

Universidade Federal do ABC
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas
Trabalho de Graduação em Engenharia de Informação

Estudo sobre Técnicas de Aprendizado de Máquina para predição do Índice Bovespa

Lucas Ortega Venzel

Santo André
Novembro de 2021

Lucas Ortega Venzel

Estudo sobre Técnicas de Aprendizado de Máquina para predição do Índice Bovespa

Trabalho de Graduação apresentado ao concluir a Graduação em Engenharia de Informação, como parte dos requisitos necessários para a obtenção do Título Bacharel em Engenharia de Informação.

Universidade Federal do ABC

Orientador: Murilo Bellezoni Loiola

Santo André

Novembro de 2021

Lucas Ortega Venzel

Estudo sobre Técnicas de Aprendizado de Máquina para predição do Índice Bovespa

Trabalho de Graduação apresentado ao concluir a Graduação em Engenharia de Informação, como parte dos requisitos necessários para a obtenção do Título Bacharel em Engenharia de Informação.

Santo André, 24 de Novembro de 2021:

Murilo Bellezoni Loiola
Orientador

Ricardo Suyama
Convidado 1

Vinícius Ormenesse
Convidado 2

Santo André
Novembro de 2021

Agradecimentos

Agradeço a todos que acreditaram que o conhecimento deveria ser compartilhado e que despenderam tempo de suas vidas por essa causa.

Agradeço especialmente aos seres vivos que me proporcionaram experiências e aprendizados até o presente momento.

*“Se você não trabalha em problemas importantes,
não é provável que você faça um trabalho importante.”
(Richard Hamming)*

Resumo

As últimas décadas tem presenciado um aumento expressivo na demanda do uso de algoritmos de aprendizado de máquina no mercado de capitais. Para entender se é possível utilizar aprendizado de máquina e indicadores de análise gráfica, este projeto tem como objetivo analisar o desempenho de quatro diferentes tipos de algoritmos e métricas relacionadas as suas saídas. Foram utilizados os modelos ARIMA, Gradient Boosted Regressor, Multilayer Perceptron e LSTM em combinação com oito indicadores técnicos para tentar realizar a predição de uma semana a frente do índice Bovespa. Os resultados apontaram que para as configurações de algoritmos, indicadores técnicos e intervalo de previsão, nenhum dos quatro diferentes tipos de modelo conseguiu obter um valor de coeficiente de determinação consistentemente acima do valor neutro zero. Também foi possível observar que nenhum dos algoritmos teve um retorno maior que o acumulado do índice para a simulação realizada.

Palavras-chaves: Aprendizado de Máquina. Séries Temporais. Índice Bovespa.

Abstract

The last few decades have witnessed a significant increase in the demand for the use of machine learning algorithms in the capital market. To understand if it is possible to use machine learning and graphical analysis indicators to predict the return of the Bovespa index, this project aims to analyze the performance of four different types of algorithms and metrics related to their outputs. The ARIMA, Gradient Boosted Regressor, Multilayer Perceptron and LSTM models were used in combination with eight technical indicators to try to perform the prediction of one week ahead of the Bovespa index. The results showed that for the algorithms, technical indicators and forecast range settings, none of the four different types of models managed to obtain a determination coefficient value consistently above the neutral zero value. It was also possible to observe that none of the algorithms had a return greater than the cumulative index for the simulation performed.

Keywords: Machine Learning. Time Series. Bovespa Index.

Lista de ilustrações

Figura 1 – Exemplo do indicador MACD para o ano de 2019	6
Figura 2 – Exemplo do indicador RSI para o ano de 2019	8
Figura 3 – Exemplo do indicador ADX para o ano de 2019	10
Figura 4 – Exemplo do indicador Aroon para o ano de 2019	11
Figura 5 – Exemplo do indicador CCI para o ano de 2019	12
Figura 6 – Exemplo do indicador DPO para o ano de 2019	13
Figura 7 – Exemplo do indicador MI para o ano de 2019	14
Figura 8 – Exemplificação funcionamento Gradient Boosted Trees.	15
Figura 9 – Exemplificação treinamento Gradient Boosted Trees.	15
Figura 10 – Perceptron com somatória das entradas e função ativação de degrau.	17
Figura 11 – Uma topologia de rede MLP completa.	18
Figura 12 – Exemplificação de funcionamento de uma célula com memória.	19
Figura 13 – Ilustração da estrutura de uma célula LSTM.	19
Figura 14 – Exemplo disposição dos dados obtidos via yfinance	20
Figura 15 – Exemplificação de valores alvo por amostra	27
Figura 16 – Exemplificação de atributos por amostra	28
Figura 17 – Exemplificação de alvo transformado por amostra	29
Figura 18 – Demonstração de espaço de procura de hiperparâmetros com o Optuna	32
Figura 19 – Topologia escolhida para o treinamento da MLP.	32
Figura 20 – Topologia escolhida para o treinamento da rede LSTM.	33

Lista de tabelas

Tabela 1 – Matriz de confusão	24
Tabela 2 – Resultados dos modelos por ano para a métrica Erro Absoluto Médio .	34
Tabela 3 – Resultados dos modelos por ano para a métrica Erro Quadrático Médio	34
Tabela 4 – Resultados dos modelos por ano para a métrica Coeficiente de Determinação	34
Tabela 5 – Resultados dos modelos por ano para a métrica Acurácia	34
Tabela 6 – Resultados dos modelos por ano para a métrica Precisão	35
Tabela 7 – Resultados dos modelos por ano para a métrica Revocação	35
Tabela 8 – Resultados dos modelos por ano para a métrica F1	35
Tabela 9 – Resultados dos modelos por ano	36

Lista de abreviaturas e siglas

IPO	Initial Public Offering (Oferta Pública Inicial)
ARIMA	Autoregressive Integrated Moving Average (Modelo Auto-Regressivo Integrado de Médias Móveis)
GBR	Gradient Boosted Regressor
MLP	Multilayer Perceptron (Perceptron Multicamadas)
LSTM	Long-Short Term Memory
VP	Verdadeiro Positivo
VN	Verdadeiro Negativo
FP	Falso Positivo
FN	Falso Negativo

Sumário

1	OBJETIVO	1
2	MOTIVAÇÃO DO ESTUDO	2
2.1	Mercado de Capitais e o Índice Bovespa	2
2.2	Modelagem de previsões financeiras	2
2.3	Aprendizado de Máquina no Mercado Financeiro	3
3	EMBASAMENTO TEÓRICO	5
3.1	Indicadores Técnicos	5
3.1.1	MACD	5
3.1.2	RSI	7
3.1.3	ADX	8
3.1.4	Aroon	10
3.1.5	CCI	11
3.1.6	DPO	12
3.1.7	MI	13
3.2	Algoritmos de Machine Learning Utilizados	14
3.2.1	Gradient Boosted Trees	14
3.2.2	ARIMA	16
3.2.3	MLP	17
3.2.4	LSTM	18
3.3	Implementação das Funções Utilizadas	20
3.3.1	Fonte e Formatação dos Dados	20
3.3.2	Atributos e Alvo	21
3.3.3	PMDARIMA	21
3.3.4	Scikit-Learn	22
3.3.5	Optuna	22
3.4	Métricas de Avaliação dos algoritmos	23
3.4.1	Coeficiente de Determinação	23
3.4.2	Erro Quadrático Médio	23
3.4.3	Erro Absoluto Médio	23
3.4.4	Acurácia	24
3.4.5	Precisão	25
3.4.6	Revocação	25
3.4.7	Métrica F1	25

4	METODOLOGIA DE TREINAMENTO	26
4.1	Atributos e Alvo	26
4.2	Treinamento e Saída dos Modelos	28
4.3	Experimento	29
5	RESULTADOS E DISCUSSÃO	31
5.1	Treinamento e Testes	31
5.1.1	ARIMA	31
5.1.2	GBR	31
5.1.3	MLP	32
5.1.4	LSTM	33
5.2	Métricas de Desempenho	33
5.2.1	Erro Absoluto Médio	34
5.2.2	Erro Quadrático Médio	34
5.2.3	Coefficiente de Determinação	34
5.2.4	Acurácia	34
5.2.5	Precisão	35
5.2.6	Revocação	35
5.2.7	Métrica F1	35
5.2.8	Comentários sobre os Resultados	35
5.3	Experimento: Tomada de decisão	36
6	CONCLUSÕES E TRABALHOS FUTUROS	37
6.1	Conclusões	37
6.2	Trabalhos Futuros	37
	REFERÊNCIAS	39

1 Objetivo

Os principais objetivos deste trabalho são analisar o desempenho de algoritmos de aprendizado de máquina aplicados ao problema de previsão de séries temporais, neste caso utilizando o índice Bovespa, e a capacidade de indicadores técnicos de serem bons atributos de treinamento para tais algoritmos.

A comparação da efetividade dos modelos treinadas serão feitas através de métricas de regressão e classificação amplamente usadas na análise de desempenho de modelos, também será realizado um experimento prático de tomada de decisão com base na estratégia de alocação semanal de recursos entre a compra índice analisado ou a alocação em moeda, no caso o real brasileiro.

2 Motivação do Estudo

2.1 Mercado de Capitais e o Índice Bovespa

A complexidade da economia global evoluiu rapidamente com a ascensão do mundo digital. Uma consequência desse fato é o aumento da importância do mercado financeiro para impulsionar também a expansão dos agentes dessa complexidade: as empresas e suas relações de cadeias produtivas e consumidoras.

Uma companhia que projeta seu crescimento, mas que demanda capital para que essa projeção se torne realidade é um agente deficitário no mercado e para ela existem diversas alternativas junto a agentes poupadores para captar esses recursos. O mercado de ações é uma dessas alternativas, frequentemente usado para expansão de empresas com um grande valor de mercado. Nesta etapa é que ocorre o mercado primário de ofertas, comumente chamado pela sua sigla em inglês *IPO (Initial Public Offering)* ou oferta pública inicial. Também nessa etapa que a empresa avalia o seu valor e o preço inicial de negociação do seu ativo. Uma companhia aberta, ou seja, que tenha parte ou o total de seu valor negociado no mercado só recebe investimentos no mercado primário.

O mercado secundário de ações é o que mantém a liquidez dos investimentos nas empresas abertas, ou seja, um investidor que tenha adquirido papéis de uma companhia pode se desfazer do seu investimento pelo preço que seja acordado com um comprador na bolsa de valores. Essa dinâmica de fechamento de negócios no mercado secundário é tomada pela percepção de valor do ativo para cada parte negociante e apesar de poder ser baseada em dados (análises de indicadores técnicos e fundamentalistas), depende da expectativa de retorno de ambas as partes.

O Índice Bovespa foi criado para ser uma referência do mercado brasileiro de ações, é composto pelas companhias mais negociadas na bolsa de valores e é atualizado periodicamente para refletir mudanças de volume de negociação dos papéis. Este índice é usado por grande parte do mercado para aferir o desempenho de estratégias de investimentos no mercado nacional, habitualmente chamado de Benchmark. Uma estratégia de investimento geral com base em ações brasileiras tem bom desempenho caso seu rendimento acumulado seja igual ou melhor que a variação do índice Bovespa no mesmo período.

2.2 Modelagem de previsões financeiras

Modelar previsões é uma tarefa requerida em diversos setores econômicos, como dito por Hyndman & Athanasopoulos, quaisquer que sejam os intervalos envolvidos, saber

modelar o futuro com base nos padrões de dados passados determina o sucesso de um planejamento [1].

O estudo massivo na modelagem de problemas financeiros começou de fato no século passado com artigos publicados no meio do século, muitos desses ainda tendo relevância nas análises atuais, como a seleção de portfólio de Markowitz [2] em 1952 e a teoria de Precificação de Ativos de Sharpe [3] em 1964, ambos os trabalhos visavam encontrar padrões nos dados passados de cada ativo. Poucos anos após estes trabalhos, Eugene F. Fama apresentou sua teoria do Mercado Eficiente, que até o momento era um estudo empírico [4] [5]; a teoria causou grande polêmica ao assumir que qualquer que seja o esforço de encontrar padrões e realizar uma gestão ativa em valores negociados no mercado financeiro, não pode tem desempenho consistentemente melhor do que estratégias de gestão passiva, pois toda informação útil disponível já foi absorvida e precificada pelos agentes negociantes.

