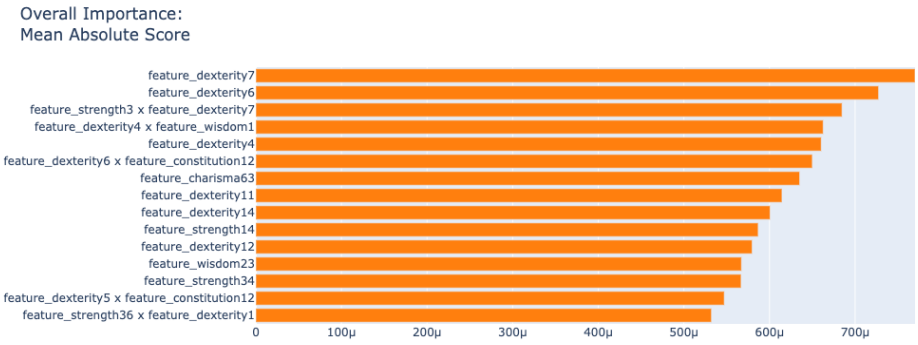
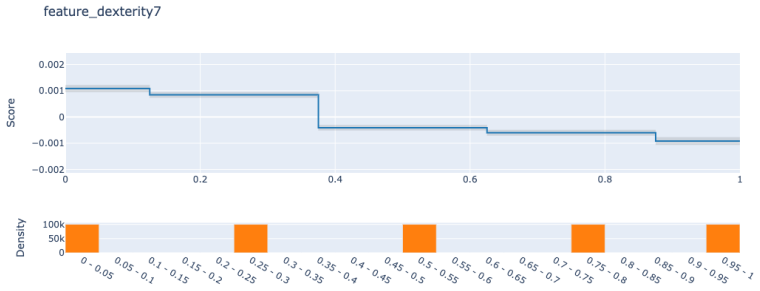
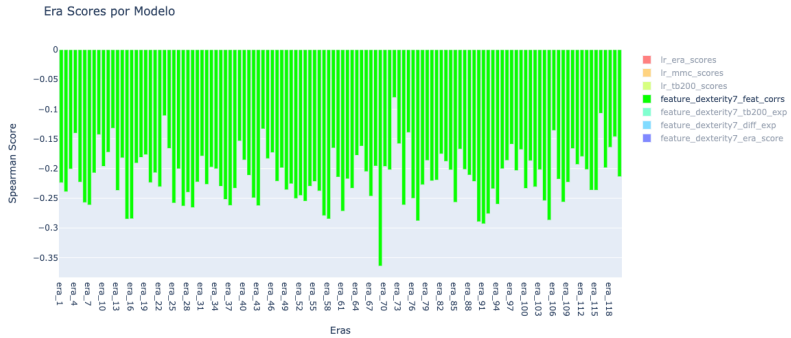


Figura 7.9: Importância das Variáveis - Regressão

a variável de estudo. Perceba que há sempre uma correlação negativa, indicando que para o caso de um estimador linear, será atribuído um coeficiente negativo para esta variável, o que também explica o comportamento da figura 7.10a. O fato desta variável possuir um comportamento bastante consistente certamente contribui para a performance do modelo de maneira positiva.



(a) Influência sobre as Predições



(b) Correlação com as Predições

Figura 7.10: Análise Variável - *Dexterity7*

Agora observe a figura 7.11 onde foi executada a mesma análise utilizando um modelo de **classificação**. Note que as variáveis com maior importância pertencem ao grupo *intelligence*.

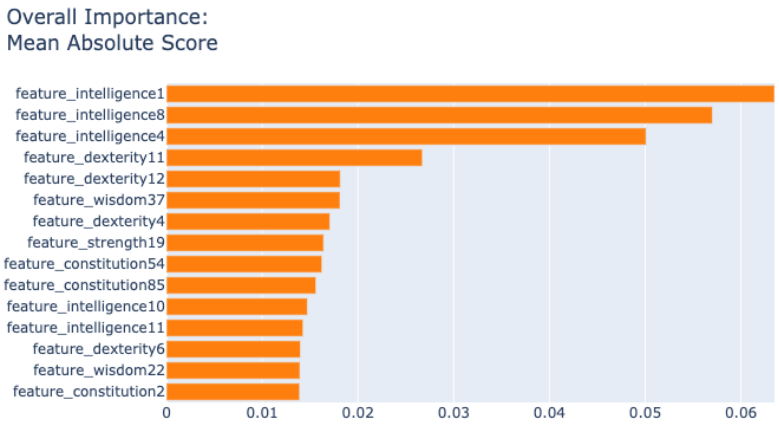
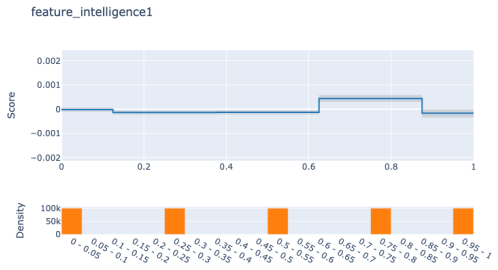
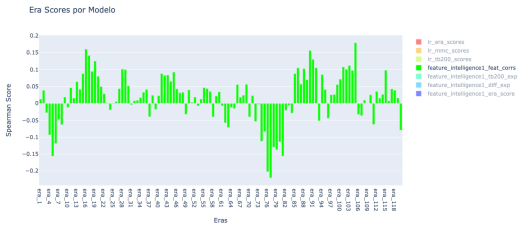


Figura 7.11: Importância das Variáveis - Classificação

Na figura 7.12, a variável mais importante é novamente analisada, porém sobre a ótica de um modelo de regressão. A 7.12a mostra uma relação irregular da influência da variável sobre as predições e a figura 7.12b mostra que a variável em determinados momentos possui correlação positiva e em outras negativa com as predições. Para o caso de um estimador linear é provável que seja atribuído um coeficiente próximo a zero, podendo ser rotulado como somente ruído.



(a) Influência sobre as Predições

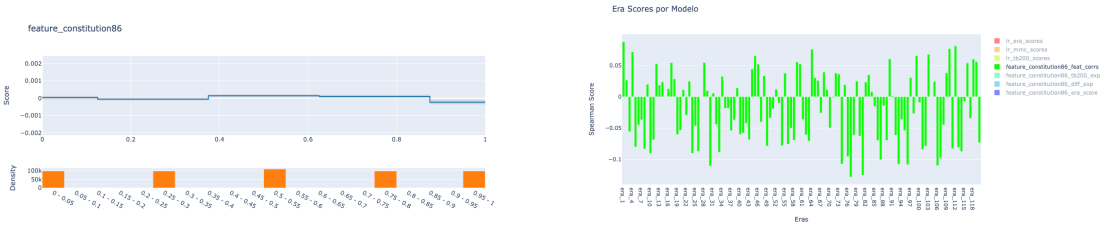


(b) Correlação com as Predições

Figura 7.12: Análise Variável - *Intelligence1*

Por fim, a variável com menor importância atribuída ao modelo de regressão foi selecionada (*Constitution 86*). Note novamente o comportamento irregular em ambos os gráficos da figura 7.13. Dessa maneira há alguma evidência de que ao menos a porção linear dessas variáveis não contribuem com o modelo e devem ser neutralizadas. Tentou-se de diversas maneiras encontrar alguma heurística capaz de identificar em quais regimes ou mesmo em quais eras a variável deveria ser neutralizada a partir a partir da concordância ou discordância do coeficiente linear atribuído a variável com sua correlação as predições do estimador linear, contudo esta estratégia não levou a bons resultados e foi descartada.

Alternativamente a solução adotada foi testar uma variável por vez a fim de



(a) Influência sobre as Predições (b) Correlação com as Predições

Figura 7.13: Análise Variável - *Constitution86*

verificar ganhos de performance. Para o regime linear, a partir das predições da regressão linear, melhor modelo para esse regime, de acordo com a figura 7.8a. Em seguida neutralizou-se uma variável por vez, a fim de ordenar quais produziram maior ganho de performance, o melhor resultado foi obtido ao neutralizar apenas 5% das melhores variáveis (de acordo com o passo anterior) a uma proporção de 125% como visto na figura 7.14a.

Para o regime não linear, adotou-se o procedimento oposto, partindo do modelo de referência, neutralizaremos todas as variáveis, exceto uma, a fim de verificar quais entregam performance superior ao modelo totalmente neutralizado (ex_FN100). O melhor resultado foi obtido ao neutralizamos 75% das variáveis com maior ganho a uma proporção de 125% como visto na figura 7.14b.

	ex_preds	lr	lr_dom	lr_fn_easy_eras_payout_05_125	lr_dom_fn_easy_eras_payout_05_125
easy_payout	0.0570	0.0636	0.0693	0.0658	0.0705
easy_asr	2.4388	2.6422	2.2457	2.7677	2.1341
easy_sharpe	1.9856	2.3498	2.0910	2.4433	2.0740
easy_sortino	57.6736	148.3390	34.6301	inf	26.1662
easy_Std_Dev	0.0287	0.0271	0.0332	0.0269	0.0340
easy_Max_DD	-0.0000	-0.0000	-0.0008	-0.0000	-0.0036
easy_smart_sharpe	1.6650	2.2932	2.0166	2.4197	2.0352

(a) Estratégias - Eras Fáceis

	ex_preds	ex_FN100	fn_dom	ex_preds_hard_payouts_s75_p100	ex_preds_hard_payouts_s75_p125
hard_payout	0.0293	0.0520	0.0470	0.0611	0.0630
hard_asr	0.7774	2.5638	1.2933	2.4902	2.8582
hard_sharpe	0.8573	2.4257	1.4357	2.5223	2.4748
hard_sortino	1.0532	33.3150	3.5928	57.2039	inf
hard_Std_Dev	0.0342	0.0214	0.0327	0.0242	0.0255
hard_Max_DD	-0.1100	-0.0000	-0.0384	-0.0000	-0.0000
hard_smart_sharpe	0.7501	2.3080	1.2924	2.3373	2.3520

(b) Estratégias - Eras Difíceis

Figura 7.14: Estratégias Seleção de Variáveis por Regime

Apesar do sucesso em superar os resultados obtidos para os modelos base de cada regime, é necessário salientar que este resultado possui um alta probabilidade de

conter sobreajuste (*overfiting*). O método adotado nesta sessão é bastante similar ao método da seção 5.2.2, onde verificou-se por diversas vezes o resultado das estratégias sem levar em consideração as tentativas. Tal procedimento não foi adotado aqui devido ao número baixo de tentativas independentes.

7.4 Diagnóstico Atribuição de Regime

Ao longo desta seção serão observados o diagnóstico dos modelos desenvolvidos ao longo deste capítulo na base de validação, com o objetivo de confirmar a efetividade das ideias desenvolvidas. **É importante salientar que o regime correto para cada era foi passado para os modelos**.

O modelo *xgb_dom* é treinado nas eras específicas de cada regime (na base de treino), na sequência utilizaremos as predições do estimador referente a cada regime em cada era. Pela figura 7.15 observe que este modelo obteve um resultado bastante superior ao seu equivalente treinado em todas as eras, indicando que a discriminação das eras nos dois regimes também houve boa aderência nas eras futuras.

	ex_preds	strategy_xgb_dom
Validation_Sharpe	0.8861	1.1720
Validation_Mean	0.0241	0.0297
Feat_neutral_mean	0.0182	0.0188
Validation_SD	0.0272	0.0253
Feat_exp_max	0.2666	0.2475
Max_Drawdown	-0.0353	-0.0186
corr_plus_mmc_sharpe	0.8861	1.0983
val_mmc_mean	0.0000	0.0093
corr_with_example_preds	1.0000	0.7041

Figura 7.15: Diagnóstico - Modelo com Atribuição de Regime

A figura 7.16 mostra que o modelo *xgb_dom* não foi capaz de superar a combinação da regressão linear e o modelo totalmente neutralizado (*lr_fn*), indicando que é necessário fazer algumas melhorias.

Voltando a neutralização por grupos, a figura 7.17 mostra que obtivemos um resultado levemente superior ao adotar uma estratégia por regime (*strategy_groups*) e ainda superamos o modelo de referência.

De maneira análoga, obteve-se ótimos resultados adotando um critério de neutralização por métrica de volatilidade em cada regime (*strategy_groups*), como visto na figura 7.18.

Os métodos de seleção de variáveis individuais performados por regime também trouxeram um resultado superior ao obtido analisando todas as eras em conjunto,

	ex_preds	lr	ex_FN100	strategy_lr_fn	strategy_xgb_dom
Validation_Sharpe	0.8861	0.5461	1.0475	1.7988	1.1720
Validation_Mean	0.0241	0.0163	0.0207	0.0321	0.0297
Feat_neutral_mean	0.0182	0.0030	0.0198	0.0205	0.0188
Validation_SD	0.0272	0.0299	0.0198	0.0179	0.0253
Feat_exp_max	0.2666	0.2784	0.0140	0.1272	0.2475
Max_Drawdown	-0.0353	-0.0853	-0.0217	0.0000	-0.0186
corr_plus_mmc_sharpe	0.8861	0.3629	0.8111	1.8930	1.0983
val_mmc_mean	0.0000	-0.0008	0.0046	0.0129	0.0093
corr_with_example_preds	1.0000	0.6550	0.6063	0.6242	0.7041

Figura 7.16: Diagnóstico - Modelos com Atribuição de Regime

	ex_preds	psr_group	strategy_groups
Validation_Sharpe	0.8861	0.9849	0.9133
Validation_Mean	0.0241	0.0208	0.0248
Feat_neutral_mean	0.0182	0.0179	0.0186
Validation_SD	0.0272	0.0211	0.0271
Feat_exp_max	0.2666	0.2692	0.2546
Max_Drawdown	-0.0353	-0.0236	-0.0285
corr_plus_mmc_sharpe	0.8861	0.8656	0.9063
val_mmc_mean	0.0000	-0.0004	0.0033
corr_with_example_preds	1.0000	0.8748	0.8550

Figura 7.17: Diagnóstico - Modelos Neutralizados por Grupos

	ex_preds	psr_vol	strategy_metric
Validation_Sharpe	0.8861	1.0437	1.2299
Validation_Mean	0.0241	0.0245	0.0270
Feat_neutral_mean	0.0182	0.0180	0.0187
Validation_SD	0.0272	0.0234	0.0219
Feat_exp_max	0.2666	0.2064	0.2122
Max_Drawdown	-0.0353	-0.0199	-0.0215
corr_plus_mmc_sharpe	0.8861	1.0627	1.3425
val_mmc_mean	0.0000	0.0021	0.0035
corr_with_example_preds	1.0000	0.8904	0.9199

Figura 7.18: Diagnóstico - Métrica de Volatilidade

como visto na figura 7.19.

	ex_preds	mda_linear_reg	strategy_sfi_mda
Validation_Sharpe	0.8861	1.2932	1.4427
Validation_Mean	0.0241	0.0268	0.0289
Feat_neutral_mean	0.0182	0.0187	0.0187
Validation_SD	0.0272	0.0207	0.0200
Feat_exp_max	0.2666	0.2023	0.1651
Max_Drawdown	-0.0353	-0.0278	-0.0264
corr_plus_mmc_sharpe	0.8861	1.4156	1.5174
val_mmc_mean	0.0000	0.0049	0.0066
corr_with_example_preds	1.0000	0.8374	0.8140

Figura 7.19: Diagnóstico - Seleção de Variáveis Individual

Por outro lado, não obteve-se o mesmo resultado quando repetimos a análise usando variáveis agrupadas (*strategy_onc_iv*), visto na figura 7.20.

	ex_preds	mda_xgb_information_variation	strategy_onc_iv
Validation_Sharpe	0.8861	1.1629	1.1187
Validation_Mean	0.0241	0.0270	0.0248
Feat_neutral_mean	0.0182	0.0196	0.0181
Validation_SD	0.0272	0.0232	0.0222
Feat_exp_max	0.2666	0.1944	0.2208
Max_Drawdown	-0.0353	-0.0224	-0.0240
corr_plus_mmc_sharpe	0.8861	1.2167	1.1520
val_mmc_mean	0.0000	0.0045	0.0032
corr_with_example_preds	1.0000	0.8663	0.8368

Figura 7.20: Diagnóstico - Seleção de Variáveis Agrupadas

Prosseguindo a análise da figura 7.16, foram utilizadas algumas das estratégias desenvolvidas na seção 7.3 a fim de aprimorar o modelo *xgb_dom*. Apesar de obtermos um resultado superior (*strategy_fs_xgb_dom_100*), note que este modelo obteve um resultado melhor que o equivalente neutralizado com 125% ao contrário do que obtivemos na base de treino, visto na figura 7.14.

Por fim, a figura 7.22 traz mais uma evidência de que o resultado da figura 7.14 é devido ao sobreajuste, uma vez que a estratégia com melhores resultados dos dados futuros não foi capaz de superar a simples combinação entre regressão linear e modelo neutralizado.

	ex_preds	strategy_xgb_dom	strategy_fs_xgb_dom_100	strategy_fs_xgb_dom_125
Validation_Sharpe	0.8861	1.1720	1.3526	1.3361
Validation_Mean	0.0241	0.0297	0.0324	0.0323
Feat_neutral_mean	0.0182	0.0188	0.0196	0.0196
Validation_SD	0.0272	0.0253	0.0240	0.0242
Feat_exp_max	0.2666	0.2475	0.1650	0.1665
Max_Drawdown	-0.0353	-0.0186	-0.0250	-0.0282
corr_plus_mmc_sharpe	0.8861	1.0983	1.4116	1.3940
val_mmc_mean	0.0000	0.0093	0.0125	0.0130
corr_with_example_preds	1.0000	0.7041	0.6249	0.5836

Figura 7.21: Diagnóstico - Seleção Iterativa

	ex_preds	strategy_lr_fn	strategy_fs_lr_xgb_dom_125_lev
Validation_Sharpe	0.8861	1.7988	1.6623
Validation_Mean	0.0241	0.0321	0.0320
Feat_neutral_mean	0.0182	0.0205	0.0199
Validation_SD	0.0272	0.0179	0.0193
Feat_exp_max	0.2666	0.1272	0.2335
Max_Drawdown	-0.0353	0.0000	-0.0085
corr_plus_mmc_sharpe	0.8861	1.8930	1.6756
val_mmc_mean	0.0000	0.0129	0.0151
corr_with_example_preds	1.0000	0.6242	0.4759

Figura 7.22: Diagnóstico - Comparativo Final

7.5 Determinando o Regime

Ao longo deste capítulo, sempre que necessário a informação sobre qual regime determinada era pertence foi passada de antemão. Isso certamente configura um vazamento de dados, uma vez que não é possível ser replicado em dados futuros, já que não é sabido a qual regime a próxima era pertence.

Voltando a observar a figura 7.1, deve-se recordar das propriedades das séries temporais utilizadas para classificar os regimes. Utilizando-se de uma heurística simples como atribuir o regime atual para a próxima era, temos uma taxa de sucesso igual a 73.26%, significativamente superior a 50%, caso a atribuição fosse totalmente aleatória.

É bastante tentador ampliar a gama de heurísticas simples levando em consideração um número maior de eras passadas ou atribuir diferentes pesos, privilegiando as eras mais recentes. Contudo um dos principais tópicos abordados ao longo neste estudo é que deve-se considerar a distribuição dos retornos gerados pela melhor estratégia e também as múltiplas tentativas independentes criadas ao longo do processo (ver seção 5.3.4). De toda forma, é fácil perceber que a medida que a série histórica

aumenta, o numero ótimo de eras a serem consideradas tende a oscilar além de não ser capaz de superar a taxa de sucesso base ($\approx 73\%$) com a consistência necessária.

Outra alternativa é se basear na informação disponível no momento de realizar as predições, que são basicamente as variáveis do problema para a era atual. Técnicas como validação adversária (ver GOODFELLOW *et al.* [54]), são úteis para classificar amostras entre classes distintas, porém as amostras são classificadas individualmente, sendo assim pouco útil para o presente estudo, uma vez que dependemos de todo o subconjunto de amostras de cada era para obtermos o ranqueamento relativo entre elas.

Esse problema pode ser contornado ao utilizar técnicas de similaridade de matrizes, considerando cada era como uma matriz individualmente e classificando-as entre regimes lineares ou não lineares. Existe uma literatura extensa nesta área e o acesso a bibliotecas de código aberto suportadas por grandes empresas como Facebook e Microsoft. Entre elas a biblioteca *FAISS* [55] que permite calcular rapidamente um algoritmo de K-Vizinhos mais próximos entre uma lista de vetores (mais detalhes em JAMES *et al.* [2]). Com isso é possível calcular a distância entre cada amostra da era futura com as outras amostras (vetores) das demais eras e calcular a média da distância do vizinho mais próximo em cada era. O problema desta e qualquer outra abordagem utilizando similaridade de matrizes é que as eras mais recentes sempre se sobressaem a qualquer mudança de regime. A figura 7.23 ilustra que a mudança de regime percebida entre as eras 300 e 400 pouco contribui para a alteração das distâncias para a era atual. A distância temporal é a força preponderante neste caso, dessa forma foi possível detectar uma mudança brusca de regime, uma vez que as eras anteriores (e mais próximas), pertencem a um regime diferente.

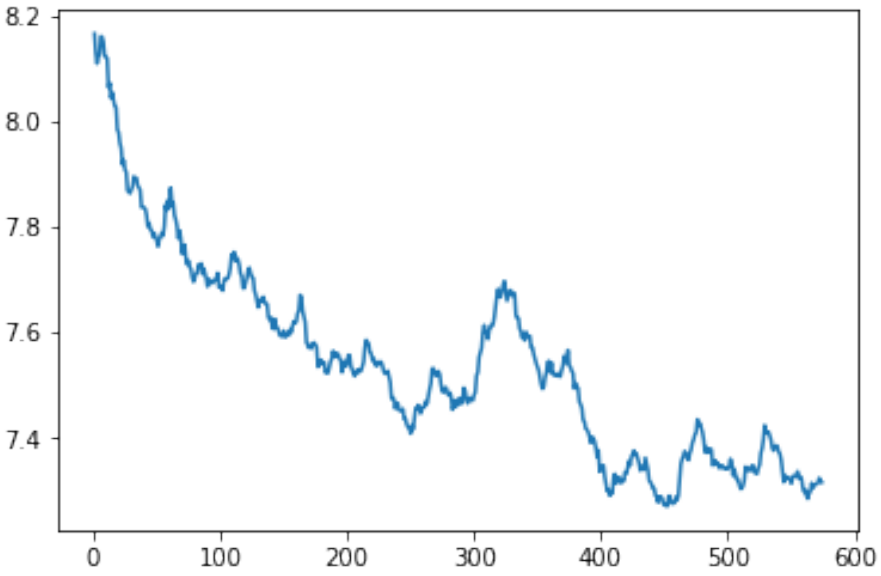


Figura 7.23: Distância Matrizes - Era Futura

Dessa maneira optou-se por atribuir o regime atual para a era futura, arcando com as eventuais perdas de performance por erro na classificação do regime. Tais efeitos serão abordados no capítulo 9.

7.6 Resumo do Capítulo

Neste capítulo verificou-se a eficácia da separação das eras em regimes distintos, foram refeitas todas as técnicas de seleção de variáveis desenvolvidas nos capítulos anteriores e verificou-se os resultados por regime de maneira discriminada e foram obtidos resultados superiores. Em seguida tentou-se sem sucesso desenvolver um método de atribuição de capaz de superar a heurística simples, onde a próxima será igual a atual. Por fim os modelos selecionados para análise posterior são:

1. **strategy_xgb_dom**: Modelo treinado nos subgrupos de eras de cada regime, alternando o modelo utilizado a depender do regime atribuído
2. **strategy_groups**: Usa o modelo de referência como base, além de uma estratégia de neutralização por grupos diferente para cada regime
3. **strategy_metric**: análogo ao anterior, porém com uma estratégia baseada em métricas de volatilidade
4. **strategy_lr_fn**: Alterna a regressão linear com o modelo de referência neutralizado, a depender do regime
5. **strategy_sfi_mda**: Alterna técnicas de seleção de variáveis simples
6. **strategy_onc_iv**: Alterna técnicas de seleção de variáveis agrupadas

Além destes, devido a grande quantidade de estratégias geradas na seção 7.3, foram selecionadas quatro estratégias com o nome **strategy_fs**, além de uma categoria como conservadora, moderada, agressiva e muito agressiva.

Capítulo 8

Otimização de Portfólio

Ao longo deste capítulo haverá uma breve introdução de modelos de otimização de portfólio com otimização convexa, em seguida serão abordadas diversas limitações para esta solução e então serão desenvolvidas alternativas que propõem resolver essas limitações. Finalmente, será feito um experimento a fim de verificar se os métodos de otimização de portfólio expostos ao longo do capítulo foram capazes de melhorar a relação risco e retorno do portfólio de modelos.

8.1 Introdução

Ao final do capítulo 7 conclui-se que nenhum dos diversos modelos desenvolvidos ao longo do trabalho foi capaz de comprovadamente performar acima dos outros e do modelo de referência com a consistência necessária. Além disso foram desenvolvidas estratégias significativamente arriscadas devido ao erro de atribuição de regime. Por esse motivo será desenvolvido um modelo de gestão de portfólio capaz de mitigar a volatilidade do portfólio de modelos.

Para esse experimento inicial serão utilizados os modelos desenvolvidos ao longo dos capítulos 4, 5 e 6, descritos nas seções de resumo ao final de cada capítulo. Além disso, serão utilizados os dados de treinamento apresentados na seção 3.2.1, sendo que as eras 1 a 60 serão utilizadas para confecção dos modelos e as 60 subsequentes para testes.

É importante destacar que será feito uma **aproximação** entre os retornos obtidos por cada modelo com os retornos periódicos obtidos por um portfólio de ações. O retorno obtido por um modelo em cada era é $R_{era} = corr + mmc$ e a série de retornos acumulada obtida pela equação 8.1.

$$RetornoAcumulado = \left(\prod_{era=1}^{N} (1 + R_{era}) \right) - 1 \tag{8.1}$$

Além disso o retorno por era de todo o portfólio de modelos pode ser obtido pela equação 8.2. Onde P_0 é o investimento inicial e r_{era} e w_{era} são vetores contendo os retornos e os pesos atribuídos em cada modelo por era.

$$P_{era} = P_0 \prod_{era=1}^{N} r_{era} \cdot w_{era} \tag{8.2}$$

Por fim, a variância do portfólio (σ_p^2) com 2 ativos (A e B) pode ser obtida pela equação 8.3, onde $\rho_{A,B}$ é a correlação entre os dois ativos.

$$\sigma_p^2 = w_A^2\sigma_A^2 + w_B^2\sigma_B^2 + 2w_A w_B \sigma_A \sigma_B \rho_{A,B} \tag{8.3}$$

8.2 Modelos de Otimização Convexa

Antes de prosseguir, é necessário introduzir rapidamente o conceito de diversificação de risco, onde um portfólio contendo apenas dois ativos A e B, sendo A o ativo menos volátil e com menor retorno esperado e o oposto para o ativo B. O modelo proposto por MARKOWITZ [56] prova que a relação entre risco e retorno da carteira não é linear e existe uma combinação entre os dois ativos onde a variância do portfólio é menor que a variância do ativo menos volátil, sendo esse o portfólio de mínima variância. Além disso existe uma outra combinação entre esses ativos onde para um mesmo nível de risco, é esperado um retorno maior. A figura 8.1 ilustra a fronteira eficiente, em verde, onde para um dado nível de risco podemos obter o melhor retorno esperado por meio de otimização convexa. Qualquer outra combinação destes ativos com relação risco e retorno fora da fronteira eficiente é considerado sub ótimo.