Existem dois métodos analíticos básicos que podem compor uma gestão ativa de recursos, ambos consideram que a teoria do mercado eficiente não representa todos os possíveis cenários de negociação, ou seja, ativos podem estar sendo negociados por preços não justos pelo valor da empresa ou bem que o ativo representa. A análise fundamentalista utiliza os dados financeiros da companhia, dados econômicos e padrões do setor de atuação de cada um para encontrar um valor determinado justo para a capitalização total e, com isso, encontrar oportunidades de rendimento até que a negociação convirja para esse preço. A origem do outro método remete ao nascimento do conceito moderno de bolsa de valores, em Amsterdã no século XVII [6], diferentemente dos princípios fundamentalistas, a análise técnica observa apenas as informações obtidas na própria negociação dos ativos para encontrar padrões juntos que indiquem qual direção a negociação tomará [7].

2.3 Aprendizado de Máquina no Mercado Financeiro

Desde o fim do século XX um novo campo no mercado financeiro se destacou por seus resultados: a análise quantitativa. Assim como os métodos mais tradicionais, os investidores quantitativos, como são chamados, tentam encontrar padrões nos dados históricos para fazer previsões de movimentação do mercado, porém, o que os diferencia é a carga estatística trazida da academia.

Como relatado por Emerson, Kahn e outros em “The Future Of Investment Management” [8], o processo de digitalização e automatização das bolsas criou uma oportunidade para profissionais com forte base matemática estatística para criar estratégias de investimentos que trouxeram o rigor científico para o mercado financeiro.

Um dos casos mais célebres desse movimento do mercado foi o do matemático estadunidense James Harris Simons, fundador da Renaissance Technologies, uma das

primeiras e mais bem sucedidas gestoras quantitativas de investimentos. Como conta Gregory Zuckerman em *The Man Who Solved the Market* [9], uma biografia de Simons e sua companhia, o fluxo de jovens acadêmicos para o mercado financeiro fez com que o aprendizado de máquina, mesmo que ainda muito restrito devido ao desenvolvimento tecnológico da época, fosse aplicado massivamente nas estratégias de negociação nas bolsas mundiais.

Assim como o fluxo de acadêmicos para o mercado financeiro levou inovações para a área, essa proximidade fez com que a quantidade de artigos de “machine learning” publicados sobre negociação de ativos disparasse. Os principais objetos de estudos foram o desempenho de cada diferente algoritmo e quais dados melhor descreviam o problema. Dentre os algoritmos mais utilizados se destacam o Gradient Boosted Trees, Autoregressive Integrated Moving Average (ARIMA) e após o aumento expressivo de artigos relatando a eficácia de redes neurais para problemas não lineares, as Multi Layer Perceptrons ou MLP e as Recurrent Neural Networks, como o LSTM (Long-Short Term Memory) também se tornaram amplamente estudadas.

3 Embasamento Teórico

3.1 Indicadores Técnicos

A análise técnica é uma disciplina de precificação de ativos que visa encontrar padrões para oportunidades de compra ou venda com base nos dados de negociação, como o movimento do preço ou o volume de negociações do ativo. A análise gráfica, como também pode ser chamada, leva em consideração que o mercado de negociação de ativos tem padrões identificáveis e mesmo que as movimentações pareçam aleatórias esses padrões identificáveis são repetidos periodicamente[10].

Os indicadores técnicos são utilizados como base para a composição das estratégias que ditam a tomada de decisão, apesar de existirem diversas estratégias diferentes, alguns indicadores são utilizados em grande parte dessas composições, como é o caso do MACD.

3.1.1 MACD

Em inglês, Mobile Average Convergence-Divergence, utiliza a diferença de duas médias móveis exponenciais, costumeiramente do fechamento das observações dentro de um intervalo, com dois períodos de análise diferentes. A indicação da movimentação precisa de outros dois componentes: o sinal e o histograma. É classificado como um indicador de tendência de compra ou venda de um ativo.

O MACD pode ser calculado com diferentes configurações, uma vez que a escolha dos períodos das médias móveis é arbitrário, porém algumas escolhas são utilizadas com frequência. A fórmula pode ser descrita por:

$$\text{MACD} = \text{MME Curta} - \text{MME Longa} \quad (3.1)$$

Onde MME significa Média Móvel Exponencial e é definida por:

$$\text{MME} = \text{Preço Atual} * K + \text{MME}(\text{Período Anterior}) * (1 - K) \quad (3.2)$$

Para K:

$$K = \frac{2}{\text{Número de intervalos} - 1} \quad (3.3)$$

As médias móveis exponenciais são calculadas com base no preço. Considerando uma escolha comum para os intervalos de cada média, a equação fica:

$$\text{MACD} = \text{MME}(16 \text{ períodos}) - \text{MME}(26 \text{ períodos}) \quad (3.4)$$

O sinal também é uma componente da leitura final deste indicador e se utiliza de uma média móvel exponencial, o número de períodos considerados é menor que para a média móvel curta. Um valor usual é de 9 períodos, sendo assim:

$$\text{Sinal} = \text{MME} (9 \text{ períodos}) \quad (3.5)$$

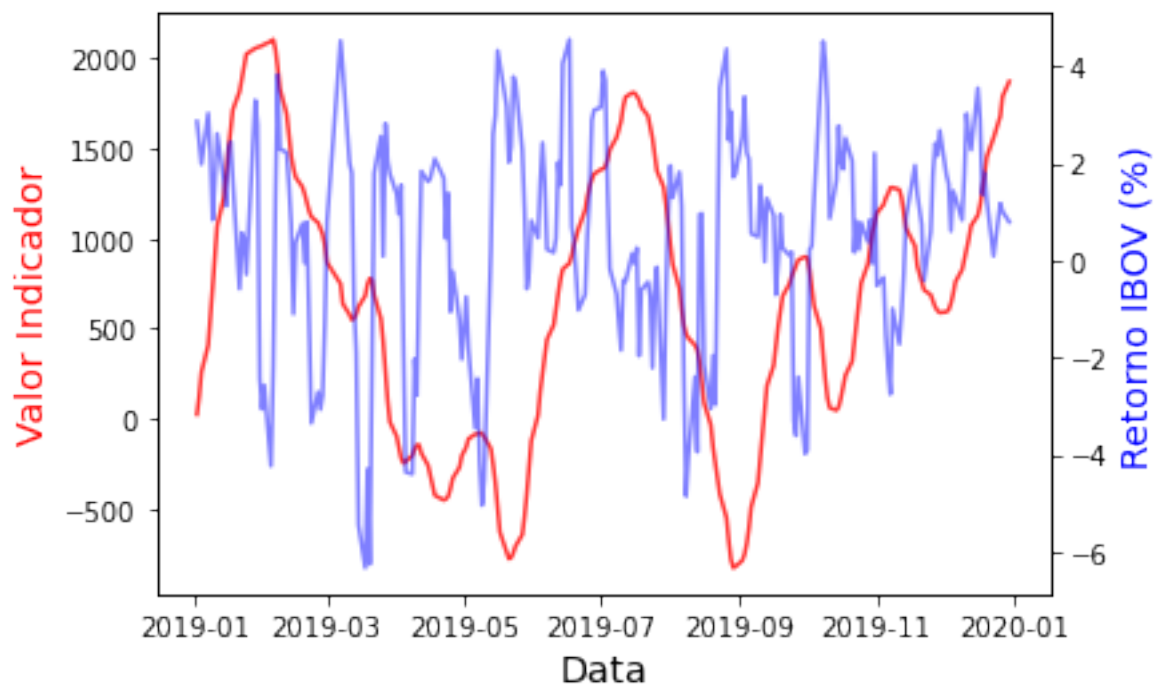
Por último, o histograma analisa a diferença entre as duas componentes anteriores e mostra a força do indicador:

$$\text{Histograma} = \text{MACD} - \text{Sinal} \quad (3.6)$$

A leitura da informação dada por este indicador é feita através do seu histograma: valores positivos indicam tendência de alta no ativo e valores negativos, de queda.

Na imagem abaixo é possível observar os valores do MACD para o índice Bovespa durante todo o ano de dois mil e dezenove, também é representado na imagem os valores de retorno do índice em valores percentuais. Lembrando que o valor do indicador é adimensional:

Figura 1 – Exemplo do indicador MACD para o ano de 2019



Fonte: Própria (2021)

3.1.2 RSI

Em inglês, Relative Strength Index, é um oscilador de momento criado por J. Welles Wilder e introduzido no seu livro “New Concepts in Technical Trading Systems”[11]. O RSI tenta indicar a informação de sobrecompra ou sobrevenda de algum ativo e o seu intervalo é definido entre 0 e 100. Este indicador pode ser calculado por:

$$RSI = 100 - 100/(1 + RS), \quad (3.7)$$

Onde RS é definido por:

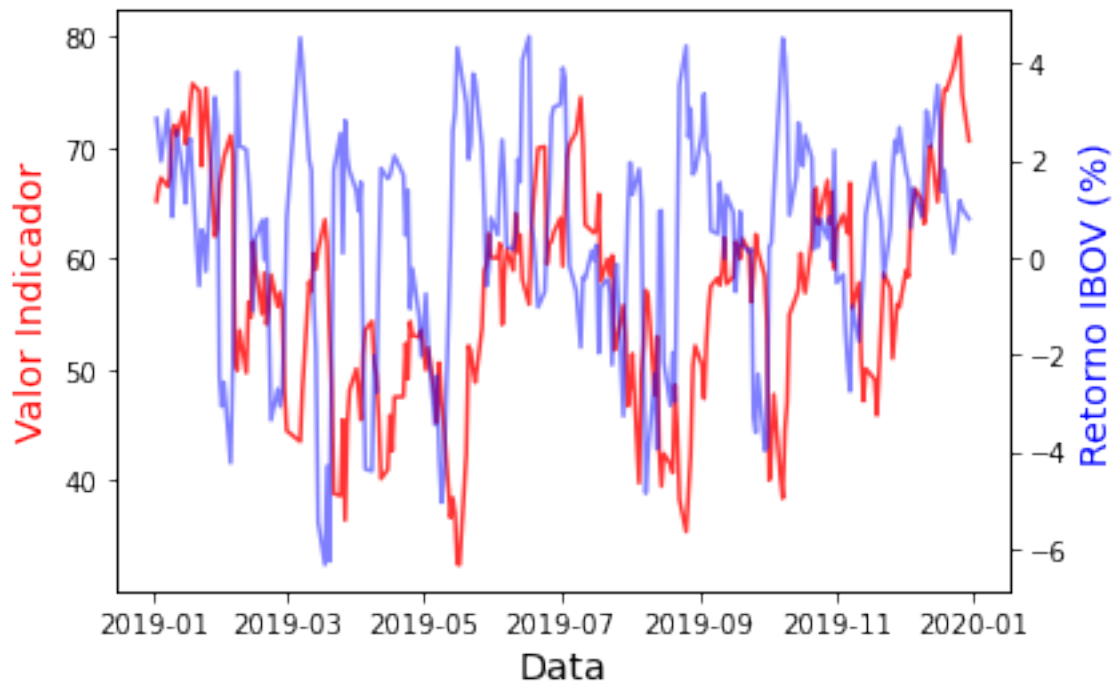
$$RS = \frac{\text{Soma de Altas no Período Observado}}{\text{Modulo da Soma de Quedas no Período Observado}} \quad (3.8)$$

Assim como no MACD, o período de observações é uma configuração arbitrária, porém é comum que sejam utilizados quatorze intervalos. É interessante notar que caso não haja quedas no período observado, o valor de RS não é definido. Neste caso, é considerado o valor zero para o RSI.

A leitura deste indicador pode ser feita através dos seus valores. É comum adotar que valores menores que 30 indicam que houve uma sobrevenda do ativo e provavelmente ocorrerá uma correção. O contrário é válido para valores maiores que 70 e, por último, o intervalo central que habita entre 30 e 70 indica estabilidade.