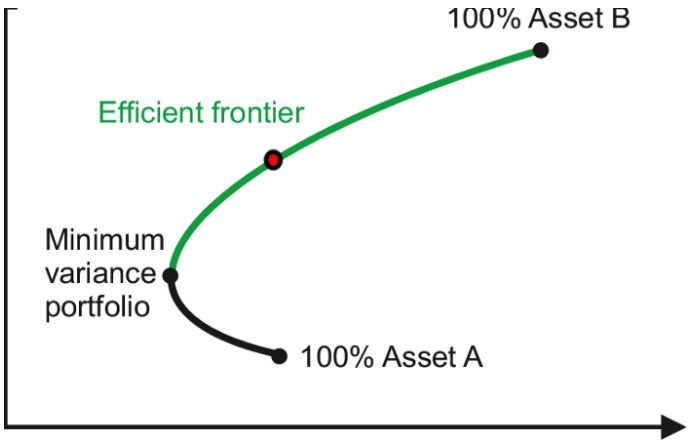


Figura 8.1: Fronteira Eficiente de Markowitz (Retirado de KIENZLE e ANDERSSON [16])

O portfólio de mínima variância ($minVol$) é obtido pela combinação do vetor

de pesos (w) que minimiza o seu produto com a matriz de covariância (Σ), onde o problema de otimização é ilustrado pela equação 8.4.

$$\begin{aligned} \min_{w} \quad & w^T \Sigma w \\ \text{s.t.} \quad & \sum_{j=1}^{n} w_j = 1 \\ & w_j \geq 0, j = 1, .., N \end{aligned} \tag{8.4}$$

A composição do portfólio de mínima variância pode ser visto pela figura 8.2. Note que o algoritmo concentrou aproximadamente 65% da carteira em apenas dois ativos.

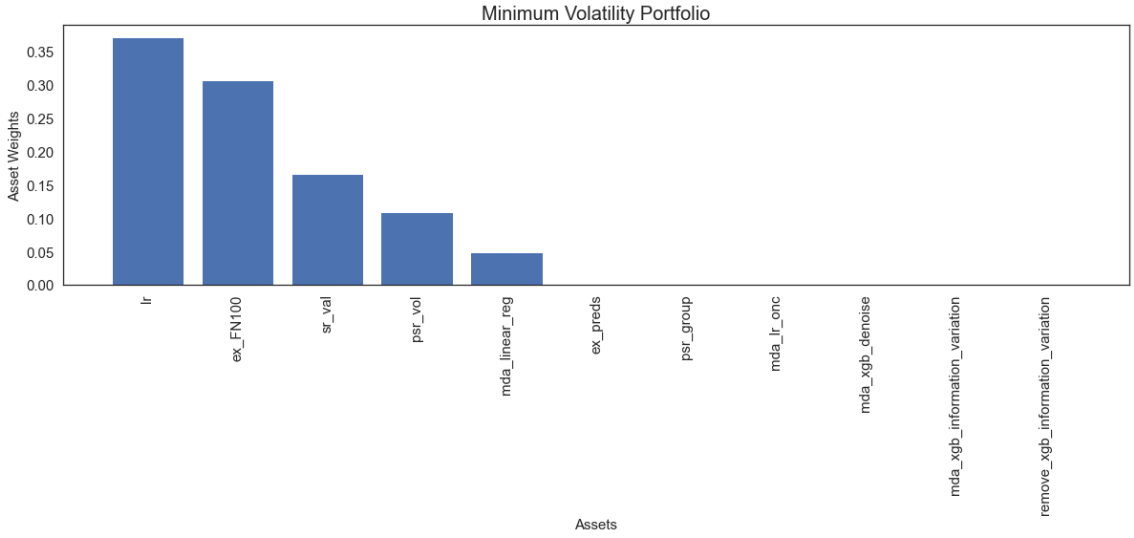


Figura 8.2: Composição Portfólio Mínima Variância

Agora será feito uma aproximação da combinação que reflete a melhor relação risco/retorno do portfólio pela carteira que otimiza o *Sharpe Ratio* (*maxSharpe*), dado pelo problema de otimização descrito na equação 8.5. Note que além de alterar para um problema de maximização, é fácil notar que a função objetivo se assemelha a μ/σ.

$$\begin{aligned} \max_{w} \quad & \frac{\mu^T w}{(w^T \sum w)^{1/2}} \\ \text{s.t.} \quad & \sum_{j=1}^{n} w_j = 1 \\ & w_j \geq 0, j = 1, .., N \end{aligned} \tag{8.5}$$

Note que novamente há uma certa concentração de ativos nesta carteira pela figura 8.3.

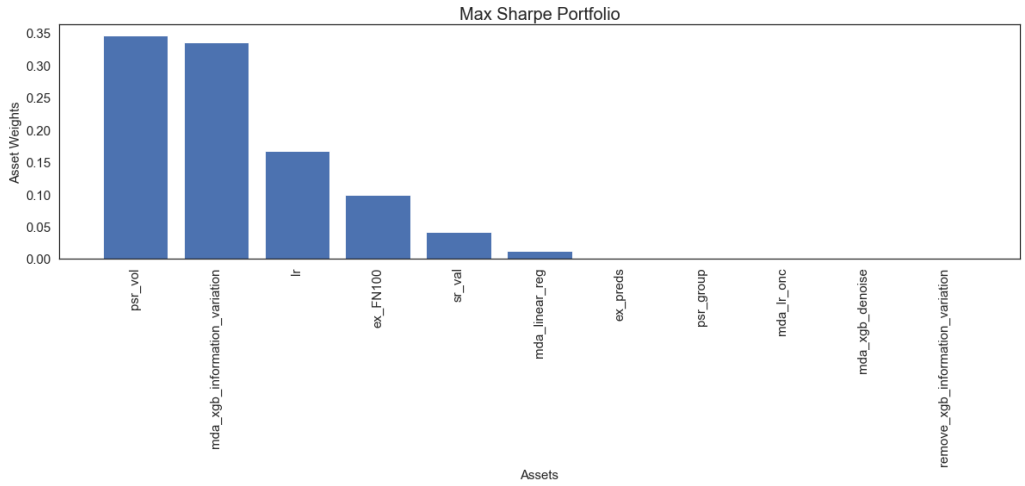


Figura 8.3: Composição Portfólio Sharpe Ótimo

8.2.1 Limitações dos Modelos de Otimização Convexa

Além da concentração dos pesos em poucos ativos já comentada, os modelos de otimização convexa também dependem da estimativa do retorno dos ativos, o que na prática é bastante difícil de se obter com precisão [12]. Além disso a estimativa obtida é claramente dentro da amostra (*in-sample*), não havendo nenhuma garantia de que irão funcionar em dados futuros. Por fim, as soluções apresentadas ainda podem ser instáveis. Na figura 8.4, obtém-se a composição do portfólio de mínima variância com 70 eras, dez a mais que o anterior ilustrado na figura 8.2, note que há uma diferença sensível na distribuição dos pesos.

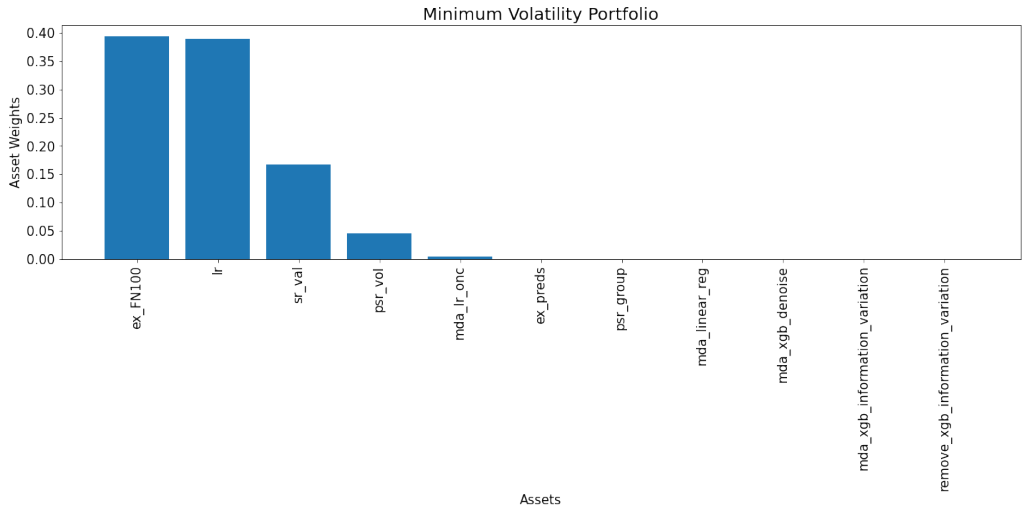


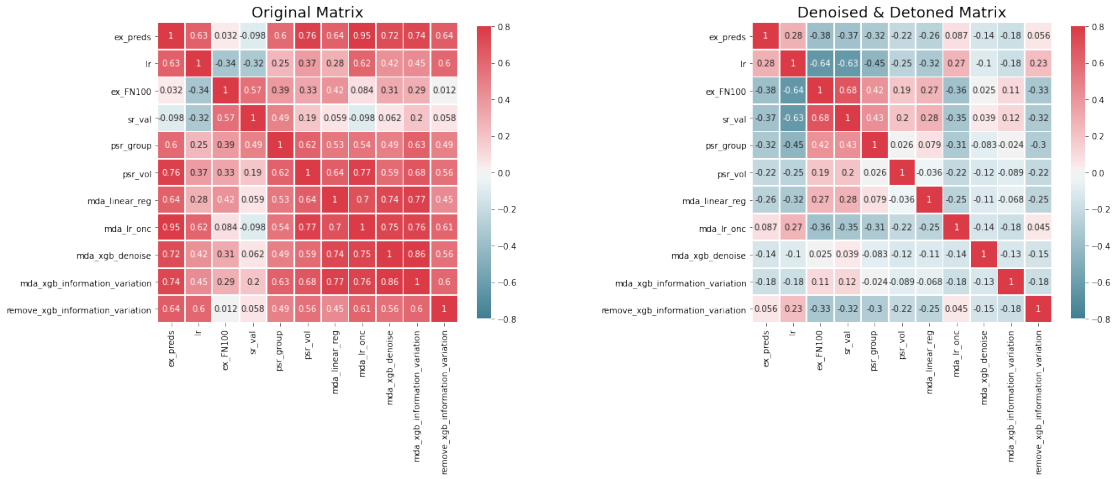
Figura 8.4: Composição Portfólio Mínima Variância 70 eras

A origem desta instabilidade é devida ao cálculo da inversa da matriz de correlação. A presença de ativos mais correlacionados aumenta a probabilidade do deter-

minante da matriz ser zero e do algoritmo não convergir. O apêndice A contêm uma explicação detalhada sobre esse caso conhecido como "Maldição de Markowitz"

Um segundo problema relacionado a matriz de correlação, é que a mesma considera todas as correlações entre os retornos de todos os ativos, como um grafo totalmente conectado. Contudo, nem todos os ativos de um portfólio estão de fato conectados, além disso um erro de estimativa nos retornos de qualquer ativo pode gerar um portfólio totalmente diferente.

Uma forma de mitigar esse problema é removendo o ruído da matriz de correlação (ver seção 6.3.3), além disso ao remover o maior autovalor, equivalente a "componente de mercado", estamos removendo correlações causadas pelo regime (figura 7.1). A figura 8.5 mostra a matriz de correlação do portfólio de modelos antes de depois da transformação.



(a) Matriz Original

(b) Matriz Transformada

Figura 8.5: Remoção da Componente de Mercado

Contudo, é importante lembrar que a matriz da figura 8.5b é uma matriz singular devido a remoção do maior autovalor e portando não podemos utilizá-la em um modelo de otimização convexa.

8.3 Modelos de Paridade de Risco

Antes de prosseguirmos, vamos introduzir mais duas estratégias de otimização de portfólio. Uma conhecida como *Buy & Hold* (*BAH*) onde distribuímos todos os pesos de maneira uniforme e nunca é rebalanceada, permitindo que os pesos se desequilibrem com o passar do tempo. A outra é conhecida como inverso a variância (*IVP*), como o nome diz, os pesos são atribuídos de acordo com a variância, dado pela equação 8.6.

$$w_i = \frac{\Sigma_{i,i}^{-1}}{\sum_{j=1}^{N}(\Sigma_{j,j})^{-1}} \tag{8.6}$$

Esse modelo é muito utilizado como referência para modelos de minimização de risco por não necessitar de uma estimativa dos retornos e fornecer uma distribuição *naive* do risco [12]. Note pela figura 8.6,que ao contrário dos modelos de otimização convexa, os pesos estão distribuídos entre todos os ativos do portfólio.

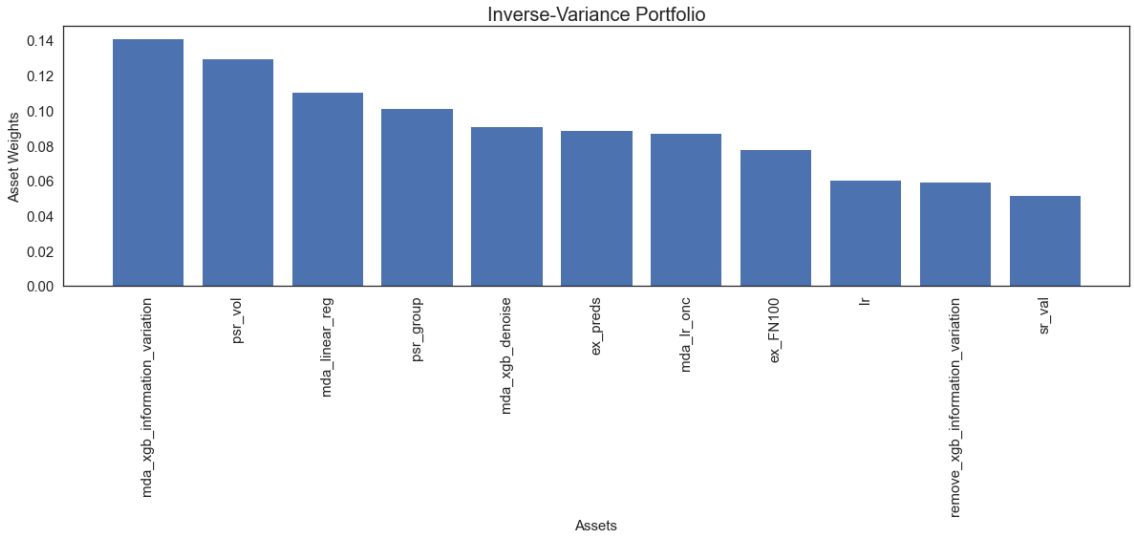


Figura 8.6: Composição Portfólio IVP

8.3.1 O Modelo HRP

O modelo *Hierarquical Parity Risk* (HRP) busca endereçar todas as ressalvas dos algoritmos de otimização convexa citados até aqui . O HRP não requer o cálculo da inversa da matriz de correlação e também não necessita de uma estimativa do retorno dos ativos do portfólio e ao implementar um algoritmo de agrupamento hierárquico (mais detalhes na seção 2.4.3) reduz a complexidade, eliminando conexões desnecessárias [12]. O algoritmo HRP é dividido em três etapas:

1. **Agrupamento Hierárquico**: Agrupamento dos Ativos
2. **Quasi-Diagonalização**: Ordenação das linhas e colunas da matriz
3. **Bissecção Recursiva**: Atribuição dos pesos de cada ativo

Agrupamento Hierárquico

O agrupamento hierárquico é feito sobre a matriz de **covariância** dos ativos, sendo que neste experimento utilizaremos a matriz transformada vista na figura

8.5b. A figura 8.7 ilustra os agrupamentos encontrados pelo algoritmo, é possível perceber que a maioria dos modelos que não foram neutralizados se encontram-se agrupados a esquerda (cor laranja) e os restante a direita (cor verde). Além disso, o artigo original (DE PRADO [57]) do algoritmo HRP utiliza o método *single* como critério de ligação dos agrupamentos, porém de acordo com PAPENBROCK [17], esse critério favorece a concentração dos pesos em poucos ativos e recomenda o uso do critério *ward*. Para mais detalhes ver seção 2.4.3.

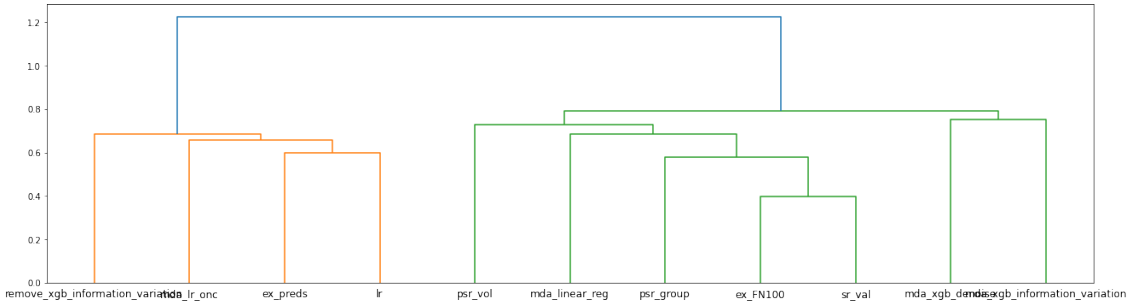


Figura 8.7: HRP - Agrupamento Hierárquico

Quasi-Diagonalização

Também conhecida como seriação de matrizes, esse passo utiliza a ordem dos agrupamentos hierárquicos do passo anterior para reorganizar as linhas e colunas na matriz de covariância de forma que os ativos similares sejam colocados juntos ao longo da diagonal principal e os ativos dissimilares mais distantes. Como a covariância entre os elementos fora da diagonal não são completamente zero, isso é chamado de matriz de covariância quase diagonal. De forma a melhorar a interpretação, transformamos a matriz de covariância em uma matriz de correlação como visto na figura 8.8 seriada pelo HRP.

Bissecção Recursiva

Finalmente, no último passo deve-se atribuir os pesos dos ativos. Inicialmente os pesos são distribuídos igualmente ($w_i = 1; i = 1, ..., N$). Em seguida utilizaremos a ordem das colunas da matriz seriada (V) obtida no segunda etapa para construir uma estrutura de árvore binária, ignorando a estrutura do dendograma da figura 8.7. Os pesos então serão atribuídos seguindo essa nova estrutura hierárquica recursivamente como se segue:

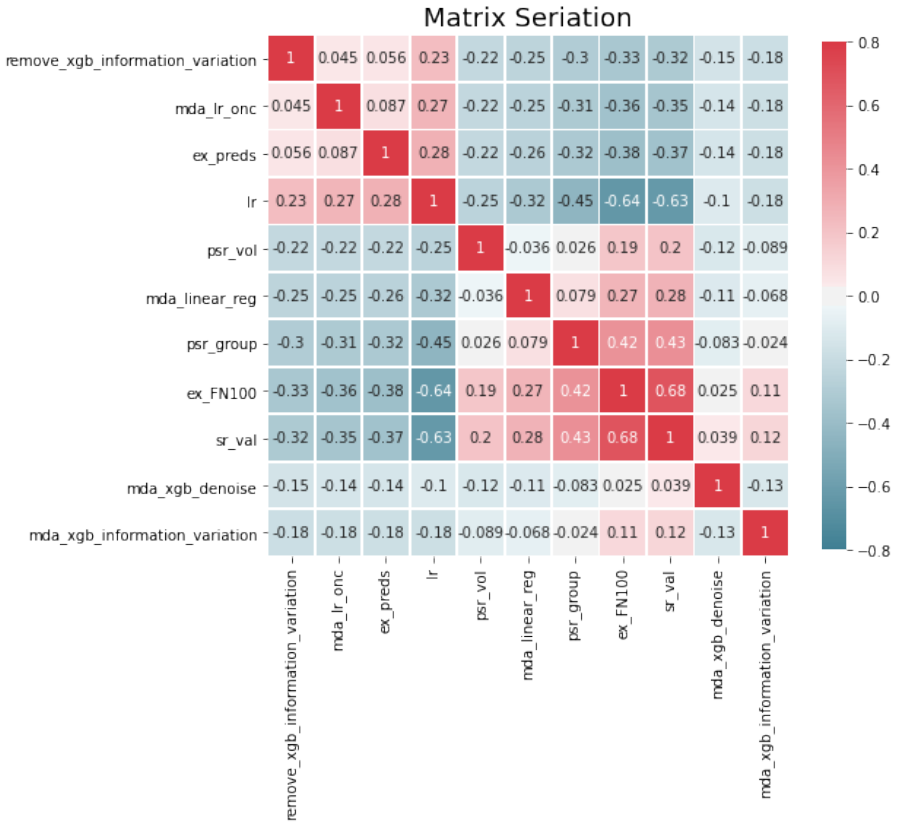


Figura 8.8: HRP - Matriz Seriada

1. **Iteração Entre os Nós da Árvore**: Particionar a matriz V de acordo com a estrutura binária em V_1 e V_2 até o nível mais baixo onde cada partição contém apenas um elemento.

2. **Atribuição dos Pesos em Pares**: Obter os pesos temporários de cada elemento por $w = \frac{diag[V_{1,2}]^{-1}}{Tr(diag[V_{1,2}]^{-1})}$

3. **Determinação das Volatilidades**: $\sigma_{1,2} = \vec{w_{1,2}}^{T} \cdot V_{1,2} \cdot \vec{w_{1,2}}$

4. **Aplicação do fator de peso**: Será atribuído um fator de peso para cada subconjunto de colunas onde $\alpha_1 = 1 - \frac{\sigma_1}{\sigma_1+\sigma_2}$ e $\alpha_2 = 1 - \alpha_1$, onde $\sigma \in [0, 1]$

Note que no passo 2, os pesos atribuídos se assemelham ao inverso da variância (IVP) para cada agrupamento. Finalmente, observe a figura 8.9 onde todos os pesos do portfólio HRP foram atribuídos.

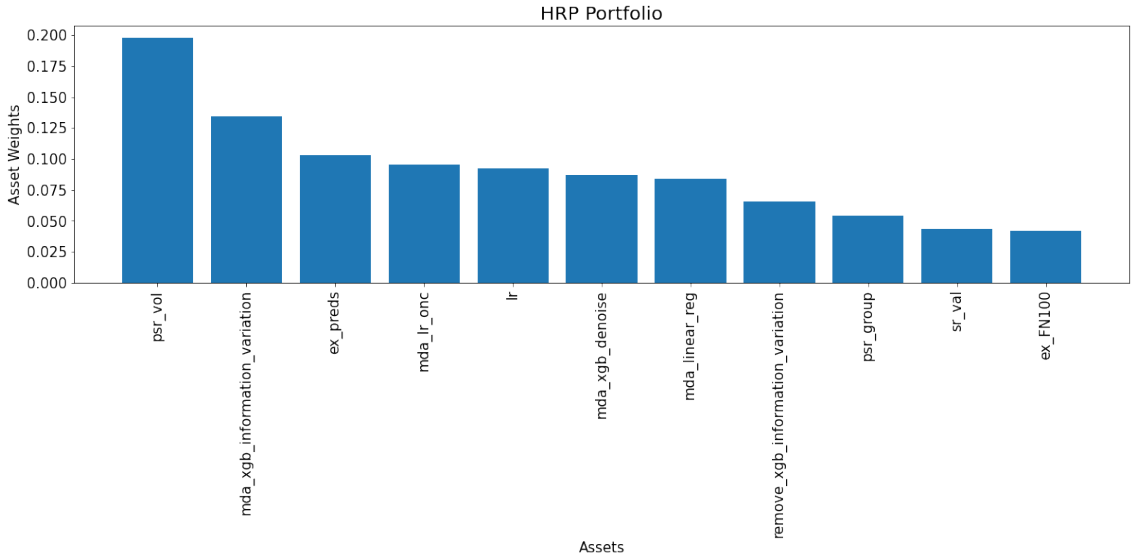


Figura 8.9: Composição Portfólio HRP

Ressalvas HRP

Apesar de resolver diversas limitações dos algoritmos de otimização convexa, modelos hierárquicos possuem o seu próprio conjunto de problemas, onde RAFFINOT [18] faz um compilado de ressalvas sobre o HRP. Conforme mencionado na seção 8.3.1, o HRP utiliza o critério de ligação *single* e esse critério possui um efeito de encadeamento que resulta em agrupamentos com apenas um ativo e consequentemente riscos excessivos [17], o que é mitigado pelo critério *ward*, como foi feito na seção 8.3.1.

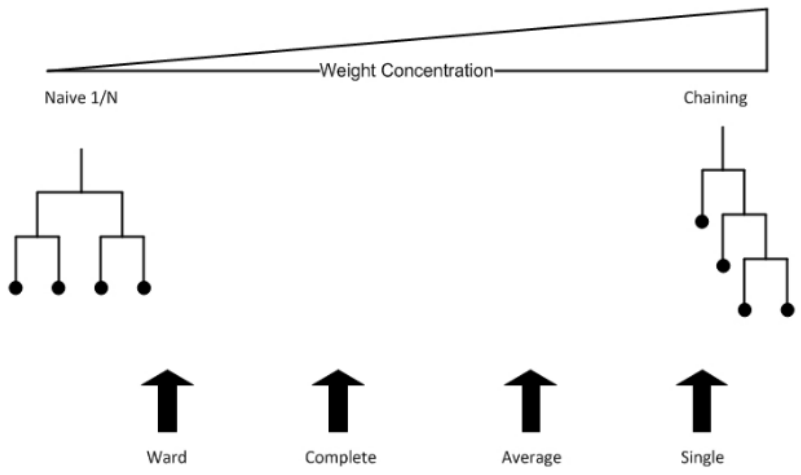


Figura 8.10: Tipos de Ligação (Retirado de PAPENBROCK [17])

A segunda ressalva é que o HRP não respeita a estrutura do dendograma, nem mesmo o número de agrupamentos. O HRP reparte os ativos recursivamente até que o número de folhas seja igual a número de amostras, o que é propenso ao sobreajuste

[18]. A figura 8.12 mostra que o número de agrupamentos altera significativamente a atribuição dos pesos.

Figura 8.11: Alocação por Número de Agrupamentos

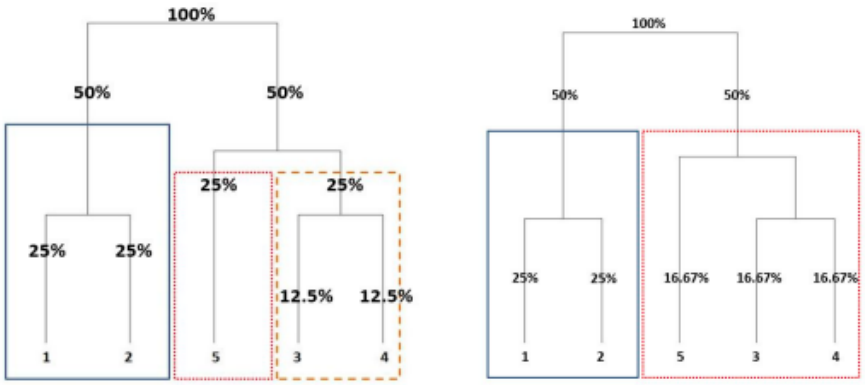
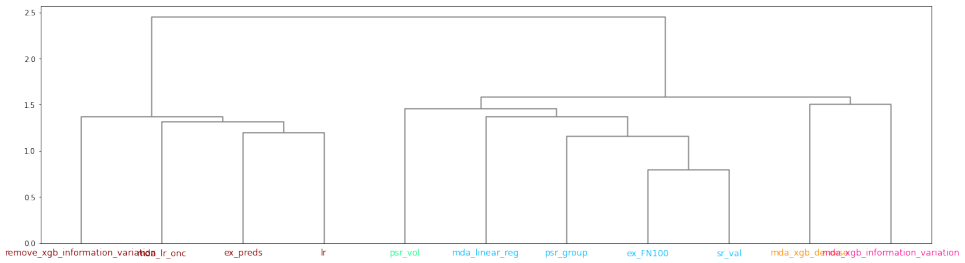


Figura 8.12: Alocação por Número de Agrupamentos (Retirado de RAFFINOT [18])

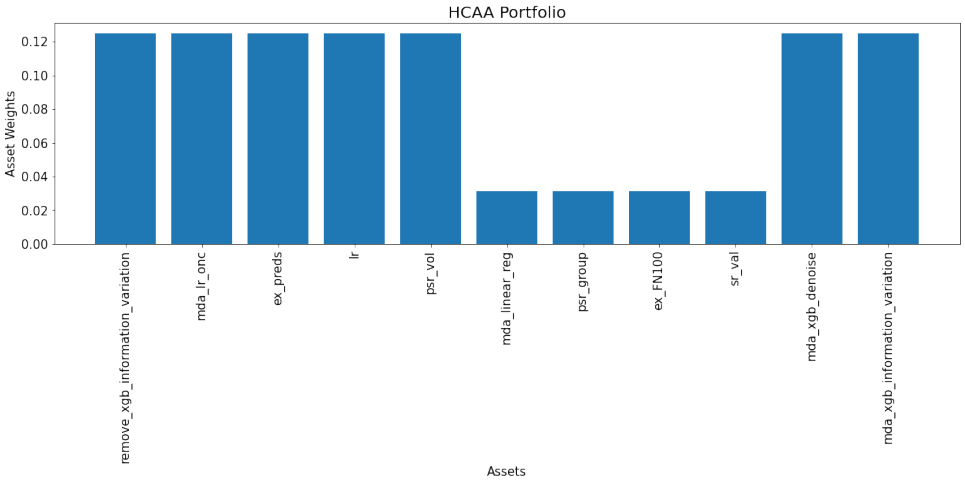
Por fim, RAFFINOT [18] critica o uso da variância como medida de risco dos agrupamentos, uma vez que a mesma advém da matriz de covariância, estando sujeita a erros e ruído. Dessa forma, o autor propõe a utilização de métricas baseadas no risco de perda como o *Expected Shortfall* (CVaR) ou o *Conditional Drawdown at Risk* (CDaR) como fator de peso (α). Essas medidas serão detalhados na seção 8.4.