Na imagem abaixo é possível observar os valores do RSI para o índice Bovespa durante todo o ano de dois mil e dezenove, também é representado na imagem os valores de retorno do índice em valores percentuais. Lembrando que o valor do indicador é adimensional:

Figura 2 – Exemplo do indicador RSI para o ano de 2019



Fonte: Própria (2021)

3.1.3 ADX

Indicador de direção média, em inglês Average Directional Index, é utilizado para verificar a força de uma tendência nas variações de um ativo. Ao contrário de outros indicadores, ele é composto por duas componentes independentes que capturam uma das direções de variação: o indicador direcional negativo (-DI) e o indicador direcional positivo (+DI). O ADX pode ser obtido através de:

$$ADX = \frac{|DIP - DIN|}{DIP + DIN} \quad (3.9)$$

com os valores intermediários:

$$DIP = (DMP \text{ Suavizado} / ATR) * 100, \quad (3.10)$$

$$DIN = (DMN \text{ Suavizado} / ATR) * 100, \quad (3.11)$$

onde:

$$DMP = \text{Alta atual} - \text{Alta anterior}, \quad (3.12)$$

$$DMN = \text{Baixa atual} - \text{Baixa anterior}, \quad (3.13)$$

e por fim:

$$DM_{\text{Suavizado}} = \sum_{t=1}^{14} DM - \frac{\sum_{t=1}^{14} DM}{14} + DM_{\text{Atual}}, \quad (3.14)$$

A componente ATR é calculada tomando o média dos maiores valores em módulo entre máxima e mínima diária; máxima e fechamento anterior e mínima e fechamento anterior, para cada um dos quatorze intervalos observados.

O ADX é utilizado como um indicador de momento, ou seja, além de mostrar a tendência é útil também para entender a força dessa movimentação. Para realizar sua leitura é necessário apenas observar o valor e classificá-lo em algum desses intervalos:

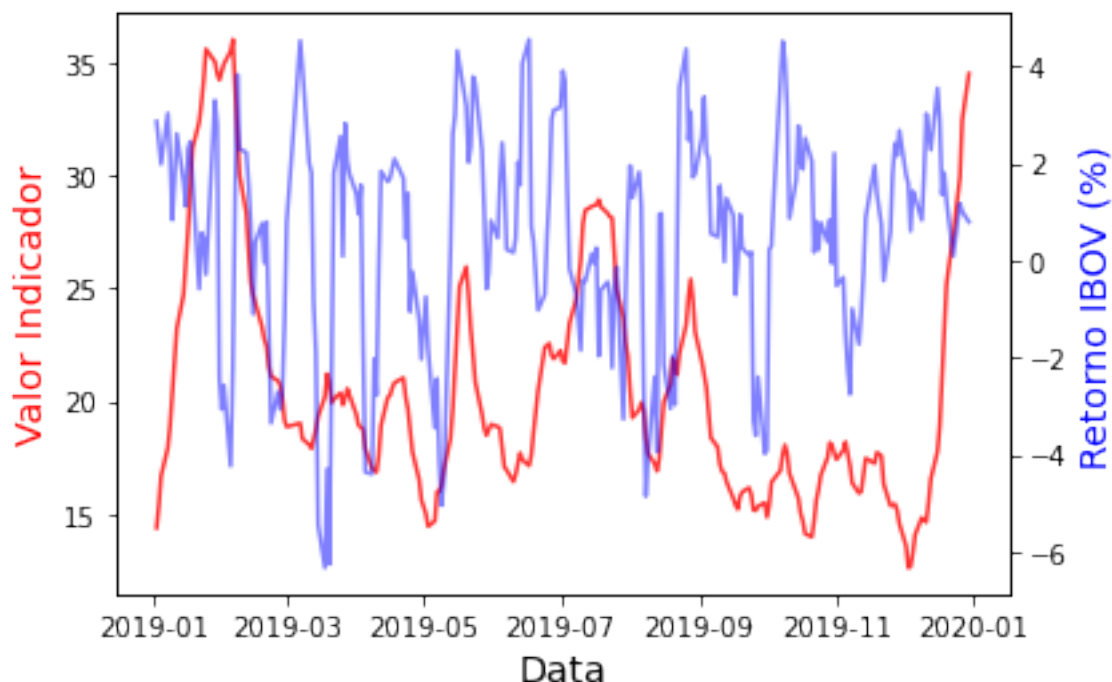
- $0 < \text{ADX} < 20$, é uma tendência fraca;
- $25 < \text{ADX}$ é uma tendência forte.

Além de formação de tendências é necessário entender o sinal de compra ou venda, e para isso é utilizado o cruzamento dos indicadores -DI e +DI. Caso -DI ultrapasse +DI, é um sinal de venda; caso +DI ultrapasse -DI, é um sinal de compra.

Com o sinal e a força da tendência é possível montar a estratégia para entrada e saída de um ativo.

Na imagem abaixo é possível observar os valores do ADX para o índice Bovespa durante todo o ano de dois mil e dezenove, também é representado na imagem os valores de retorno do índice em valores percentuais. Lembrando que o valor do indicador é adimensional:

Figura 3 – Exemplo do indicador ADX para o ano de 2019



Fonte: Própria (2021)

3.1.4 Aroon

O indicador Aroon é um indicador de tendência, usado para verificar a direção da tendência, assim como sua força. Este indicador parte do pressuposto que fortes tendências de altas são sucedidas por novas valorizações no preço de um ativo e fortes tendências de baixa são também sucedidas por desvalorização no preço do ativo.

Dois componentes integram a leitura deste indicador: o componente para tendências de valorização, Aroon up, e o componente para tendências de desvalorização, Aroon down. Eles são definidos por:

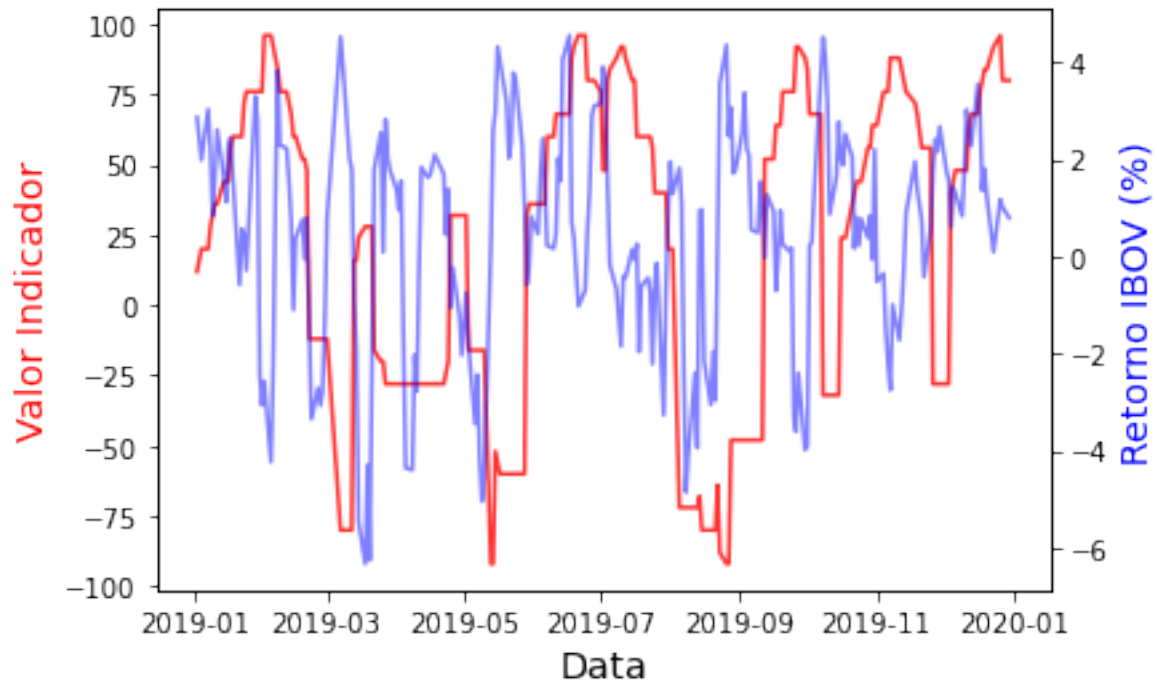
$$Aroonup = \frac{25 - \text{numero de observacoes desde a ultima valorizacao}}{25} * 100, \quad (3.15)$$

$$Aroondown = \frac{25 - \text{numero de observacoes desde a ultima desvalorizacao}}{25} * 100. \quad (3.16)$$

A leitura da tendência é feita pela diferença entre as componentes. Caso a componente de valorização seja maior que a de desvalorização, isso pode indicar uma tendência de alta no preço e caso a componente de desvalorização seja maior que a de valorização, uma tendência de baixa no preço.

Na imagem abaixo é possível observar os valores do Aroon para o índice Bovespa durante todo o ano de dois mil e dezenove, também é representado na imagem os valores de retorno do índice em valores percentuais. Lembrando que o valor do indicador é adimensional:

Figura 4 – Exemplo do indicador Aroon para o ano de 2019



Fonte: Própria (2021)

3.1.5 CCI

Do inglês, Commodity Channel Index, este indicador é utilizado para verificar momentum, semelhante ao conceito de inércia em física para a taxa de mudança de preço em um ativo. É definido por:

$$CCI = \frac{\text{preço típico} - MM}{0.015 * \text{variancia das observacoes}}; \quad (3.17)$$

onde:

$$\text{preço típico} = \frac{\sum(\text{maxima} + \text{minima} + \text{fechamento})}{3} \quad (3.18)$$

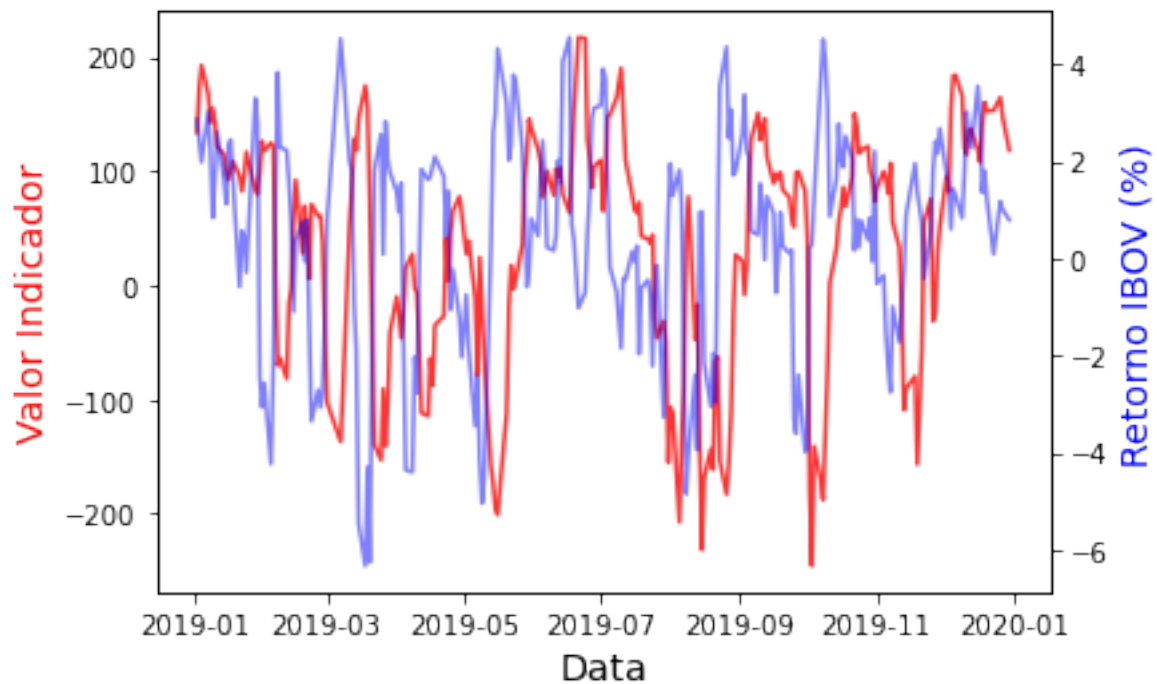
$$MM = \text{média móvel do número de períodos sendo analisados.} \quad (3.19)$$

O CCI pode ser interpretado pelo sinal do indicador e pela variação entre as observações. Uma mudança de um valor próximo ao nulo para um valor próximo à 100

indica um começo de tendência de compras, e a saída de um valor próximo ao nulo para um valor próximo a -100 indica um começo de tendência de vendas.