8.3.2 Os Modelos HCAA e HERC

O modelo *Hierarquical Clustering Asset Allocation* (HCAA) proposto por RAFFINOT [58], segue a hierarquia criada pelo modelo de agrupamento e atribui os pesos de maneira similar a figura 8.12. A figura 8.13a mostra o dendograma que resultou em cinco agrupamentos, onde os pesos de cada agrupamento foram distribuídos da forma $w_C = [0.5, 0.125, 0.125, 0.125, 0.125]$, resultando na figura 8.13b. Apesar desse modelo seguir uma atribuição *naive*, é considerado uma estimativa bastante difícil de superar [58].



(a) HCAA - Agrupamento Hierárquico



(b) Composição Portfólio HCAA

Figura 8.13: Análise Modelo HCAA

A principal diferença do HCAA para o modelo *Hierarquical Equal Risk Contribution* (HERC) está na escolha de uma medida de risco para a atribuição do fator de peso (α) [18]. A estrutura hierárquica é idêntica a figura 8.13a e a figura 8.14 mostra a composição do portfólio HERC utilizando a métrica CDaR, note que os pesos dentro de cada agrupamento também varia de acordo com a medida de risco.

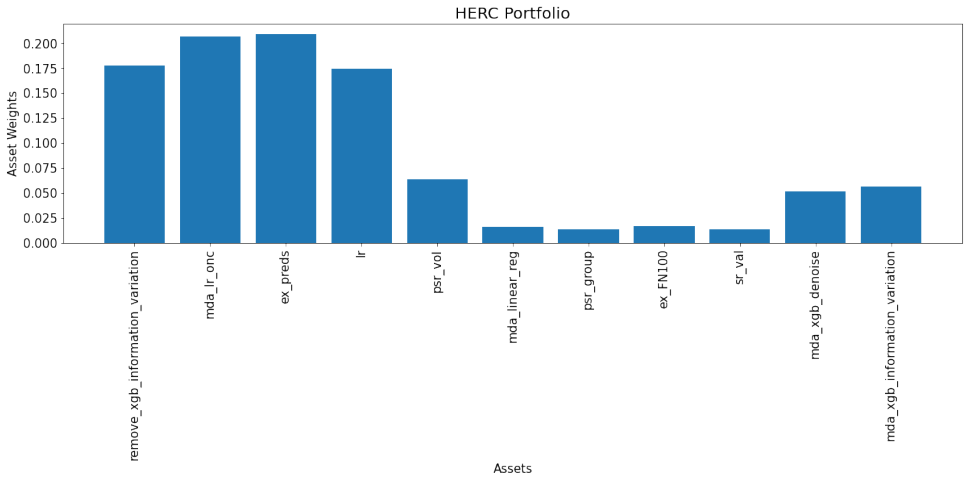


Figura 8.14: Composição Portfólio HERC

8.4 Comparativo dos Algoritmos de Otimização de Portfólio

Nesta seção serão analisados o risco e o retorno obtido pelos portfólios gerados ao longo deste capítulo. É importante salientar que não há nenhuma garantia que as mesmas premissas aceitas para o comportamento de um portfólio de ativos tradicional irá se repetir para este estudo de caso, **esse experimento reflete uma tentativa de verificar se os conceitos são aplicáveis**.

Como já explicitado na seção 8.1, os modelos serão testados nas eras 61 a 120 da base de treinamento. Além disso, serão utilizados os todos os modelos debatidos ao longo deste capítulo, também iremos verificar o algoritmo de retorno eficiente (EffRet), que é 100% alocado no modelo com maior retorno nos dados dentro da amostra (eras 1 a 60). Por fim deve-se analisar o comportamento dos portfólios a partir das seguintes métricas. *Retorno Médio*, *Desvio Padrão*, *Sharpe Ratio* e *Max Drawdown* já explicadas nas seções 4.3.1 e 4.3.2. Além destas, também será usada a métrica *Value at Risk* (VaR) que estima a perda máxima de um investimento para um dado período e intervalo de confiança (α). Na equação 8.7 vamos estimar o VaR para um período (t) com 95% de confiança (*z-score*), $\vec{\mu}$ os retornos da carteira com desvio padrão $\sigma = w^T \Sigma w$.

$$VaR_{t,\alpha} = \vec{\mu} - (\sigma \cdot Z_\alpha) \tag{8.7}$$

Em outras palavras, o VaR representa a perda máxima em um período com 95% de confiança. Por outro lado o *Expected Shortfall* ou *Conditional VaR* (CVaR) quantifica a perda esperada caso esse limite seja ultrapassado. Pela equação 8.8 há a função de densidade da distribuição dos retornos (φ) e Φ^{-1} o quintil para o dado intervalo de confiança.

$$CVaR = \vec{\mu} + \sigma \left(\frac{\varphi(\Phi^{-1}(\alpha))}{\alpha} \right) \tag{8.8}$$

A métrica *Conditional Drawdown at Risk* (CDaR) é análoga ao CVaR, com a única diferença será utilizado o *Drawdown* dado pela equação 4.8 no lugar da série de retornos ($\vec{\mu}$).

Finalmente, tendo em mãos o comparativo entre as performances das carteiras obtidas pelos algoritmos de otimização de portfólio desenvolvidas ao longo de todo o capítulo. É fácil perceber que há uma grande similaridade nos resultados obtidos em ambas as amostras. Denotando que há uma certa monotonicidade na performance relativa entre os modelos que compõem as carteiras, favorecendo modelos de otimização convexa que sobreajustam os dados de treino. Contudo, perceba que há um "dilema"(*trade-off*) entre risco e retorno, partindo do modelo de mínima variância,

até o modelo de máximo retorno. Será feito um novo experimento na seção 9.4, utilizando uma base de dados mais apropriada a fim de selecionar uma carteira de modelos com uma relação risco e retorno apropriada.

	minVol	maxSharpe	bah	IVP	HRP	HCAA	HERC	EffRet
Sharpe	2.3417	2.6308	2.4068	2.4286	2.3999	2.3972	2.3264	1.7555
Mean_Returns	0.0386	0.0490	0.0470	0.0485	0.0470	0.0455	0.0412	0.0596
SD	0.0165	0.0186	0.0195	0.0200	0.0196	0.0190	0.0177	0.0340
Max_DD	-0.0007	-0.0000	-0.0000	-0.0000	-0.0000	-0.0002	-0.0009	-0.0235
VaR	-0.0387	-0.0493	-0.0473	-0.0488	-0.0473	-0.0458	-0.0413	-0.0601
CVaR	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
CDaR	-0.0757	-0.0778	-0.0936	-0.0939	-0.0962	-0.0930	-0.0845	-0.1432

(a) Comparativo - Eras de Treino

	minVol	maxSharpe	bah	IVP	HRP	HCAA	HERC	EffRet
Sharpe	1.6965	1.6393	1.5910	1.5748	1.6898	1.6042	1.6118	1.3408
Mean_Returns	0.0325	0.0393	0.0391	0.0398	0.0393	0.0383	0.0352	0.0523
SD	0.0191	0.0240	0.0246	0.0253	0.0232	0.0239	0.0218	0.0390
Max_DD	-0.0103	-0.0183	-0.0300	-0.0316	-0.0258	-0.0272	-0.0221	-0.0495
VaR	-0.0326	-0.0393	-0.0391	-0.0397	-0.0393	-0.0383	-0.0352	-0.0515
CVaR	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
CDaR	-0.0893	-0.1107	-0.1245	-0.1257	-0.1199	-0.1211	-0.1091	-0.1670

(b) Comparativo - Eras de Teste

Figura 8.15: Comparativo - Modelos de Portfólio

Capítulo 9

Análise dos Resultados

Ao longo deste capítulo será apresentada uma nova base de dados de teste utilizada exclusivamente para a análise de resultados deste capítulo. A seguir será feita uma comparação da performance obtida pelos modelos desenvolvidos até aqui e em seguida aprofundar a análise dos modelos observando-os por categoria e estudando o comportamento dos modelos por regime. Por fim serão analisados os modelos de otimização de portfólio e concluir com uma carteira de modelos constituída.

9.1 Analisando a Base de Teste

No mês de setembro de 2021, um novo conjunto de dados foi disponibilizado pela Numerai[1], contendo as eras referentes as 105 semanas anteriores (aproximadamente 2 anos). Dessa maneira existe a certeza de que não há nenhum tipo de sobreajuste nesses dados, uma vez que em nenhum momento foi feito qualquer tipo de treinamento ou mesmo múltiplas tentativas a fim de aprimorar os resultados. Agora pode-se verificar como esses modelos teriam performado caso estivessem em produção durante esse período (Jul-19/Jun/21).

A principal diferença entre esta base e as outras utilizadas até aqui é que as eras são disponibilizadas semanalmente, ou seja há uma sobreposição no intervalo compreendido entre as eras (mais detalhes na seção 3.2.1). Contudo isso não é um problema aqui, pois não iremos treinar modelos a partir destes dados, pois serão feitas apenas predições. Por se tratar de uma base semanal, utilizou-se essa base ilustrar o comportamento dos regimes, visto na figura 7.1. Outra diferença importante é que 63 das 105 eras são classificadas como difíceis e apenas 42 como fáceis, em oposição a base de treinamento, onde a maioria era classificada como fácil.

[1]Lançamento Dados de Teste.
https://forum.numer.ai/t/super-massive-data-release-deep-dive/4053 (Acessado em 15/01/2022)

9.2 Analisando Todos os Modelos

Os modelos analisados aqui são os mesmos apresentados nas seções de resumo ao final dos capítulos 4, 5, 6 e 7. Além disso, exceto quando mencionado, todos os modelos utilizam como base o modelo de referência seguido alguma transformação (Remoção ou Neutralização de variáveis).

Na figura 9.1 verifica-se a performance de todos os modelos ordenados pelo retorno financeiro ($payout = corr + mmc$), além disso os 3 melhores resultados em cada coluna se encontram na cor verde.

	full_payout	full_sharpe	full_sortino	full_Max_DD
strategy_fs_very_aggressive_dummy	0.0371	0.8748	1.5897	-0.0819
mda_xgb_denoise	0.0361	1.4048	2.9714	-0.0477
strategy_fs_aggressive_dummy	0.0360	0.8432	1.4823	-0.0942
strategy_sfi_mda_dummy	0.0337	1.2224	2.2453	-0.0441
strategy_lr_fn_dummy	0.0336	1.0953	2.0176	-0.0586
strategy_fs_conservative_dummy	0.0335	1.1396	2.3231	-0.0454
strategy_fs_moderate_dummy	0.0335	0.8513	1.3945	-0.0937
mda_xgb_information_variation	0.0333	1.2211	2.8363	-0.0339
mda_linear_reg	0.0322	1.2289	2.5277	-0.0755
strategy_groups_dummy	0.0284	0.7710	1.1823	-0.1764
strategy_xgb_dom_dummy	0.0270	0.6816	0.8288	-0.1074
strategy_metric_dummy	0.0269	0.9170	1.4571	-0.0685
mda_lr_onc	0.0266	0.8470	1.2639	-0.1120
sr_val	0.0238	0.6815	0.7573	-0.1250
strategy_onc_iv_dummy	0.0237	0.7626	0.8916	-0.1101
psr_vol	0.0237	0.8660	1.2133	-0.0631
ex_FN100	0.0231	0.7350	0.7497	-0.1600
remove_xgb_information_variation	0.0216	0.5942	0.5902	-0.2560
ex_preds	0.0207	0.6297	0.6339	-0.1664
psr_group	0.0184	0.5763	0.4463	-0.1557
lr	0.0100	0.2303	-0.0144	-0.3863

Figura 9.1: Comparativo - Todos os Modelos

Note que adicionou-se o sufixo "*dummy*"em todos os modelos que utilizam uma estratégia dependente de regime, pois nesses casos adotou-se a estratégia de atribuir o regime da era anterior para a seguinte (conforme seção 7.5), logo apenas 73% dos casos foram atribuídos corretamente. De toda forma mesmo com essa perda, esses modelos não deixaram de ter os melhores retornos, porém esses erros de atribuição de regime os deixaram mais instáveis, visto que nenhum destes modelos performaram entre os três melhores em nenhuma das outras métricas em análise. Outro ponto

importante a ser mencionado é que os **modelos base** apresentados no capítulo 4, ficaram entre as piores performances entre os modelos analisados.