Na imagem abaixo é possível observar os valores do CCI para o índice Bovespa durante todo o ano de dois mil e dezenove, também é representado na imagem os valores de retorno do índice em valores percentuais. Lembrando que o valor do indicador é adimensional:

Figura 5 – Exemplo do indicador CCI para o ano de 2019



Fonte: Própria (2021)

3.1.6 DPO

Do inglês, Detrended Price Oscillator, é um indicador utilizado para verificar pontos históricos de saída e entrada na série de preço do ativo. Ele é definido por:

$$DPO = \text{preço de fechamento de } t^{-\frac{X+2}{2}} - MM(X), \quad (3.20)$$

onde:

$$X = \text{número de observações no período analisado}, \quad (3.21)$$

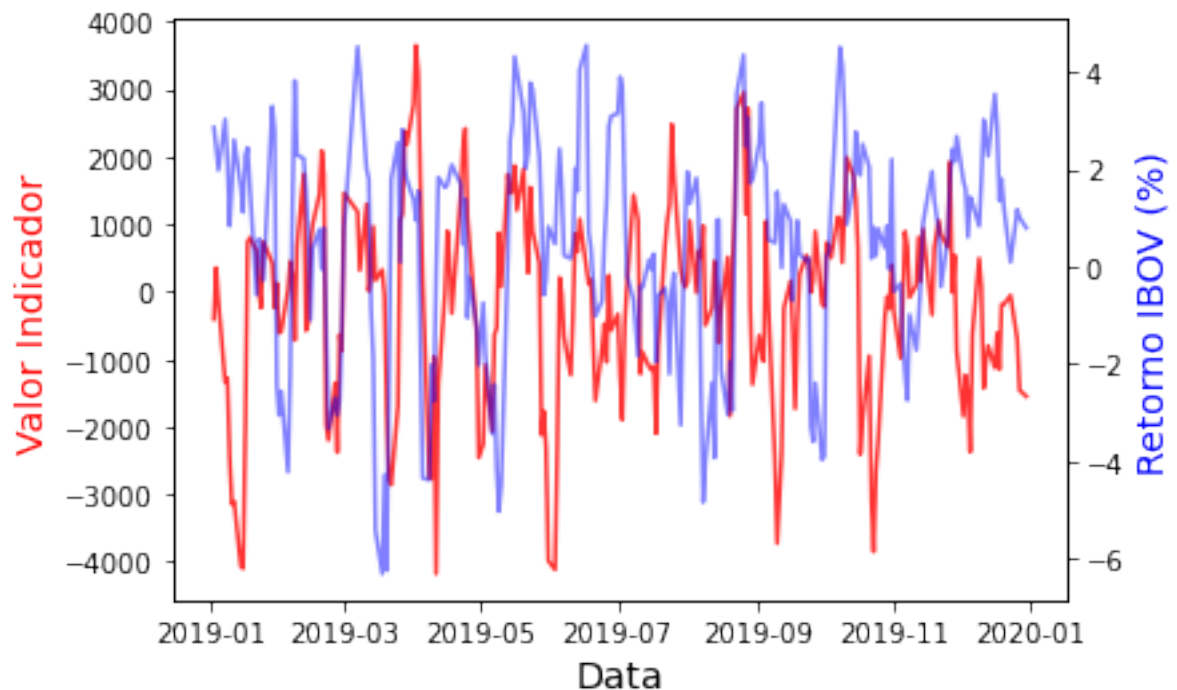
$$t^{-y} = y \text{ dias anteriores ao período atual}, \quad (3.22)$$

$$MM(X) = \text{média móvel dos últimos } X \text{ períodos}. \quad (3.23)$$

O DPO tenta identificar ciclos na variação dos preços, sendo assim, um valor negativo de DPO indica um ciclo de desvalorização do ativo e um valor positivo um ciclo de valorização, sempre considerando a média móvel das últimas X observações.

Na imagem abaixo é possível observar os valores do DPO para o índice Bovespa durante todo o ano de dois mil e dezenove, também é representado na imagem os valores de retorno do índice em valores percentuais. Lembrando que o valor do indicador é adimensional:

Figura 6 – Exemplo do indicador DPO para o ano de 2019



Fonte: Própria (2021)

3.1.7 MI

Do inglês, Mass Index Indicator, é um indicador de tendência que analisa o intervalo entre a máxima e a mínima nos preços de um ativo durante um determinado período. Ele é determinado por:

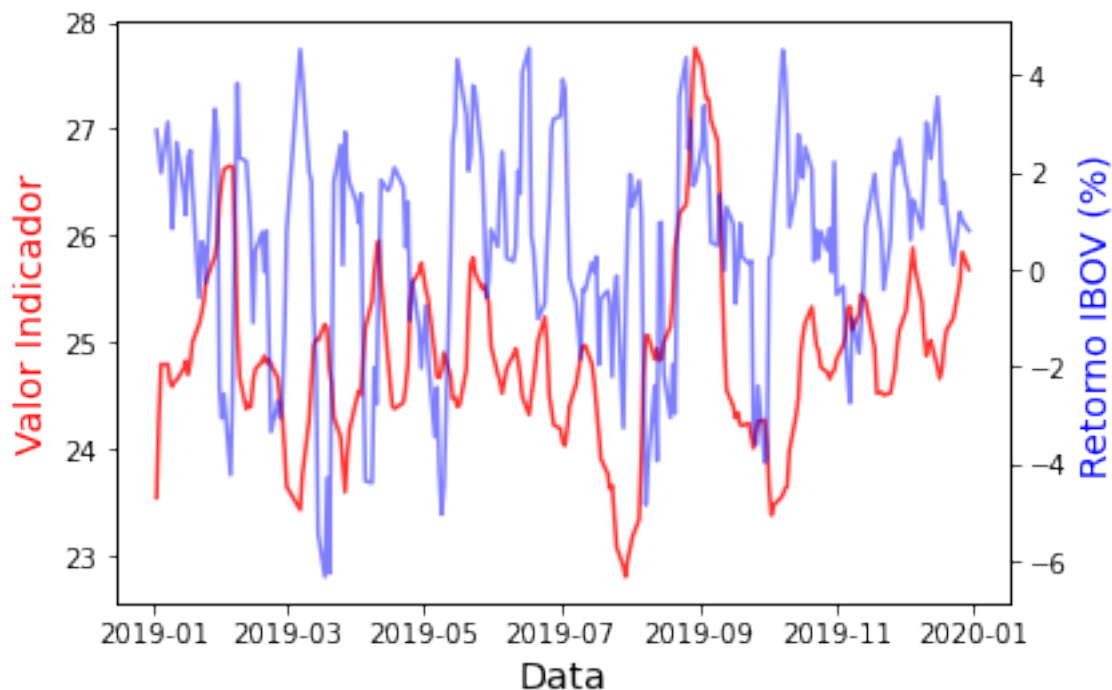
$$MI = \sum \frac{MME9(\text{maxima} - \text{minima})}{MME9(MME9(\text{maxima} - \text{minima}))} \quad (3.24)$$

Onde, MME9 é a média móvel exponencial dos últimos nove períodos de observação.

O valor do MI de um ativo indica um conceito semelhante ao conceito de inércia na física, porém no caso deste indicador quanto maior o valor de MI, maior será a volatilidade do ativo.

Na imagem abaixo é possível observar os valores do MI para o índice Bovespa durante todo o ano de dois mil e dezanove, também é representado na imagem os valores de retorno do índice em valores percentuais. Lembrando que o valor do indicador é adimensional:

Figura 7 – Exemplo do indicador MI para o ano de 2019



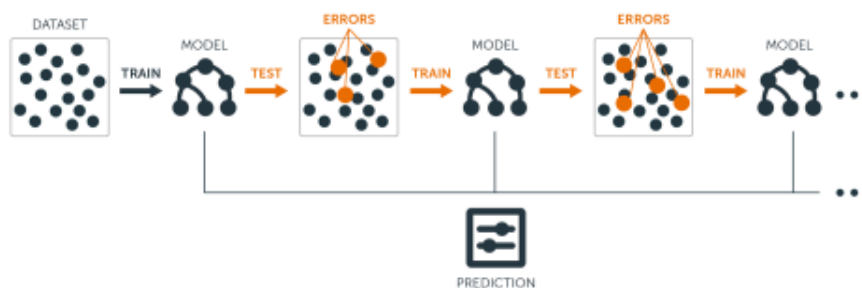
Fonte: Própria (2021)

3.2 Algoritmos de Machine Learning Utilizados

3.2.1 Gradient Boosted Trees

As GBT são uma família de algoritmos supervisionados baseados em árvores de decisão e que podem ser usados para problemas de classificação e também para regressão. O Gradient Boosting é uma técnica de conjunto, em inglês “ensemble”, assim como as florestas aleatórias, porém ao contrário dessas as GBTs não criam diversas árvores independentes, mas sim um conjunto de árvores em que cada modelo subsequente é treinado com base nos resíduos do anterior.

Figura 8 – Exemplificação funcionamento Gradient Boosted Trees.



Fonte: [11]

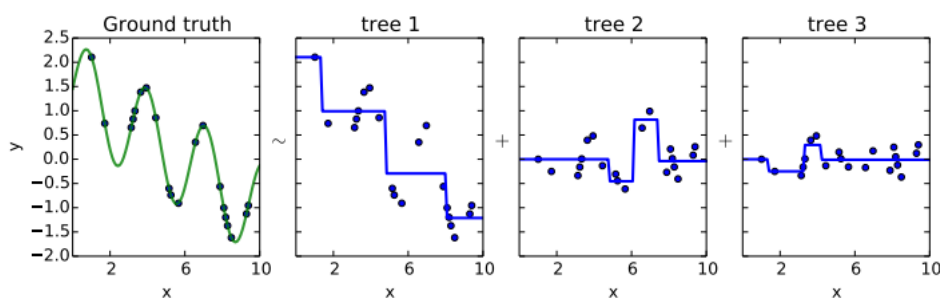
As vantagens das GBTs são:

- Capacidade de manipular dados heterogêneos: atributos em diferentes escalas e atributos categóricos;
- Podem ser utilizadas com base em diferentes funções custo;
- Ao contrário de árvores que utilizam apenas os atributos originais, conseguem capturar a interação de atributos em árvores que são alimentadas com os resíduos.

O algoritmo básico para o treinamento de uma GBT é dado por:

- Treinar uma única árvore de decisão no conjunto de atributos, $f_1(x)$, na variável de predição (y);
- Treinar uma próxima árvore com os resíduos da árvore anterior: $h_2(x) = y - f_1(x)$;
- Adicionar a árvore anterior no conjunto de predição: $f_2(x) = h_2(x) + f_1(x)$;
- Repetir os passos anteriores até que algum dos critérios de parada estabelecidos na declaração do modelo seja obtido.

Figura 9 – Exemplificação treinamento Gradient Boosted Trees.



Fonte: [11]

3.2.2 ARIMA

Modelo Auto-Regressivo Integrado de Médias Móveis, ou em inglês, Autoregressive Integrated Moving Average. Este modelo é composto por três diferentes estágios que são capazes de trabalhar diretamente com dados de uma série temporal não estacionária.

As duas primeiras letras de ARIMA, AR, significam auto-regressivo, ou seja, pontos anteriores da série temporal podem ser usados para prever pontos futuros. Nesta etapa é feita uma auto-regressão em que cada ponto anterior recebe um peso para a previsão. Um modelo auto-regressivo pode ser descrito por:

$$y_t = c + \phi^1 y_{t-1} + \phi^2 y_{t-2} + \dots + \phi^p y_{t-p} + \epsilon_t; \quad (3.25)$$

onde:

$$y_t = \text{previsão}, \quad (3.26)$$

$$c = \text{constante}, \quad (3.27)$$

$$\phi = \text{peso atribuído para aquele instante } t^{-x}, \quad (3.28)$$

$$y_{t-x} = \text{observação passada ao instante } p \text{ em } x \text{ intervalos}, \quad (3.29)$$

$$\epsilon = \text{ruído branco padrão com média nula e variância unitária}. \quad (3.30)$$

A segunda combinação é MA, para “moving average” ou médias móveis, responsável por considerar as janelas de tempo anteriores consideradas para a previsão e sua suavização para eliminar ruídos de apenas observações unitárias. Ao contrário de uma auto-regressão, médias móveis levam em consideração os erros passados para calcular seus pontos futuros:

$$y_t = c + \epsilon_t + \phi^1 \epsilon_{t-1} + \phi^2 \epsilon_{t-2} + \dots + \phi^q \epsilon_{t-q} \quad (3.31)$$

Onde as definições de símbolos permanecem as mesmas da equação 4.24.