Agora deve-se analisar a consistência dos modelos observando a performance de cada intervalo de um ano (52 eras) em separado. Perceba que 3 modelos estão entre os 5 melhores retornos financeiros (*payout*) em ambos os intervalos (ver figura 9.2), dando alguma evidência de que esses podem ser os melhores modelos em um eventual terceiro ano.

	full_half1_payout	full_half1_sharpe	full_half1_sortino	full_half1_Max_DD
strategy_fs_aggressive_dummy	0.0326	0.7509	1.3680	-0.0942
strategy_fs_very_aggressive_dummy	0.0319	0.7569	1.3912	-0.0819
mda_xgb_denoise	0.0318	1.3105	2.1398	-0.0477
strategy_fs_moderate_dummy	0.0301	0.7538	1.2116	-0.0937
strategy_sfi_mda_dummy	0.0295	1.2017	2.0477	-0.0324

(a) Comparativo - Primeiro Ano

	full_half2_payout	full_half2_sharpe	full_half2_sortino	full_half2_Max_DD
strategy_fs_very_aggressive_dummy	0.0425	1.0088	1.7567	-0.0797
mda_xgb_information_variation	0.0412	1.3840	4.1158	-0.0302
mda_xgb_denoise	0.0404	1.5351	4.2333	-0.0246
strategy_lr_fn_dummy	0.0404	1.3143	2.7247	-0.0586
strategy_fs_aggressive_dummy	0.0395	0.9471	1.5769	-0.0839

(b) Comparativo - Segundo Ano

Figura 9.2: Comparativo Anual

Por outro lado, a figura 9.3 ilustra um agrupamento hierárquico dos 8 melhores modelos em relação ao retorno financeiro. Note que entre esses modelos existem basicamente dois agrupamentos, um com 2 e o outro com 6 modelos, o ideal seria que houvesse uma maior diversidade entre os modelos de boa performance a fim de uma melhor gestão de risco. Contudo espera-se que o modelo de otimização de portfólio desenvolvido ao longo do capítulo 8 seja capaz de lidar com esse problema.

Um último ponto a ser observado nesta seção é análogo ao desenvolvido na seção 6.3.6, onde calculou-se a correlação entre as métricas obtidas na primeira metade dos dados com o retorno financeiro da segunda metade e assim como na seção 6.3.6, a métrica com maior correlação com os retornos futuros foi o *Sortino Ratio* ($\approx 84\%$), dando indício de que está é uma boa métrica para estimar retornos futuros.

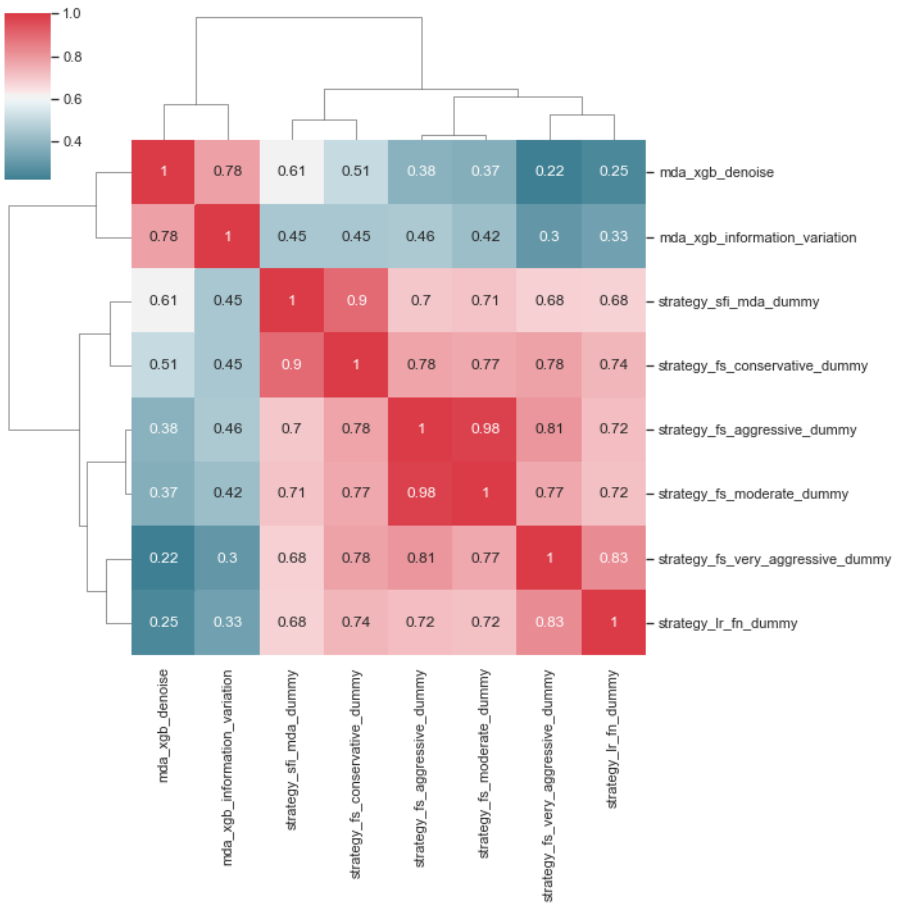


Figura 9.3: Agrupamento Modelos

9.3 Analisando Modelos por Categoria

Ao longo desta seção os modelos estarão separados por categorias de forma a aproximar modelos mais comparáveis.

Modelos Base

A figura 9.4 mostra os resultados dos três modelos base, além dos dois modelos com alternância por regime (**xgb_dom** e **lr_fn**). Observe que o modelo **lr_fn** consegue obter excelentes resultados apenas alternando entre dois modelos supostamente ruins (**lr** e **ex_FN100**). Superando até mesmo um modelo treinado especificamente nas eras de cada regime **xgb_dom**.

Modelos Neutralizados por Grupo

Os modelos com estratégia de neutralização por grupos obtiveram performance apenas razoável se comparados com outras categorias. A se destacar o modelo **psr_groups**, que foi o único modelo a não conseguir superar o modelo de referência, fato que já havia sido classificado como improvável no experimento da seção 5.3.5.

	full_payout	full_sharpe	full_sortino	full_Max_DD
strategy_lr_fn_dummy	0.0336	1.0953	2.0176	-0.0586
strategy_xgb_dom_dummy	0.0270	0.6816	0.8288	-0.1074
ex_FN100	0.0231	0.7350	0.7497	-0.1600
ex_preds	0.0207	0.6297	0.6339	-0.1664
lr	0.0100	0.2303	-0.0144	-0.3863

Figura 9.4: Comparativo - Modelos Base

	full_payout	full_sharpe	full_sortino	full_Max_DD
strategy_groups_dummy	0.0284	0.7710	1.1823	-0.1764
sr_val	0.0238	0.6815	0.7573	-0.1250
ex_preds	0.0207	0.6297	0.6339	-0.1664
psr_group	0.0184	0.5763	0.4463	-0.1557

Figura 9.5: Comparativo - Por Grupos

Modelos Neutralizados por Métricas de Volatilidade

Por outro lado os modelos que neutralizaram as variáveis mais voláteis apresentaram um *Sharpe* melhor que os neutralizados por grupos.

	full_payout	full_sharpe	full_sortino	full_Max_DD
strategy_metric_dummy	0.0269	0.9170	1.4571	-0.0685
psr_vol	0.0237	0.8660	1.2133	-0.0631
ex_preds	0.0207	0.6297	0.6339	-0.1664

Figura 9.6: Comparativo - Por Métricas de Volatilidade

Modelos Neutralizados por Seleção de Variáveis Individuais

Os modelos com seleção de variáveis individuais apresentaram resultados significativamente melhores que os anteriores, porém a seleção individual por regime **sfi_mda_dummy** foi apenas levemente superior ao modelo equivalente feito em todas as eras simultaneamente.

	full_payout	full_sharpe	full_sortino	full_Max_DD
strategy_sfi_mda_dummy	0.0337	1.2224	2.2453	-0.0441
mda_linear_reg	0.0322	1.2289	2.5277	-0.0755
ex_preds	0.0207	0.6297	0.6339	-0.1664

Figura 9.7: Comparativo - Seleção de Variáveis Individual

Modelos Neutralizados por Seleção de Variáveis Agrupadas

Entre os modelos de seleção de variáveis agrupadas, o modelo **onc_iv** destacou-se negativamente, sendo o único modelo com alternância de estratégia por regime que performou pior que os equivalentes que não consideram os regimes. Por outro lado, o modelo **mda_xgb_denoise** foi o segundo melhor modelo em retorno financeiro entre todos deste levantamento e o melhor em *Sharpe* e *Sortino*, mostrando estabilidade e boa performance. Dispondo apenas destas informações, este modelo poderia ser apontado como o melhor candidato para melhor performance futura.

	full_payout	full_sharpe	full_sortino	full_Max_DD
mda_xgb_denoise	0.0361	1.4048	2.9714	-0.0477
mda_xgb_information_variation	0.0333	1.2211	2.8363	-0.0339
mda_lr_onc	0.0266	0.8470	1.2639	-0.1120
strategy_onc_iv_dummy	0.0237	0.7626	0.8916	-0.1101
remove_xgb_information_variation	0.0216	0.5942	0.5902	-0.2560
ex_preds	0.0207	0.6297	0.6339	-0.1664

Figura 9.8: Comparativo - Seleção de Variáveis Agrupadas

Modelos Neutralizados por Seleção Iterativa por Regime

Nesta seção serão avaliados os modelos desenvolvidos na seção 7.3, observe pela figura 9.9 que a estratégia **muito agressiva** obteve o melhor retorno financeiro entre todos os modelos analisados, porém a estratégia **conservadora** pode ser considerada a mais estável, justamente por aceitar perdas relativamente menores quando há erro na atribuição do regime.

	full_payout	full_sharpe	full_sortino	full_Max_DD
strategy_fs_very_aggressive_dummy	0.0371	0.8748	1.5897	-0.0819
strategy_fs_aggressive_dummy	0.0360	0.8432	1.4823	-0.0942
strategy_lr_fn_dummy	0.0336	1.0953	2.0176	-0.0586
strategy_fs_conservative_dummy	0.0335	1.1396	2.3231	-0.0454
strategy_fs_moderate_dummy	0.0335	0.8513	1.3945	-0.0937
strategy_xgb_dom_dummy	0.0270	0.6816	0.8288	-0.1074
ex_preds	0.0207	0.6297	0.6339	-0.1664

Figura 9.9: Comparativo - Seleção de Variáveis Iterativa

9.3.1 Analisando os Erros de Atribuição de Regime

A figura 9.10 ilustra todos os modelos desenvolvidos com estratégias que dependem da atribuição de regime. Realizamos um comparativo entre o retorno financeiro

destes modelos quando os mesmos nunca erram a classificação do regime (*perfect*) com aqueles que adotaram a estratégia *naive* com 73% de acurária (*dummy*).

Perceba que o modelo **muito agressivo** foi o que obteve a maior perda relativa ($\approx 20\%$), como o esperado, apesar de ainda apresentar o melhor retorno financeiro entre todos os modelos. Destaca-se também que o modelo neutralizado por grupos, obteve uma melhora na performance, evidenciando que este não é um bom modelo.

	payout	payout_dummy	diff(%)
strategy_fs_very_aggressive_perfect	0.0464	0.0371	-20.0167
strategy_lr_fn_perfect	0.0419	0.0336	-19.7501
strategy_xgb_dom_perfect	0.0328	0.0270	-17.4776
strategy_fs_moderate_perfect	0.0393	0.0335	-14.9307
strategy_fs_aggressive_perfect	0.0422	0.0360	-14.5969
strategy_fs_conservative_perfect	0.0371	0.0335	-9.8115
strategy_sfi_mda_perfect	0.0373	0.0337	-9.6803
strategy_metric_perfect	0.0285	0.0269	-5.6122
strategy_onc_iv_perfect	0.0249	0.0237	-4.6807
strategy_groups_perfect	0.0271	0.0284	4.8227

Figura 9.10: Perda de Performance - Atribuição de Regime

Em seguida observe a performance dos melhores modelos por regime. A figura 9.11a mostra que a regressão linear possui a melhor performance nas eras fáceis, contudo para uma análise mais precisa, devemos observar as eras **classificadas** como fáceis e assim contabilizar os erros de atribuição. A figura 9.11b mostra que há uma diferença significativa no ranqueamento.

	easy_payout	easy_sharpe	easy_sortino	easy_Max_DD
lr	0.0420	1.3062	4.9323	-0.0145
mda_xgb_information_variation	0.0385	1.3717	3.4083	-0.0277
strategy_fs_very_aggressive_dummy	0.0384	0.7189	1.3826	-0.0819
strategy_groups_dummy	0.0374	0.9758	2.1706	-0.0412
strategy_fs_aggressive_dummy	0.0357	0.6896	1.2192	-0.0942

(a) Comparativo - Eras Fáceis

	easy_payout	easy_sharpe	easy_sortino	easy_Max_DD
strategy_fs_aggressive_dummy	0.0452	0.8864	1.7615	-0.0942
strategy_fs_very_aggressive_dummy	0.0416	0.7813	1.4943	-0.0793
strategy_fs_moderate_dummy	0.0404	0.8875	1.5905	-0.0937
mda_xgb_information_variation	0.0400	1.6501	6.6930	-0.0098
strategy_groups_dummy	0.0386	1.1350	2.1509	-0.0484

(b) Comparativo - Eras Classificadas como Fáceis

Figura 9.11: Comparativo - Performance Eras Fáceis

Analogamente, o mesmo acontece nas eras difíceis, onde o modelo neutralizado (**ex_FN100**) deixa de ser o melhor modelo.

	hard_payout	hard_sharpe	hard_sortino	hard_Max_DD
ex_FN100	0.0419	2.0467	10.3443	-0.0121
strategy_fs_conservative_dummy	0.0401	1.5231	3.9740	-0.0363
sr_val	0.0401	1.2936	2.7253	-0.0862
strategy_sfi_mda_dummy	0.0392	1.6040	3.3906	-0.0403
mda_xgb_denoise	0.0378	1.5619	4.6847	-0.0171