A terceira modelagem é a terceira letra, I, significa integrado, ou seja, o inverso de derivado. Existe uma tendência na série temporal observada, essa etapa é responsável por reduzir a sazonalidade dos dados analisados. Como as observações anteriores podem ter sido derivadas para transformar os dados observados em estacionários, a combinação de AR e MA para um previsão genérica é dada por:

$$\mathbf{y}'t = c + \phi^1 \mathbf{y}'t^{-1} + \dots + \phi^p \mathbf{y}'t^{-p} + \dots + \theta^1 \varepsilon t^{-1} + \dots + \theta^q \varepsilon t^{-q} + \varepsilon t \quad (3.32)$$

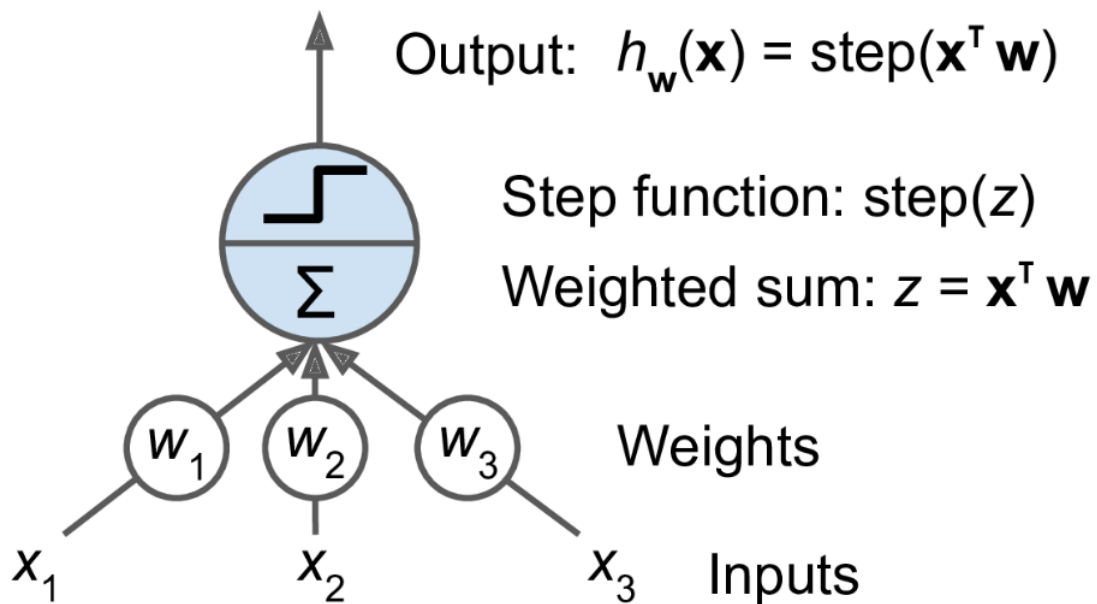
Onde todas as observações \mathbf{y}' estão diferenciadas, podendo ser diferenciações de ordem maior que unitária, a modelagem I reverte o processo de diferenciação anterior para que a previsão mostre os valores futuros absolutos e não suas variações.

3.2.3 MLP

Um Perceptron de Multicamadas, em inglês Multilayer Perceptron, é uma rede neural generalista que utiliza o processo de interconexão dos modelos anteriores de perceptrons unitários descritos basicamente por uma regressão logística, podendo ter diversos outros modos de ativação.

O Perceptron é a base de um modelo generalista de rede neural, é muito semelhante a uma regressão logística, mas em uma MLP pode assumir diversas outras formas de operação, tanto na entrada quanto na saída. No exemplo abaixo retirado do Hands-On Machine Learning with Scikit-Learn and Tensor Flow [11], temos um modelo de perceptron em que as entradas são somadas com seus devidos pesos e sua ativação é binária:

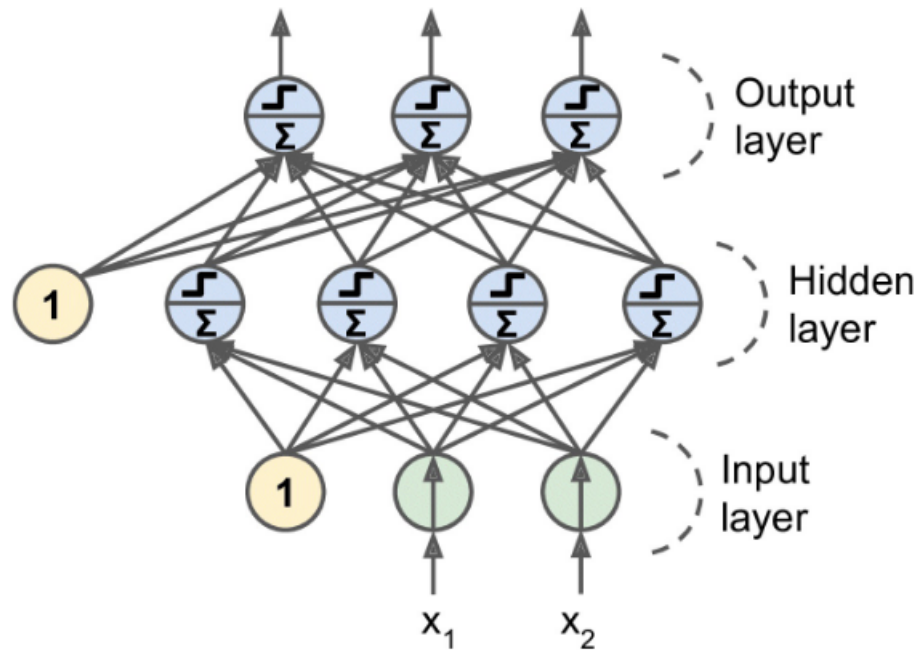
Figura 10 – Perceptron com somatória das entradas e função ativação de degrau.



Fonte: [11]

A rede completa é feita com base em múltiplos perceptrons conectados em mais de uma camada, ou seja, a saída da primeira camada de perceptrons deverá ser a entrada de no mínimo uma outra camada de perceptrons, comumente também chamado de neurônios. Essa topologia pode ser descrita com a imagem abaixo, retirada da mesma fonte da imagem anterior:

Figura 11 – Uma topologia de rede MLP completa.



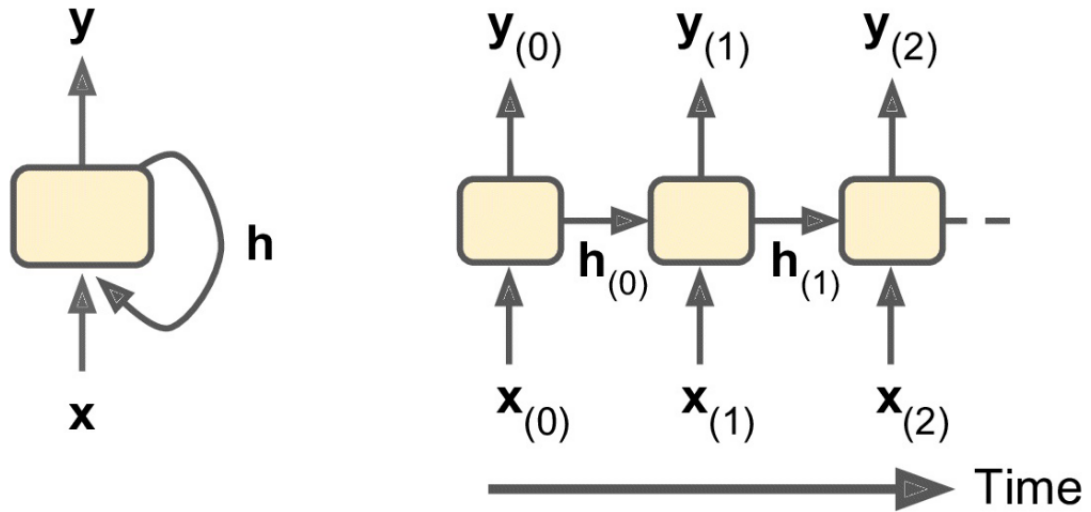
Fonte: [11]

No exemplo acima, temos duas entradas que seriam os dados de treinamento, X_1 e X_2 , que são passadas como parâmetros para a primeira camada de perceptrons; esta primeira camada é treinada e suas saídas servem de alimentação para a camada de saída, que repete o treinamento e que dessa vez realmente sai com o resultado da previsão do modelo, neste caso, cada neurônio de saída tem um significado, podendo ser a probabilidade de uma classe ou uma regressão. As células amarelas são constantes, adicionadas às entradas de todas as camadas, seu valor é unitário, mas seu peso varia para cada neurônio que recebe seu sinal.

3.2.4 LSTM

Em inglês, Long Short Term Memory, é uma rede neural recorrente, uma topologia específica das redes neurais tradicionais. Em uma rede neural recorrente, as saídas de uma camada podem ser adicionadas como entradas a camadas anteriores e essa configuração se assemelha muito a sistemas de controle e blocos de memória digitais. A topologia pode ser exemplificada por:

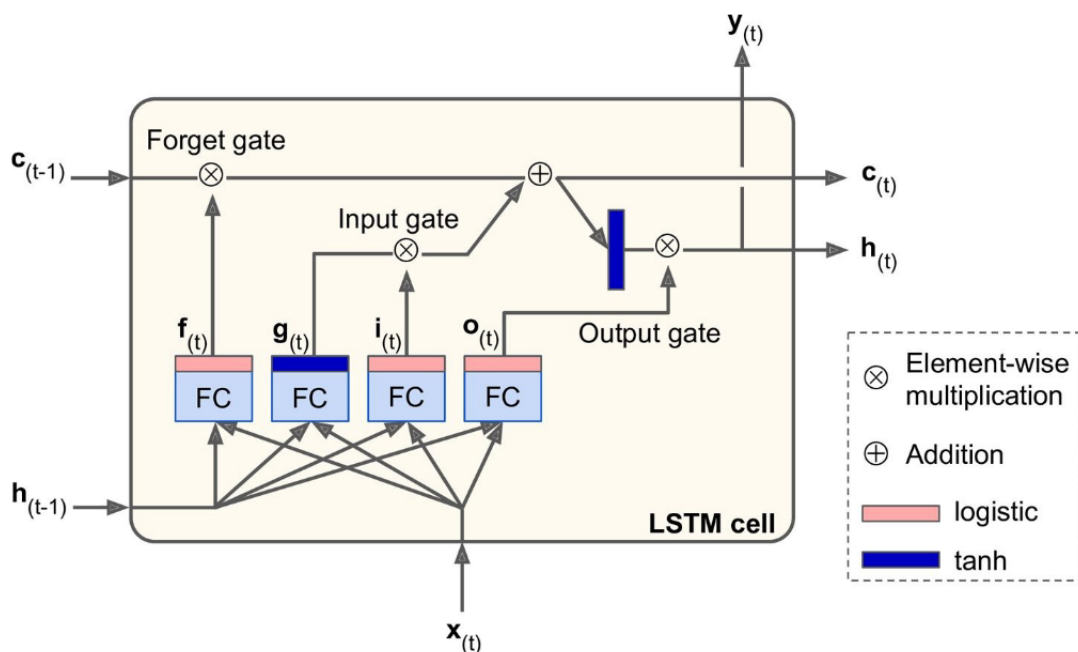
Figura 12 – Exemplificação de funcionamento de uma célula com memória.



Fonte: [11]

O LSTM em específico foi proposto antes dos anos 2000 por Sepp Hochreiter e Jürgen Schmidhuber e após anos de melhorias se tornou um dos maiores expoentes de redes recorrentes. Apesar de muitas vezes ser utilizado como um modelo caixa preta (“black box”), em que apenas são manipuladas as entradas e saídas e descartados os detalhes do seu funcionamento, uma célula LSTM pode ser estudada para entendimento do seus mecanismos entendendo alguns pontos chave:

Figura 13 – Ilustração da estrutura de uma célula LSTM.



Fonte: [11]

A célula do LSTM é composta por quatro camadas, uma delas, $g(t)$, é a principal e se comporta como um neurônio padrão, as outras três camadas $f(t)$, $i(t)$ e $o(t)$ são controladores de fluxo, cada um definindo se um sinal passará em determinado ponto da célula:

- $F(t)$ - controla o forget gate, que determina se uma memória de longo prazo deve ser apagada;
- $I(t)$ - controla o input gate, que determina quais componentes de $g(t)$ serão adicionadas a memória;
- $O(t)$ - controla o output gate, que determina quais componentes da saída devem ser contabilizados em $y(t)$ neste momento.

Esses controladores de fluxo de funcionamento fazem com que durante o treinamento o LSTM seja capaz de observar quando dados de entrada e de memória devem ser utilizados e quando podem ser descartados.

3.3 Implementação das Funções Utilizadas

3.3.1 Fonte e Formatação dos Dados

Os dados utilizados para o experimento foram obtidos da fonte Yahoo Finance [12], um tradicional meio de extração de notícias e informações sobre finanças no geral. A forma de extração desses dados em si foi feita por meio da biblioteca para a linguagem de programação python chamada yfinance [13], que implementa uma interface simples para obter os dados que ficam expostos na plataforma do Yahoo.

Figura 14 – Exemplo disposição dos dados obtidos via yfinance

	Open	High	Low	Close	Volume
Date					
2020-12-22	115825.0	116903.0	115648.0	116348.0	6947900
2020-12-23	116636.0	118311.0	116636.0	117857.0	6483300
2020-12-28	117806.0	119213.0	117805.0	119051.0	7171700
2020-12-29	119130.0	119861.0	118750.0	119475.0	6769700
2020-12-30	119410.0	120150.0	118919.0	119306.0	8235700

Fonte: Própria (2021)

Com a interface para o Yahoo Finance é possível obter os dados de negociação de milhares de ativos nacionais, internacionais, além de índices, como é o caso dos dados

extraídos para este estudo. Os dados de negociação são obtidos em formato de tabela em que as linhas correspondem as datas de vigência dos valores e as colunas os atributos, nesse caso: Open (preço de abertura), High (preço máximo do intervalo), Low (preço mínimo do intervalo), Close (preço de fechamento do intervalo) e Volume (volume em reais negociados).

3.3.2 Atributos e Alvo

O alvo para treinamento dos algoritmos nesse experimento será o retorno futuro de uma semana do índice Bovespa, tal alvo pode ser obtido facilmente apenas dividindo a observação futura pela observação da data em que se quer partir a previsão. Utilizar o retorno ao invés do preço em si é uma estratégia muito razoável, pois, além de facilitar o entendimento da previsão (valores maiores do que um significam retornos positivos e menores do que um significam retornos negativos) também tornam a série estacionária, ou seja, os valores do retorno tendem a se distribuir normalmente em torno de um valor fixo, no caso específico a série é estacionária e oscila ao redor do número um.

Os atributos utilizados para o treinamento dos algoritmos serão os indicadores técnicos descritos na seção anterior, ou seja, para cada data observada será calculado o valor dos indicadores listados. Para gerar estes indicadores nos dados coletados será utilizada uma implementação para python de funções de análise técnica, com estas funções é possível gerar e personalizar individualmente cada opção de intervalo, valor ou método dos indicadores baseados em análise gráfica. A implementação chama-se *TA (Technical Analysis)* [13], que também foi feita para ser acoplada nos dados extraídos pelo yfinance.

Para o treinamento com os atributos também é necessário atentar-se ao fato de que alguns dos algoritmos citados não trabalham bem com séries temporais, por isso foi feita a escolha de indicadores que operam em faixas de valores definidas, ao contrário por exemplo de médias móveis ou móveis exponenciais puras que tendem a seguir o preço do ativo que é observado. Dado isso não será necessário nenhuma modificação nesses atributos para o funcionamento adequado.

Para lidar com a natureza de alguns algoritmos escolhidos, principalmente das redes neurais (MLP e LSTM), todos os atributos foram escalados pela sua média e desvio padrão, para que dessa forma oscilem na mesma ordem de grandeza. Para escalar os atributos foi utilizado o conjunto de ferramentas para modelagem em python chamado scikit-learn [14], o pacote será mais detalhado nos tópicos seguintes.

3.3.3 PMDARIMA

O treinamento de modelos ARIMA demanda a escolha de seus parâmetros manualmente, como descrito na descrição desse algoritmo, é necessário determinar as componentes

(p, q, d) responsáveis pelas letras em seu nome. Cada um destes passos pode ser realizados com testes de hipóteses de diferentes metodologias, por exemplo: para descobrir se é necessário diferenciar a série temporal e subsequentemente o seu componente 'd' pode-se aplicar o teste 'ADF', sigla para o inglês Augmented Dickey Fuller Test, esse teste retorna a significância da não estacionariedade da série. Este é apenas um dos possíveis testes para uma das componentes do algoritmo.

A escolha por utilizar a implementação AutoARIMA do pacote PMDARIMA[15] se dá pelo fato da automação deste processo, tanto da variável alvo quanto das variáveis exôgenas. Com o PMDARIMA é possível declarar os testes que deseja implementar para calcular as componentes, os valores mínimos e máximos para serem testados e ao final o modelo parametrizado é treinado nos dados disponibilizados.

O pacote que contém a função utilizada traz diversas opções também para análise de séries temporais, como algumas funções para pré-processamento de dados e formatação de processos consecutivos para realização do treinamento.

3.3.4 Scikit-Learn

O Scikit-Learn[14] é um dos pacotes mais famosos para modelagem de dados em python. Ele conta com inúmeras classes e funções úteis na preparação, treinamento e avaliação de modelos. Este pacote foi utilizado diversas vezes durante o projeto para preparar os dados com os escaladores, para treinar os modelos de Gradient Boosted Trees e para avaliar as previsões de cada modelo.

3.3.5 Optuna

O Optuna[16] é um conjunto de ferramentas de otimização automática de hiperparâmetros que foi construído para otimizar algoritmos com base em um espaço de possibilidades previamente definidos pelo usuário. A ferramenta constrói estudos tentando otimizar a métrica especificada, onde cada configuração testada é uma tentativa.

Esse pacote é muito útil para testar as combinações que tendem ao ponto ótimo em um espaço de hiperparâmetros muito grande, em que a busca aleatória tomaria demasiado tempo e poder de processamento.

Dentro do pacote é possível escolher diversos métodos de otimização presentes nas opções, como uma procura aleatória, uma busca extensiva entre todas as possibilidades ou o padrão que é o estimador baseado em árvores.

O Optuna foi utilizado para escolher os melhores hiperparâmetros do modelo GBR, uma vez que esse modelo tem um espaço de possibilidades realmente grande.

3.4 Métricas de Avaliação dos algoritmos

3.4.1 Coeficiente de Determinação

O R^2 ou coeficiente de determinação é uma métrica para regressões muito interessante, uma vez que normaliza o erro das previsões para a variação média das observações que estão sendo testadas. Sua fórmula é:

$$R^2 = 1 - \frac{\sum (\text{valor previsto} - \text{valor observado})^2}{\sum (\text{valor observado} - \text{valor medio das observacoes})^2} \quad (3.33)$$

Uma regressão perfeita tem um R^2 igual 1. O valor diminui quanto maiores forem os resíduos das previsões, sendo que um valor negativo faz previsões piores que considerar a média das observações.

3.4.2 Erro Quadrático Médio

O erro quadrático médio é uma métrica simples para regressões. Ela leva em consideração uma penalização quadrática para o resíduo das previsões, semelhante ao modo em que é calculado o coeficiente de determinação. Um ponto negativo para esta métrica é o fato de o erro quadrático médio não ter uma escala padrão de avaliação entre um bom modelo e um ruim, ao contrário do R^2 .

O erro quadrático médio pode ser obtido por:

$$EQM = \frac{\sum (\text{valor previsto} - \text{valor observado})^2}{\text{numero de observacoes}} \quad (3.34)$$

3.4.3 Erro Absoluto Médio

O erro absoluto médio é uma métrica relativamente simples para regressões,. Ela penaliza linearmente um resíduo na previsão, ao contrário do coeficiente de determinação e do erro quadrático médio que penalizam de forma quadrática. A fórmula para o erro absoluto médio é:

$$EAM = \frac{\sum |(\text{valor previsto} - \text{valor observado})|}{\text{numero de observacoes}} \quad (3.35)$$

Por conter uma penalização de ordem diferente do coeficiente de determinação, essa métrica pode conter uma informação extra sobre os erros das previsões no geral, se há pontos sendo massivamente penalizados ou se o conjunto como um todo não está próximo do valor real observado.

3.4.4 Acurácia

Diferentemente das métricas apresentadas anteriormente a Acurácia é uma métrica utilizada para classificações. A saída final dos modelos treinados serão contínuas, ou seja, os modelos testados serão regressões, porém a escolha é binária, comprar ou vender o índice Bovespa em determinada semana, desta forma as saídas dos modelos podem ser binarizadas. Caso o modelo preveja uma alta no preço para a próxima semana será considerada a classe 1, caso contrário será considerado a classe 0.

Desta forma é possível calcular a acurácia dos modelos em prever a tendência do índice uma semana à frente. Uma forma de dividir previsões feitas por classificações é analisar sua matrix de confusão, uma tabela que analisa os eventos previstos com os eventos que realmente ocorreram. Neste caso, é possível criar a matriz de confusão exemplificada na Tabela 1 para analisar os resultados:

Tabela 1 – Matriz de confusão

Previsto/Real	Positivo	Negativo
Positivo	VP	FP
Negativo	FN	VN

Onde:

- VP (Verdadeiro Positivo) são amostras previstas positivas e que realmente são positivas;
- FP (Falso Positivo) são amostras previstas positivas mas que realmente são negativas;
- VN (Verdadeiro Negativo) são amostras previstas negativas e que realmente são negativas;
- FN (Falso Negativo) são amostras previstas negativas mas que realmente são positivas.

Para o cálculo da acurácia apenas tomamos a taxa de acertos nas previsões dividido pelo total de previsões, sendo assim a acurácia pode ser definida por:

$$acuracia = \frac{VP + VN}{VP + VN + FP + FN} \quad (3.36)$$

A acurácia é uma métrica superficial para indicar apenas a porcentagem de acertos de um modelo, sendo que apenas essa métrica não pode ser analisada de maneira única.

3.4.5 Precisão

A precisão, de maneira parecida com a acurácia, utiliza elementos da matriz de confusão para o seu cálculo, porém, a ideia da métrica discutida neste item é analisar a porcentagem de acertos dentre os eventos em quem o modelo previu a classe positiva:

$$precisao = \frac{VP}{VP + FP} \quad (3.37)$$

A precisão pode ser uma métrica muito importante para casos em que há um custo considerável para uma previsão falso positivo, porém desconsidera as previsões negativas.

3.4.6 Revocação

A revocação é de certa forma semelhante à métrica anterior, porém a revocação considera os casos realmente verdadeiros, ao contrário da precisão, que considera os casos previstos verdadeiros. Desta forma, é uma métrica que pode ser muito utilizada para casos em que há um custo para uma previsão falso negativa.

A revocação é definida por:

$$revocacao = \frac{VP}{VP + FN} \quad (3.38)$$

3.4.7 Métrica F1

Em ambas as duas métricas anteriores há uma noção de custo por previsões falso negativas ou falso positivas, porém os casos em que apenas umas destas previsões erradas é custoso não são os mais comuns. A Métrica F1 é uma mescla da precisão e revocação, ela considera o produto de ambas para uma métrica final que varia de zero até um. Ela pode ser obida por:

$$F1 = 2 * \frac{precisao * revocacao}{precisao + revocacao} \quad (3.39)$$

É uma métrica interessante, pois combina a importância da precisão e revocação. A métrica terá um valor ótimo quando a precisão e a revocação estiverem balanceadas. Para os casos em que algum dos componentes tender à zero, o resultado da métrica também tenderá ao nulo, penalizando fortemente esses desbalanços.