(a) Comparativo - Eras Difíceis

	easy_payout	easy_sharpe	easy_sortino	easy_Max_DD
strategy_sfi_mda_dummy	0.0371	1.6225	4.3777	-0.0246
mda_xgb_denoise	0.0364	1.4147	3.6152	-0.0245
sr_val	0.0349	1.0431	1.8504	-0.0831
strategy_fs_conservative_dummy	0.0348	1.2980	3.1416	-0.0326
ex_FN100	0.0345	1.2992	2.6493	-0.0365

(b) Comparativo - Eras Classificadas como Difíceis

Figura 9.12: Comparativo - Performance Eras Difíceis

Agora o objetivo é compreender que tipo de erro de classificação de regime é mais danoso ao retorno dos modelos. Para isso, comparamos os modelos que sempre acertam (*perfect*) com aqueles que sempre erram (*minus*), nas estratégias **conservadora** e **agressiva**. Perceba que em ambos os casos a perda relativa nas eras fáceis (figura 9.13a) é maior do que nas eras difíceis (9.13b). O que leva a interpretar que um erro nas eras fáceis é pior do que nas eras difíceis ou em suma, em caso de dúvida é menos pior classificar como fácil.

	easy_payout	easy_payout_minus	diff(%)
strategy_fs_aggressive_perfect	0.0511	-0.0096	-118.7045
strategy_fs_conservative_perfect	0.0314	0.0018	-94.2820

(a) Comparativo - Eras Fáceis

	hard_payout	hard_payout_minus	diff(%)
strategy_fs_aggressive_perfect	0.0362	0.0064	-82.3071
strategy_fs_conservative_perfect	0.0409	0.0184	-54.9802

(b) Comparativo - Eras Difíceis

Figura 9.13: Comparativo - Perda de Performance por Regime

Por fim, o retorno financeiro dos modelos nas eras fáceis possui correlação igual a $\rho_{easy} = -0.33$ com o retorno nas eras difíceis, contudo a autocorrelação dos retornos

($AR1$) possui uma correlação de $\rho_{AR1} = -0.70$ com o retorno dos modelos ao considerar todas as eras, comprovando que modelos que dependem exageradamente de um regime possuem uma performance ruim no geral (ver figura 9.14). Esse conceito foi abordado na seção 5.4.1.

	full_payout	full_sharpe	full_sortino	full_Max_DD	full_AR1
lr	0.0100	0.2303	-0.0144	-0.3863	0.7353
remove_xgb_information_variation	0.0216	0.5942	0.5902	-0.2560	0.7151
ex_preds	0.0207	0.6297	0.6339	-0.1664	0.6789
psr_vol	0.0237	0.8660	1.2133	-0.0631	0.6515
sr_val	0.0238	0.6815	0.7573	-0.1250	0.6402
ex_FN100	0.0231	0.7350	0.7497	-0.1600	0.6396

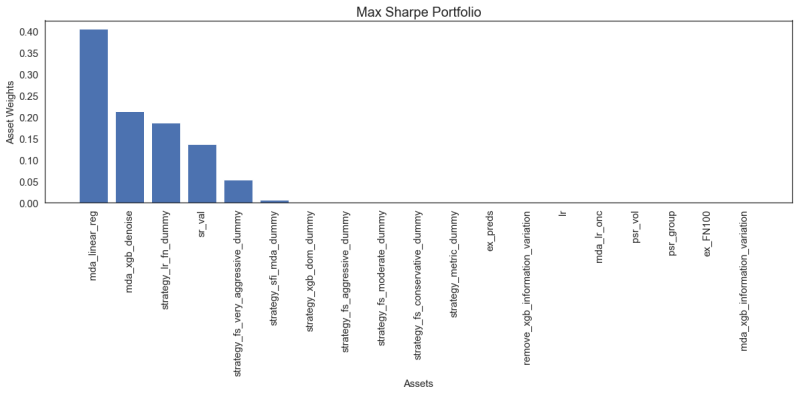
Figura 9.14: Correlação AR1

9.4 Modelos de Otimização de Portfólio

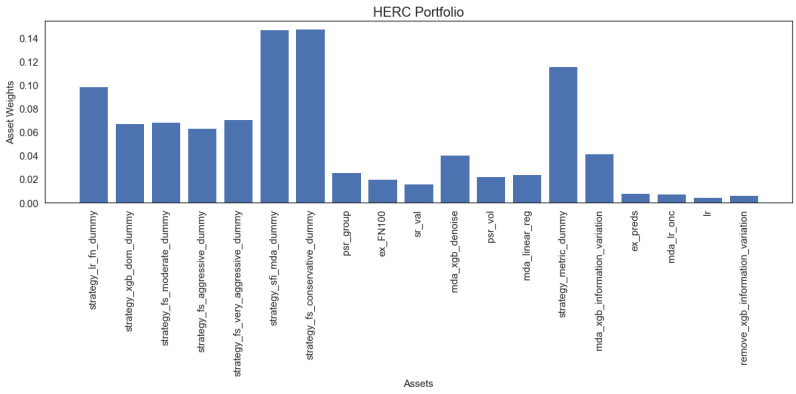
Em relação aos modelos de otimização de portfólio desenvolvidos no capítulo 8, serão utilizadas em um experimento as primeiras 80 eras desta mesma base de teste para treinar os modelos e a últimas 25 para teste, período que compreende o primeiro semestre de 2021. Também é necessário destacar que removemos os modelos **strategy_groups_dummy** e **strategy_onc_iv_dummy** que apresentaram resultados ruins durante a análise de resultados, totalizando um portfólio de 19 modelos.

Os problemas relativos aos modelos de otimização convexa comentados ao longo do capítulo 8, permaneceram neste novo experimento. Veja que o modelo de otimização do *Sharpe* concentrou a carteira em poucos ativos (figura 9.15a). Já o algoritmo HERC encontrou 10 agrupamentos em 19 possíveis, denotando uma certa diversidade nos modelos do portfólio. Além disso os pesos foram distribuídos de maneira mais equilibrada.

Em seguida observe as métricas que avaliam a relação risco e retorno das carteiras nesse período. Na amostra de treino o algoritmo *maxSharpe* obteve bons resultados, o que não se repetiu na base de teste, não sendo capaz de superar nem a distribuição uniforme dos pesos (BAH). Note também que mesmo na base de teste o algoritmo de mínima volatilidade ainda obteve o melhor resultado nas métricas de risco, assim como o algoritmo de retorno eficiente obteve o melhor retorno em ambas as bases de dados. É possível interpretar que os modelos de otimização convexa funcionaram relativamente bem neste contexto porque modelos com boa performance tendem a se manter melhores que os demais ao longo do tempo, o que não pode ser afirmado em um portfólio de ativos financeiros tradicional.



(a) Composição Portfólio - Max Sharpe



(b) Composição Portfólio - HERC

Figura 9.15: Composição Modelos de Portfólio

	minVol	maxSharpe	bah	IV	HRP	HCAA	HERC	EffRet
Sharpe	1.1935	1.6910	1.1846	1.2770	1.1665	1.2467	1.1873	0.9026
Mean_Returns	0.0213	0.0334	0.0266	0.0275	0.0265	0.0261	0.0244	0.0369
SD	0.0179	0.0198	0.0224	0.0215	0.0227	0.0209	0.0205	0.0409
Max_DD	-0.0250	-0.0217	-0.0348	-0.0327	-0.0342	-0.0286	-0.0281	-0.0819
VaR	-0.0212	-0.0334	-0.0264	-0.0274	-0.0264	-0.0259	-0.0243	-0.0359
CVaR	-0.0250	0.0000	-0.0310	-0.0327	-0.0309	-0.0286	-0.0281	-0.0606
CDaR	-0.0627	-0.0809	-0.1086	-0.0984	-0.1116	-0.0976	-0.0947	-0.2387

(a) Comparativo - Eras de Treino

	minVol	maxSharpe	bah	IV	HRP	HCAA	HERC	EffRet
Sharpe	1.4347	1.4026	1.4401	1.3733	1.4025	1.4116	1.3551	0.8039
Mean_Returns	0.0235	0.0294	0.0323	0.0318	0.0335	0.0314	0.0301	0.0378
SD	0.0164	0.0210	0.0224	0.0231	0.0239	0.0222	0.0222	0.0471
Max_DD	-0.0000	-0.0233	-0.0111	-0.0162	-0.0131	-0.0160	-0.0159	-0.0669
VaR	-0.0235	-0.0295	-0.0322	-0.0317	-0.0333	-0.0313	-0.0301	-0.0360
CVaR	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0516
CDaR	-0.0538	-0.0599	-0.0893	-0.0872	-0.0974	-0.0815	-0.0769	-0.1995

(b) Comparativo - Eras de Teste

Figura 9.16: Comparativo - Modelos de Portfólio

Agora, observe a figura 9.17 e note que o modelo de risco eficiente obteve resultados bastante instáveis ao longo de 2021, já o modelo de mínima volatilidade obteve uma performance muito abaixo dos demais. O restante dos modelos obteve um retorno acumulado bastante similar, mas com aspectos monotônicos.

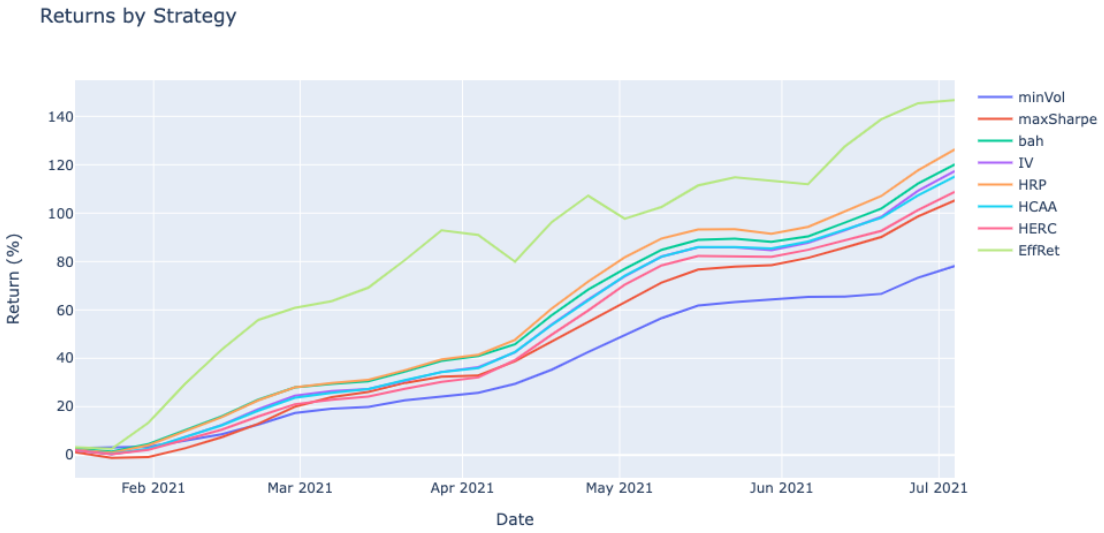


Figura 9.17: Performance Portfólio 2021

Para finalizar esse experimento o modelo HRP foi o escolhido e realizou-se um novo balanceamento utilizando todas as 105 eras. A figura 9.18 ilustra a composição final do portfólio, estando razoavelmente equilibrada, mas ainda favorecendo bons modelos em relação aos retornos, como desejado.

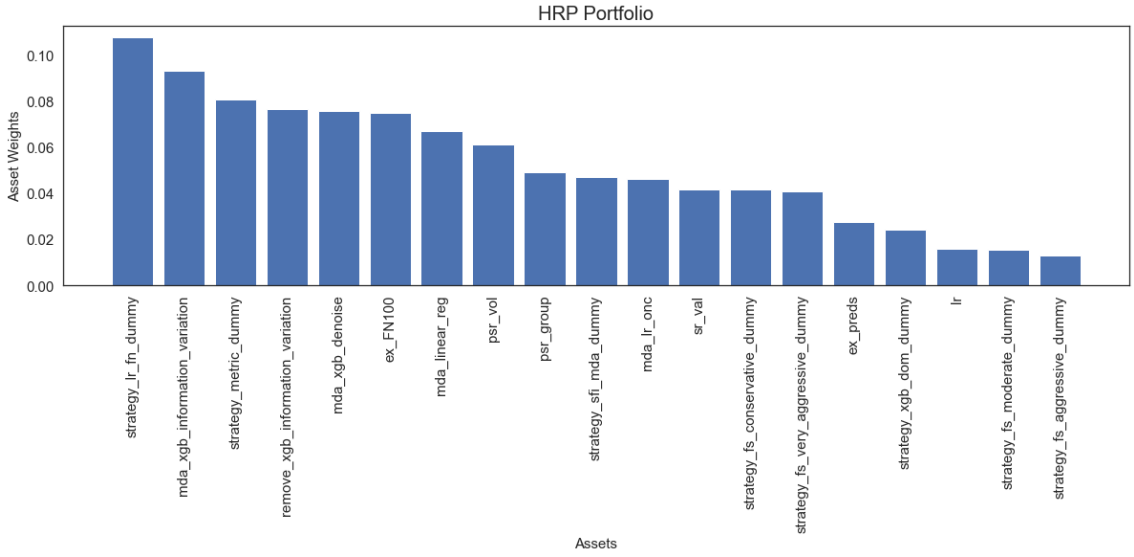


Figura 9.18: Composição Portfólio Final

Capítulo 10

Considerações Finais

Ao longo deste trabalho foi feita uma longa apresentação do problema e dos dados utilizados, necessária para entender o contexto que estamos inseridos. Em seguida definiu-se um modelo que foi utilizado como referência ao longo de todo o trabalho, cujo o objetivo principal era superá-lo com consistência.

Introduziu-se uma técnica capaz de remover a porção linear das predições geradas pelo estimador que comprovou-se mais eficaz que uma remoção completa da variável. Em seguida utilizou-se um método que leva em consideração todo o conjunto de estratégias testadas para avaliar com a robustez necessária se a estratégia escolhida é ou não capaz de superar o modelo de referência. Dois experimentos independentes foram desenvolvidos com esse fim, porém ambos foram rejeitados.