4 Metodologia de Treinamento

Neste capítulo serão descritos os procedimentos que foram realizados durante o treinamento dos algoritmos descritos no capítulo anterior, assim como foram geradas as métricas e o experimento de tomada de decisão analisados na seção de resultados. O código utilizado neste trabalho está disponível em um repositório que contém cada um dos passos necessários para reproduzir estes resultados [17]

4.1 Atributos e Alvo

Os modelos utilizados trabalharão como regressões, ou seja, as suas saídas serão valores analógicos que representarão a previsão do algoritmo para o retorno do índice na próxima semana. Considerando y como um vetor dos valores alvo, y pode ser definido por:

$$y = \frac{fechamento^t}{fechamento^{t+1}} - 1, \quad (4.1)$$

Onde:

$$t = \text{amostra na qual terá o valor } y \text{ calculado}, \quad (4.2)$$

$$t + 1 = \text{amostra uma semana posterior à amostra } t. \quad (4.3)$$

Desta forma o valor de cada amostra representará o acréscimo de valor ao índice uma semana à frente. Neste caso, valores maiores do que zero representam valorização e valores menores do que zero representam desvalorização no período. Uma amostra dos valores alvo para o treinamento podem ser observadas na imagem abaixo, onde a coluna *Date* se refere a data da amostra no formato ano - mês - dia e a segunda coluna representa o retorno calculado como na expressão descrita acima:

Figura 15 – Exemplificação de valores alvo por amostra

Date	
2019-01-02	0.028579
2019-01-03	0.024486
2019-01-04	0.019784
2019-01-07	0.030262
2019-01-08	0.021992
2019-01-09	0.008332
2019-01-10	0.016470
2019-01-11	0.026042
2019-01-14	0.016258
2019-01-15	0.011132
2019-01-16	0.022936
2019-01-17	0.024394
2019-01-18	0.016442
2019-01-21	-0.005895
2019-01-22	0.005636
2019-01-23	0.004536
2019-01-24	-0.002897
2019-01-28	0.032951
2019-01-29	0.027938
2019-01-30	-0.024331

Fonte: Própria (2021)

Tendo os rótulos de alvo para o treinamento, os atributos considerados nos modelos, são os valores de cada um dos oito indicadores técnicos descritos no capítulo de embasamento deste trabalho. Para cada amostra, que também será uma data diferente de pregão da bolsa nacional, o valor do índice será calculado com base nos dados passados necessários. Considerando os valores calculados, cada amostra contará com oito atributos para o treinamento. Abaixo podem ser observados os atributos para as mesmas amostras exemplificadas na figura 15:

Figura 16 – Exemplificação de atributos por amostra

Date	MACD	RSI	ADX	Aroon	CCI	DPO	MI	TRIX
2019-01-02	22.749472	65.149857	14.370693	12.0	133.956097	-407.70	23.544825	-0.002928
2019-01-03	122.247274	66.539185	15.342166	16.0	173.451848	350.30	24.276654	0.009336
2019-01-04	257.545418	67.244860	16.677983	20.0	192.868191	-138.75	24.793426	0.027849
2019-01-07	405.337444	66.470944	17.918385	20.0	167.915119	-1342.50	24.793470	0.049369
2019-01-08	557.601866	67.417952	18.884107	28.0	142.559683	-1282.10	24.631275	0.072607
2019-01-09	728.640591	71.529452	20.345567	32.0	155.125199	-2456.45	24.585955	0.099370
2019-01-10	903.899036	71.994056	21.837534	36.0	142.367840	-3146.00	24.631016	0.127537
2019-01-11	1068.193559	71.036733	23.222932	36.0	121.703330	-3105.15	24.649067	0.154798
2019-01-14	1227.587693	73.156165	24.714258	44.0	114.752878	-4068.85	24.748801	0.181744
2019-01-15	1366.087788	70.317550	26.187473	44.0	100.643087	-4098.80	24.826881	0.205968
2019-01-16	1486.494547	71.284944	27.555459	44.0	92.504829	-1999.55	24.696386	0.227616
2019-01-17	1601.145142	73.890076	29.334361	56.0	99.482523	730.40	24.804345	0.248230
2019-01-18	1714.430371	75.736105	31.231055	60.0	108.782060	797.55	25.000300	0.268359
2019-01-21	1815.625648	75.069511	32.392993	60.0	95.637311	604.55	25.174808	0.286502
2019-01-22	1885.329522	68.318487	33.242366	60.0	82.502481	-8.90	25.263697	0.299909
2019-01-23	1950.557829	72.578727	34.253051	72.0	95.243222	-240.35	25.376378	0.312133
2019-01-24	2022.968838	75.326670	35.625920	76.0	116.975967	741.65	25.572475	0.324809
2019-01-28	2055.848954	61.978936	35.068378	76.0	83.757454	419.25	25.807869	0.331655
2019-01-29	2060.653658	62.602148	34.550661	76.0	79.680662	-237.70	25.950631	0.334062
2019-01-30	2064.562862	66.693532	34.262874	76.0	98.043483	122.85	26.308427	0.335661

Fonte: Própria (2021)

4.2 Treinamento e Saída dos Modelos

Para avaliação dos modelos, foi decidido treinar cada algoritmo uma vez para cada ano em que serão calculadas as métricas, cada modelo contará com os dados oito anos anteriores ao período em que está sendo avaliado. Deste modo, para as previsões do ano de dois mil e desesseis, por exemplo, as últimas amostras utilizadas para treinamento serão do ano de dois mil e quinze. Para algoritmos como o GBR que necessitam de amostrar de validação dos hiperparâmetros, o último ano de treinamento será retirado para este propósito.

As métricas que podem ser utilizadas com regressão citadas no capítulo anterior, podem ser acopladas diretamente na saída das previsões feitas pelo esquema de treinamento dos modelos, uma vez que eles também são regressões, porém para avaliar o resultado e desempenho dos modelos nas métricas de classificação é necessário transformar a saída e rótulos analógicos para digitais. Como o experimento final de tomada de decisão de aporte toma em conta apenas dois cenários, as saídas e rótulos serão binarizados para valorização, representado por 1, e desvalorização representado por zero. Essa conversão pode ser feita de forma simples apenas utilizando a condição do sinal do valor previsto pela

saída do modelo e dos rótulos do alvo do período observado, caso seja negativo pertencerá a previsão de classe 0, caso seja neutro ou positivo pertencerá a previsão de classe 1. Utilizando tal transformação é possível calcular as métricas faltantes sem a necessidade de treinar os modelos novamente como classificações. Abaixo podem ser observados exemplos da transformação do alvo para as mesmas amostras apresentadas na figura 15:

Figura 17 – Exemplificação de alvo transformado por amostra

Date	
2019-01-02	1
2019-01-03	1
2019-01-04	1
2019-01-07	1
2019-01-08	1
2019-01-09	1
2019-01-10	1
2019-01-11	1
2019-01-14	1
2019-01-15	1
2019-01-16	1
2019-01-17	1
2019-01-18	1
2019-01-21	0
2019-01-22	1
2019-01-23	1
2019-01-24	0
2019-01-28	1
2019-01-29	1
2019-01-30	0

Fonte: Própria (2021)

4.3 Experimento

Para analisar o desempenho de cada algoritmo em cada período analisado na tomada de decisão de investimentos, será realizado um experimento utilizando as previsões geradas e os valores de retorno reais do índice no período.

O experimento consiste em semanalmente, no primeiro pregão realizado, analisar a saída do modelo em questão e verificar se a saída é uma previsão de valorização ou desvalorização. Caso haja uma previsão de valorização um valor fictício inicial será multiplicado pelo real retorno do índice no período, caso haja uma previsão de desvalorização, o valor inicial permanecerá o mesmo, simulando a manutenção do valor em caixa. O valor inicial

para cada ano e para cada cada modelo treinado será de R\$100,00 que será passado à frente com base no resultado de cada semana até a última semana do ano analisado.

O experimento em questão é uma simulação de aporte de recursos apenas comprando fundos que repliquem o índice ou mantendo o valor em caixa. Serão desconsiderados todas as possíveis taxas e encargos que poderiam ser provenientes da transação.

5 Resultados e Discussão

5.1 Treinamento e Testes

5.1.1 ARIMA

Para o treinamento deste algoritmo foi utilizada a implementação `pmdarima`[23]. Este pacote, além de implementar uma interface amigável para o treinamento, também realiza os testes necessários para a escolha da ordem de cada uma das componentes. Os testes para escolha da ordem mais apropriada de um ARIMA, ou um SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors model) podem ser feitos manualmente, mas envolvem diversas etapas que podem ser automatizadas, por isso foi feita a escolha de utilizar a implementação citada acima.

5.1.2 GBR

O GBR é um dos algoritmos mais utilizados e possui algumas variações criadas, como o XGboost criado por Tianqi Chen [18] e o `lightgbm` desenvolvido inicialmente pela Microsoft [19]. O GBR utilizado neste experimento foi a implementação feita pelo pacote `scikit learn`. Esta implementação foi escolhida por fazer parte de um dos pacotes de aprendizado de máquina mais utilizados atualmente e por sua facilidade de uso, bem como sua flexibilidade. Como no último algoritmo descrito, há uma lista de hiperparâmetros passíveis de escolha para o treinamento do modelo. No caso de algoritmos baseados em árvore existe um espaço gigantesco com possibilidades. Especificamente no caso da classe Gradiente Boosting Regressor do `scikit learn`, há mais de 20 hiperparâmetros, o que pode ocasionar em mais de um milhão de tentativas se todas as combinações possíveis forem testadas.

O método de otimização escolhido utiliza um otimizar baseado em árvore, implementado pelo `Optuna`. O `Optuna` conta com uma maneira flexível para realizar a busca, por isso foi escolhido para este caso. A busca foi feita em 50 iterações, isto é, 50 tentativas de melhoria da métrica de pontuação (coeficiente de determinação) no conjunto de validação e o modelo final foi o que obteve o melhor desempenho neste conjunto.

Figura 18 – Demonstração de espaço de procura de hiperparâmetros com o Optuna

```

loss = trial.suggest_categorical('loss', ['ls', 'lad', 'huber', 'quantile'])
learning_rate = trial.suggest_float('learning_rate', 1e-6, 1e-1, log=True)
n_estimators = trial.suggest_int('n_estimators', 10, 1000)
subsample = trial.suggest_float('subsample', 0.1, 1)
criterion = trial.suggest_categorical('criterion', ['friedman_mse'])
min_samples_split = trial.suggest_int('min_samples_split', 2, 50)
min_samples_leaf = trial.suggest_int('min_samples_leaf', 2, 10)
max_depth = trial.suggest_int('max_depth', 2, 15)
max_features = trial.suggest_categorical('max_features', ['auto', 'sqrt', 'log2'])

```

Fonte: Própria (2021)

5.1.3 MLP

O treinamento do MLP, em comparação com os outros algoritmos, é mais simples, uma vez que seus hiperparâmetros seriam a sua topologia e sua ativação. Para a escolha da melhor topologia para este problema foi utilizado o guia de Mansour M. e etc de boas práticas para treinamento de MLPs [20]. A primeira tentativa de topologia foi uma rede sem camadas escondidas, apenas com os neurônios de entrada e um neurônio com ativação linear na saída; após esse teste foram adicionadas sucessivamente camadas escondidas de forma losangular (as camadas mais ao centro da rede sempre tendo mais neurônios do que aquelas mais próximas às camadas expostas). O resultado das sucessivas adições de complexidade no modelo não foram satisfatórios e a cada camada adicionada, como o tempo de treinamento cresce exponencialmente com o incremento no número de neurônios e camadas, fazendo com que uma rede mais simples com apenas duas camadas escondidas fosse a melhor escolha de desempenho mantendo uma diferença baixa entre o desempenho do teste e da validação.

Figura 19 – Topologia escolhida para o treinamento da MLP.

Layer (type)	Output Shape	Param #
dense_8 (Dense)	(None, 8)	72
dropout_6 (Dropout)	(None, 8)	0
dense_9 (Dense)	(None, 12)	108
dropout_7 (Dropout)	(None, 12)	0
dense_10 (Dense)	(None, 6)	78
dropout_8 (Dropout)	(None, 6)	0
dense_11 (Dense)	(None, 1)	7
Total params: 265		
Trainable params: 265		
Non-trainable params: 0		

Fonte: Própria (2021)

5.1.4 LSTM

O treinamento de uma rede LSTM necessita de alguns ajustes extras em comparação a uma MLP padrão. Como a célula LSTM trabalha de forma recorrente é necessário criar um lookback (série de valores passados) na série temporal para que ela entenda a entrada de dados na rede. Neste caso, o tamanho do lookback pode ser considerado um hiperparâmetro a ser otimizado. Quatro valores de lookback diferentes foram testados: 50, 100, 150 e 200, e não houve perda ou incremento considerável para nenhum destes valores. O valor considerado na seção de resultados será o modelo com o lookback de 50 amostras, pois juntamente com o incremento de camadas foi o arranjo com desempenho ligeiramente acima dos outros.

Como uma rede LSTM também é uma rede neural artificial, todos os passos anteriores utilizados no treinamento da MLP também são válidos para esse modelo. Desta forma foram testadas diversas topologias até encontrar a que teve o melhor desempenho no conjunto de validação.