Foi realizado um extenso estudo de técnicas de seleção de variáveis, desde as mais simples analisando-as de maneira unitária e foram utilizados diversos métodos para medir os a eficácia destas técnicas, sem precisar avaliar as predições. Foram estudados métodos mais robustos com variáveis agrupadas a fim de mitigar a multicolinearidade e tentou-se otimizar os resultados utilizando técnicas de remoção de ruído da matriz de correlação. Foram avaliadas medidas de dependência não lineares com uso de teoria da informação. Os resultados tiveram uma sutil evolução cronológica.

Na sequência buscou-se entender em quais momentos uma variável contribui ou atrapalha o estimador. É possível afirmar que existem ao menos 2 regimes onde um subgrupo de variáveis ou outro funcionam melhor, obtendo resultados expressivos. Contudo, não foi possível desenvolver um método capaz de prever o regime futuro que superasse uma heurística simples.

Tendo em mãos um conjunto de modelos de altamente voláteis, decidiu-se tratá-los de maneira conjunta e se apoiar na literatura de algoritmos de paridade de risco para otimização de portfólio, afim de criar uma carteira de modelos com uma relação risco retorno satisfatória. Contudo o dilema previsto na relação risco e retorno se manteve entre as soluções testadas.

Por fim, realizou-se uma extensa análise dos resultados obtidos pelos modelos em um conjunto de dados nunca antes visto, reforçando uma das premissas deste estudo que é mitigar o sobreajuste das estratégias e foi investigado quais tipos de risco valem a pena correr e o trabalho foi concluído com a composição do portfólio de modelos obtido por uma das soluções testadas.

Acredita-se que o trabalho deixa ao menos duas contribuições. A primeira é a extensão feita ao algoritmo ONC (ver seção 6.3.2.1), que mitiga o risco sobreavaliar a importância de variáveis com pouca importância. A outra é a necessidade de identificar regimes que fazem determinados conjuntos de indicadores funcionarem ou deixarem de funcionar ao longo do tempo, o que é bastante útil, especialmente em problemas envolvendo séries temporais não estacionárias.

Acredita-se que o tema de seleção de variáveis neste estudo de caso ainda pode ser ampliado envolvendo técnicas baseadas em redes neurais como *Autoencoders* que ficaram fora do escopo deste estudo. Contudo o autor do presente estudo não acredita que investigações sobre a distribuições das variáveis poderá contribuir para aprimorar a atribuição de regimes.

Finalmente, é preciso afirmar que não foi possível provar a partir das técnicas de seleção de variáveis, que o modelo de referência pode ser superado com consistência, apesar deste resultado ser obtido em grande parte das observações. Apesar da promessa de excelentes resultamos imagina-se que a busca por detecção de regime futuro tende a ser bastante infrutífera. Acredita-se que a investigação de modelos capazes de gerar uma maior porção não linear do sinal (*Feature Neutral Mean*), como a melhor candidata a apresentar resultados melhores e suficientemente estáveis.

Referências Bibliográficas

[1] EVSUKOFF, A. G. *Ensinando máquinas*. 1º ed. Rio de Janeiro, RJ, UFRJ, 2018.

[2] JAMES, G., WITTEN, D., HASTIE, T., et al. *An Introduction to Statistical Learning: with Applications in R*. New York, NY, Springer, 2013.

[3] DOMINGOS, P. "A few useful things to know about machine learning", *Communications of the ACM*, v. 55, n. 10, pp. 78–87, 2012.

[4] MISRA, S., LI, H. "Chapter 9 - Noninvasive fracture characterization based on the classification of sonic wave travel times". In: *Machine Learning for Subsurface Characterization*, Gulf Professional Publishing, pp. 243–287, Houston, TX, 2020.

[5] QIAN, B., SU, J., WEN, Z., et al. "Orchestrating Development Lifecycle of Machine Learning Based IoT Applications: A Survey", *ACM Computing Surveys*, v. 53, 2019.

[6] TIBSHIRANI, R., WALTHER, G., HASTIE, T. "Estimating the number of clusters in a data set via the gap statistic", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, v. 63, n. 2, pp. 411–423, 2001.

[7] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., et al. "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, v. 12, pp. 2825–2830, 2011.

[8] RASCHKA, S. "MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack", *The Journal of Open Source Software*, v. 3, n. 24, abr. 2018.

[9] BERGSTRA, J., BENGIO, Y. "Random search for hyper-parameter optimization", *Journal of Machine Learning Research*, v. 13, pp. 281–305, 2012.

[10] FRAZIER, P. I. "A Tutorial on Bayesian Optimization". 2018. Disponível em: <`http://arxiv.org/abs/1807.02811`>.

[11] DODGE, Y. *The Concise Encyclopedia of Statistics*. 1º ed. New York, NY, Springer New York, 2008.

[12] DE PRADO, M. L. *Advances in Financial Machine Learning*. 1º ed. Hoboken, New Jersey, Wiley Publishing, 2018.

[13] LI, Y., TURKINGTON, D., YAZDANI, A. "Beyond the Black Box: An Intuitive Approach", *The Journal of Financial Data Science*, v. 2, n. 1, 2020.

[14] DE PRADO, M. L. *Machine Learning for Asset Managers*. 1º ed. Cambridge, United Kingdom, Cambridge University Press, 2020.

[15] SALKIND, N. "Encyclopedia of Measurement and Statistics". 2007. Disponível em: <https://sk.sagepub.com/reference/statistics>.

[16] KIENZLE, F., ANDERSSON, G. "Efficient multi-energy generation portfolios for the future". In: *Proceedings 4th Annual Carnegie Mellon Conference on the Electricity Industry, Pittsburgh, USA*, Pittsburgh, PA, 01 2008.

[17] PAPENBROCK, J. *Asset Clusters and Asset Networks in Financial Risk Management and Portfolio Optimization*. Tese de Doutorado, Karlsruher Institut für Technologie (KIT), 2011.

[18] RAFFINOT, T. "The Hierarchical Equal Risk Contribution Portfolio", *SSRN Electronic Journal*, 01 2018.

[19] DAVENPORT, T. H., PATIL, D. J. "Data Scientist: The Sexiest Job of the 21st Century". Oct 2012. Disponível em: <https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century>.

[20] SINGH, S., SHARMA, S. "Forecasting Stock Price Using Partial Least Squares Regression". In: *2018 8th International Conference on Cloud Computing, Data Science & Engineering*, pp. 587–591, 2018.

[21] CRAIB, R. "Building The Last Hedge Fund - Introducing Numerai Signals". Out 2020. Disponível em: <https://medium.com/numerai/building-the-last-hedge-fund-introducing-numerai-signals-12de26dfa69c>.

[22] DE PRADO, M. L., FABOZZI, F. J. "Crowdsourced Investment Research Through Tournaments", *The Journal of Financial Data Science*, v. 2, n. 1, pp. 86–93, 2020.

[23] DE PRADO, M. L. "The 10 reasons most machine learning funds fail", *Journal of Portfolio Management*, v. 44, n. 6, pp. 120–133, 2018.

[24] DIXON, M. F., HALPERIN, I., BILOKON, P. *Machine Learning in Finance*. 1º ed. New York, NY, Springer, 2016.

[25] LIU, T. Y. "Learning to rank for Information Retrieval", *Foundations and Trends in Information Retrieval*, v. 3, n. 3, pp. 225–231, 2009.

[26] BURGES, C., SHAKED, T., RENSHAW, E., et al. "Learning to rank using gradient descent", *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96, 2005.

[27] HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. *The Elements of Statistical Learning*. New York, NY, USA, Springer New York Inc., 2001.

[28] CHEN, T., GUESTRIN, C. "XGBoost: A scalable tree boosting system", *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, v. 13-17-August-2016, pp. 785–794, 2016.

[29] KE, G., MENG, Q., FINLEY, T., et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, p. 3149–3157, Red Hook, NY, USA, 2017. Curran Associates Inc.

[30] NAKAMOTO, Y. "A Short Introduction to Learning to Rank", *IEICE Transactions on Information and Systems*, v. E94-D, n. 1, pp. 1–2, 2011.

[31] BURGES, C. J. C. *From RankNet to LambdaRank to LambdaMART: An Overview*. Relatório técnico, Microsoft Research Technical Report MSR-TR-2010-82, 2010.

[32] ZHOU, Z.-H. *Ensemble Methods: Foundations and Algorithms*. 1st ed. New York, NY, Chapman & Hall CRC, 2012.

[33] CERQUEIRA, V., TORGO, L., MOZETIC, I. "Evaluating time series forecasting models: An empirical study on performance estimation methods". 2019.

[34] CAWLEY, G. C., TALBOT, N. L. "On over-fitting in model selection and subsequent selection bias in performance evaluation", *Journal of Machine Learning Research*, v. 11, pp. 2079–2107, 2010.

[35] SNOEK, J., LAROCHELLE, H., ADAMS, R. P. "Practical Bayesian optimization of machine learning algorithms", *Advances in Neural Information Processing Systems*, v. 4, pp. 2951–2959, 2012.

[36] WILDER, J. *New Concepts in Technical Trading Systems*. Indianápolis, Indiana, Trend Research, 1978.

[37] FRIEDMAN, J. "Greedy Function Approximation: A Gradient Boosting Machine", *Annals of Statistics*, v. 29, pp. 1189–1232, 2001.

[38] GREENE, W. H. *Econometric Analysis*. 5th ed. New York, NY, Pearson Education, 2003.

[39] LO, A. W. "The Statistics of Sharpe Ratios", *Financial Analysts Journal*, v. 58, n. 4, pp. 36–52, 2002.

[40] BAILEY, D. H., DE PRADO, M. L. "The sharpe ratio efficient frontier", *Journal of Risk*, v. 15, n. 2, pp. 3–44, 2012.

[41] ALEXANDER, C., SHEEDY, E. *The Professional risk Managers' Handbook:A Comprehensive Guide to Current Theory and Best Practices*. New York, NY, PRMIA Publications, 2005.

[42] BAILEY, D. H., BORWEIN, J. M., LÓPEZ DE PRADO, M., et al. "Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance", *Notices of the American Mathematical Society*, v. 61, n. 5, pp. 458, 2014.

[43] BAILEY, D. H., LÓPEZ DE PRADO, M. "The deflated Sharpe ratio: Correcting for selection bias, backtest overfitting, and non-normality", *Journal of Portfolio Management*, v. 40, n. 5, pp. 94–107, 2014.

[44] DE VIÉVILLE, R. B., GELRUBIN, R., LINDET, E., et al. *An Alternative Portfolio Theory*. Relatório técnico, KeyQuant, 2017.

[45] WASSERMAN, L. *All of statistics : a concise course in statistical inference*. New York, Springer, 2010.

[46] LUNDBERG, S. M., LEE, S.-I. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., pp. 4765–4774, Red Hook, NY, 2017.

[47] NORI, H., JENKINS, S., KOCH, P., et al. "InterpretML: A Unified Framework for Machine Learning Interpretability", *arXiv preprint arXiv:1909.09223*, 2019.

[48] ROUSSEEUW, P. J. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis", *Journal of Computational and Applied Mathematics*, v. 20, pp. 53–65, 1987.

[49] DE PRADO, M. L. "A DATA SCIENCE SOLUTION TO THE MULTIPLE-TESTING CRISIS IN FINANCIAL RESEARCH Marcos", *The Journal of Financial Data Science Winter*, v. 1, 2019.

[50] LALOUX, L., CIZEAU, P., BOUCHAUD, J.-P., et al. "Random matrix theory and financial correlations", *International Journal of Theoretical and Applied Finance*, v. 3, pp. 391, 2000.

[51] DEVROYE, L., GYÖRFI, L. *Distribution and Density Estimation*. Vienna, Springer Vienna, 2002.

[52] COVER, T. M., THOMAS, J. A. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA, Wiley-Interscience, 2006.

[53] MEILĂ, M. *Comparing Clusterings by the Variation of Information*. Berlin, Heidelberg, Springer Berlin Heidelberg, 2003.

[54] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., et al. "Generative adversarial nets". In: *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[55] JOHNSON, J., DOUZE, M., JÉGOU, H. "Billion-scale similarity search with GPUs", *arXiv preprint arXiv:1702.08734*, 2017.

[56] MARKOWITZ, H. "PORTFOLIO SELECTION*", *The Journal of Finance*, v. 7, n. 1, pp. 77–91, 1952.

[57] DE PRADO, M. L. "Building Diversified Portfolios that Outperform Out of Sample", *The Journal of Portfolio Management*, v. 42, n. 4, pp. 59–69, 2016.

[58] RAFFINOT, T. "Hierarchical Clustering-Based Asset Allocation", *The Journal of Portfolio Management*, v. 44, n. 2, pp. 89–99, 2017.

Apêndice A

A Instabilidade da Matriz de Correlação

Esse trecho foi retirado integralmente de DE PRADO [14], que esclarece a causa da instabilidade da matriz de correlação, também conhecida como "Maldição de Markowitz". Considere uma matriz de correlação C entre dois ativos, e ρ a correlação entre os retornos.

$$C = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

O traço de C, $tr(C) = \Lambda_{1,1} + \Lambda_{1,2} = 2$, logo ρ define o tamanho de um autovalor em detrimento do outro. É fácil verificar que o determinante da matriz é dado por $|C| = \Lambda_{1,1} \cdot \Lambda_{1,2} = 1 - \rho^2$. Onde $|C|$ é máximo quando $\Lambda_{1,1} = \Lambda_{1,2} = 1$, que corresponde ao caso onde $\rho = 0$ e $|C|$ é mínimo quando $\Lambda_{1,1}$ ou $\Lambda_{2,2}$ são iguais a zero, que corresponde ao caso onde $|\rho| = 1$. O valor da inversa de C é dado por:

$$C^{-1} = W\Lambda^{-1}W' = \frac{1}{|C|}\begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix}$$

Implicando que quanto mais ρ se afasta de zero, maior um autovalor se torna em relação ao outro fazendo com que $|C|$ se aproxime de zero e tornando C instável. Lembrando que a matriz de correlação é apenas uma versão normalizada da matriz de covariância, utilizada nos algoritmos de otimização convexa.