Adicionando camadas ocultas ou número de neurônios não afetaram diretamente o desempenho da rede LSTM, porém as disposições com uma camada LSTM de entrada, uma camada LSTM escondida e um neurônio padrão de saída obtiveram um decréscimo na função perda escolhida no treinamento. A topologia final da rede LSTM foi:

Figura 20 – Topologia escolhida para o treinamento da rede LSTM.

Layer (type)	Output Shape	Param #
lstm_20 (LSTM)	(None, None, 24)	4704
dropout_53 (Dropout)	(None, None, 24)	0
lstm_21 (LSTM)	(None, None, 6)	744
dropout_54 (Dropout)	(None, None, 6)	0
dense_54 (Dense)	(None, None, 1)	7

=====
 Total params: 5,455
 Trainable params: 5,455
 Non-trainable params: 0

Fonte: Própria (2021)

5.2 Métricas de Desempenho

As métricas mostradas a seguir são os resultados das previsões dos modelos para cada semana do ano seguinte aos dispostos para treinamento. Ao final da exposição dos resultados será feita uma breve discussão sobre os valores obtidos.

5.2.1 Erro Absoluto Médio

Tabela 2 – Resultados dos modelos por ano para a métrica Erro Absoluto Médio

Modelo/Ano	2016	2017	2018	2019	2020
ARIMA	0.0292	0.0202	0.0230	0.0186	0.0340
GBR	0.0297	0.0198	0.0230	0.0181	0.0339
MLP	0.0290	0.0202	0.0229	0.0182	0.0339
LSTM	0.0293	0.0199	0.0228	0.0186	0.0343

5.2.2 Erro Quadrático Médio

Tabela 3 – Resultados dos modelos por ano para a métrica Erro Quadrático Médio

Modelo/Ano	2016	2017	2018	2019	2020
ARIMA	0.001417	0.000648	0.000852	0.000516	0.002919
GBR	0.001472	0.000634	0.000839	0.000501	0.002897
MLP	0.001418	0.000640	0.000840	0.000504	0.002897
LSTM	0.001446	0.000633	0.000847	0.000518	0.002935

5.2.3 Coeficiente de Determinação

Tabela 4 – Resultados dos modelos por ano para a métrica Coeficiente de Determinação

Modelo/Ano	2016	2017	2018	2019	2020
ARIMA	-0.0347	-0.0569	-0.0099	-0.0281	-0.0067
GBR	-0.0746	-0.0341	0.0052	0.0018	0.0008
MLP	-0.0348	-0.0448	-0.0004	-0.0040	0.0009
LSTM	-0.0558	-0.0321	-0.0043	-0.0314	-0.0120

5.2.4 Acurácia

Tabela 5 – Resultados dos modelos por ano para a métrica Acurácia

Modelo/Ano	2016	2017	2018	2019	2020
ARIMA	0.482	0.498	0.500	0.490	0.522
GBR	0.474	0.559	0.508	0.534	0.551
MLP	0.542	0.546	0.504	0.595	0.559
LSTM	0.462	0.571	0.549	0.607	0.538

5.2.5 Precisão

Tabela 6 – Resultados dos modelos por ano para a métrica Precisão

Modelo/Ano	2016	2017	2018	2019	2020
ARIMA	0.597	0.593	0.556	0.642	0.625
GBR	0.583	0.679	0.566	0.698	0.613
MLP	0.618	0.581	0.560	0.663	0.642
LSTM	0.598	0.606	0.564	0.645	0.593

5.2.6 Revocação

Tabela 7 – Resultados dos modelos por ano para a métrica Revocação

Modelo/Ano	2016	2017	2018	2019	2020
ARIMA	0.484	0.496	0.507	0.491	0.507
GBR	0.503	0.490	0.507	0.503	0.676
MLP	0.667	0.850	0.515	0.770	0.595
LSTM	0.379	0.796	0.846	0.882	0.730

5.2.7 Métrica F1

Tabela 8 – Resultados dos modelos por ano para a métrica F1

Modelo/Ano	2016	2017	2018	2019	2020
ARIMA	0.534	0.541	0.531	0.556	0.560
GBR	0.540	0.569	0.535	0.585	0.643
MLP	0.642	0.690	0.536	0.712	0.618
LSTM	0.464	0.688	0.676	0.745	0.655

5.2.8 Comentários sobre os Resultados

De uma forma geral foi possível observar que todos os algoritmos utilizados neste trabalho tiveram resultados muito semelhantes em todas as métricas utilizadas. Dentre as três métricas para regressões utilizadas, não houve nenhum modelo que foi capaz de ter resultados consistentemente acima dos outros modelos utilizados. Também é possível constatar que dentre essas métricas não houve nenhum valor consideravelmente diferente para nenhum dos modelos, sugerindo que os dados utilizados não deixam margem para um aprendizado e inferência melhor utilizando apenas os indicadores técnicos apresentados.

Dentre as métricas de classificação houve uma dispersão maior dos valores para alguns algoritmos, resultado oriundo da binarização dos valores analógicos da saída dos modelos. Novamente nenhum modelo obteve resultados consistentes acima de seus pares.

Uma vez constatado que não houve aprendizado considerável na previsão dos modelos testados, não seria possível defender a eficiência dos algoritmos para estratégias de investimentos mesmo que houvesse resultados positivos, porém a simulação foi realizada para analisar a coerência entre as métricas citadas anteriormente e o experimento sugerido.

5.3 Experimento: Tomada de decisão

A simulação leva em conta um patrimônio inicial de R\$100,00. ao começo de cada semana, os modelos serão consultados sobre a previsão de alta ou queda do índice analisado, caso o modelo constate tendência de alta, o retorno do índice naquela semana será o multiplicador do patrimônio, caso contrário, o patrimônio ficará inalterado. Os períodos analisados serão anuais, assim como as métricas anteriores. Os resultados obtidos foram:

Tabela 9 – Resultados dos modelos por ano

Modelo/Ano	2016	2017	2018	2019	2020	Acumulado
ARIMA	170.28	124.43	112.62	122.46	77.51	226.50
GBR	110.58	112.07	123.55	133.20	96.25	196.30
MLP	111.62	116.38	113.77	132.76	88.67	173.98
LSTM	100.36	120.73	119.83	115.96	84.57	142.38
IBOV	138.93	126.85	115.03	131.94	102.88	275.17

Como é possível observar, todos os modelos obtiveram um resultado inferior ao índice no acumulado dos cinco anos analisados, além disso, também é possível observar que mesmo com uma dispersão maior entre os retornos anuais, nenhum dos modelos utilizados foi superior aos seus para em todos os períodos.

6 Conclusões e Trabalhos Futuros

6.1 Conclusões

O objetivo principal do projeto foi analisar como diferentes tipos de algoritmos se comportam em problemas de séries temporais utilizando as métricas de regressões e classificações expostas anteriormente, além de realizar simulações nos períodos para verificar a possibilidade de gerar rentabilidade utilizando suas previsões.

Pode-se concluir que apesar das diferentes naturezas e métodos utilizados para o treinamento dos modelos, as métricas dispostas foram extremamente semelhantes entre todos os modelos e períodos analisados, sendo possível afirmar que utilizar os indicadores técnicos expostos no projeto, aliado aos algoritmos de aprendizado de máquina não são um método eficiente para o tratamento de séries temporais financeiras para o índice Bovespa.

O retorno acumulado do índice nos cinco anos analisados foi superior a todos os algoritmos analisados, apesar de haver pontos fora da curva para anos específicos.

6.2 Trabalhos Futuros

Para dar continuidade ao desenvolvimento desse projeto, seria necessário a reformulação nas hipóteses de treinamento dos algoritmos ou sua seleção.

Como a decisão final na saída dos algoritmos são classes binárias, pode-se adotar o treinamento de algoritmos de classificação, isso poderia facilitar a análise e o reconhecimento de padrões por parte dos algoritmos, uma vez que seria perdida apenas as métricas erro médio absoluto, erro médio quadrático e coeficiente de determinação.

Outro ponto a ser considerado é incluir outros indicadores técnicos. Neste experimento, foram utilizados apenas oito indicadores de um leque bem maior de possibilidades. Assim, não deve haver dificuldade na implementação de tais atributos no treinamento, pois, o pacote TA tem implementado muitos outros indicadores além dos utilizados. Fica apenas a tarefa de analisar cada atributo individualmente para entender seu comportamento e faixas possíveis de variação, sendo que há grande possibilidade que seja necessário realizar pré-processamentos para adicioná-los ao modelo.

O período de previsão e configuração dos indicadores também podem ser analisados. O projeto foi realizado com apenas um intervalo, uma semana, porém é extremamente importante considerar outros intervalos para verificar o incremento ou não de desempenho, neste caso testes com intervalos diários ou mensais podem ser proveitosos. Também há a

possibilidade de configurar os intervalos dos indicadores analisados, uma possibilidade que também não foi considerada neste projeto, todos os indicadores utilizados tem janelas de observações para os cálculos de médias móveis.

Referências

- 1 HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. [S.l.]: OTexts, 2018. Citado na página 3.
- 2 MARKOWITZ, H. Portfolio selection. *The Journal of Finance*, v. 7, n. 1, p. 77, 1952. Citado na página 3.
- 3 SHARPE, W. F. Capital asset prices: A theory of market equilibrium under conditions of risk*. *The Journal of Finance*, v. 19, n. 3, p. 425–442, 1964. Citado na página 3.
- 4 FAMA, E. F. The behavior of stock-market prices. *The Journal of Business*, v. 38, n. 1, p. 34, 1965. Citado na página 3.
- 5 FAMA, E. F. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, v. 25, n. 2, p. 383, 1970. Citado na página 3.
- 6 VEGA, J. d. l. *Confusion de confusiones*. [S.l.]: Martino Publishing, 2013. Citado na página 3.
- 7 MURPHY, J. J. *Technical analysis of the financial markets: a comprehensive guide to trading methods and applications*. [S.l.]: New York Institute of Finance, 1999. Citado na página 3.
- 8 EMERSON, S. et al. Trends and applications of machine learning in quantitative finance. In: *8th international conference on economics and finance research (ICEFR 2019)*. [S.l.: s.n.], 2019. Citado na página 3.
- 9 ZUCKERMAN, G. *The Man Who Solved the Market: How Jim Simons Launched the Quant Revolution SHORTLISTED FOR THE FT et MCKINSEY BUSINESS BOOK OF THE YEAR AWARD 2019*. [S.l.]: Penguin Books Ltd, 2019. Citado na página 4.
- 10 LO, A.; MAMAYSKY, H.; WANG, J. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. 2000. Citado na página 5.
- 11 GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*. [S.l.]: OReilly, 2020. Citado 4 vezes nas páginas 15, 17, 18 e 19.
- 12 YAHOO Finance - Stock Market Live, Quotes, Business & Finance News. Yahoo!, 2021. Disponível em: <<https://finance.yahoo.com/>>. Citado na página 20.
- 13 RANAROUSSI. *ranaroussi/yfinance: Yahoo! Finance market data downloader (faster Pandas Datareader)*. 2021. Disponível em: <<https://github.com/ranaroussi/yfinance>>. Citado 2 vezes nas páginas 20 e 21.
- 14 PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, JMLR. org, v. 12, p. 2825–2830, 2011. Citado 2 vezes nas páginas 21 e 22.

- 15 ALKALINE-ML. *alkaline-ml/pmdarima: A statistical library designed to fill the void in Python's time series analysis capabilities, including the equivalent of R's auto.arima function*. 2021. Disponível em: <<https://github.com/alkaline-ml/pmdarima>>. Citado na página 22.
- 16 AKIBA, T. et al. Optuna: A next-generation hyperparameter optimization framework. In: *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.: s.n.], 2019. Citado na página 22.
- 17 VENZEL, L. O. *0zym4nd145/bovespa-index-prediction: Bovespa Index prediction built as my graduation work*. 2021. Disponível em: <<https://github.com/0zym4nd145/bovespa-index-prediction>>. Citado na página 26.
- 18 CHEN, T.; GUESTRIN, C. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. Citado na página 31.
- 19 KE, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, v. 30, p. 3146–3154, 2017. Citado na página 31.
- 20 MANSOUR, M.; ALSULAMY, S.; DAWOOD, S. Prediction of implementing iso 14031 guidelines using a multilayer perceptron neural network approach. *Plos One*, v. 16, n. 1, 2021. Citado na página 32.