UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

# Abordagens de aprendizado de máquina para séries temporais financeiras

**Júlio Cesar dos Santos**

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
**Instituto de Ciências Matemáticas e de Computação**

# Abordagens de aprendizado de máquina para séries temporais financeiras

***Júlio Cesar dos Santos***

USP - São Carlos

2022

Júlio Cesar dos Santos

# Abordagens de aprendizado de máquina para séries temporais financeiras

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. José F. Rodrigues Jr

USP - São Carlos

2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

SANTOS, JULIO CESAR DOS
S237a Abordagens de aprendizado de máquina para séries temporais financeiras / JULIO CESAR DOS SANTOS; orientador JOSE F. RODRIGUES Jr. -- São Carlos, 2022.
53 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2022.

1. Machine Learning. 2. Random Forest. 3. LigthGBM. 4. Séries Financeiras . I. RODRIGUES Jr, JOSE F., orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

# DEDICATÓRIA

A Deus, sem o qual, NADA é possível.

Aos meus familiares, pelas oportunidades oferecidas e pelo apoio.

# AGRADECIMENTOS

Ao meu orientador Dr. José F. Rodrigues Jr, pelo apoio, compreensão e confiança.

# RESUMO

SANTOS, J. C. **Abordagens de aprendizado de máquina para séries temporais financeiras.** 2022. 52 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2022.

As novas abordagens de aprendizado de máquina (Machine Learning - ML) utilizando aprendizado profundo (Deep Learning – DL) supera os modelos de séries temporais e geralmente tem se mostrado com maior precisão do que os algoritmos tradicionais de ML. Contudo, esses mesmos modelos (DL) tem como desvantagem a grande quantidade de tempo gasto para treiná-los em uma tarefa sofisticada de customização dos seus hiperparâmetros. Percebe-se que menos atenção tem sido dada a outra classe poderosa de abordagens de ML a saber: Florestas aleatórias (*Random Forest* - RF) e Máquina de Aumento do Gradiente (*Gradient Boosting Machine* - GBM) que se utiliza de técnica *bagging* (RF) e *boosting* (GBM). Eles se mostram menos custosos computacionalmente que os modelos de séries temporais além da atividade de customização dos hiperparâmetros ser bem menos complexa. Diante desta constatação foram escolhidos duas dessas abordagens – *Random Forest* e *LightGBM* – já que representam métodos que se mostram poderosos e podem capturar com eficiência padrões não lineares complexos em dados. A partir da análise destas técnicas buscou-se estabelecer uma metodologia para, sistematicamente, obter uma ferramenta capaz de auxiliar o analista no processo decisório sobre investir, realizar lucros, ou aguardar e traçar conclusões sobre o potencial de uso de técnicas de Aprendizado de Máquina no mercado brasileiro, sugerindo práticas recomendadas e/ou técnicas a serem evitadas. Primeiramente, foi feita a partição dos dados importados em três conjuntos (treino, validação e teste) e essa divisão foram adotadas duas abordagens de separação dos dados: uma utilizando do aspecto temporal do dado e outra uma divisão aleatória. O processo seguiu as etapas de coleta e armazenamento dos dados, tratamento e normalização das séries de preço, análise das propriedades das séries, criação de um novo atributo a partir dos atributos originais, utilização de modelos de previsão e análise dos resultados. Vale ressaltar que os dados também foram rotulados utilizando uma customização do método descrito como barras de tempo. Concluindo, considerando o problema investigado, apesar da divisão aleatória apresentar medidas mais eficientes é pertinente que se utilize a divisão temporal para um sistema real. Com relação aos algoritmos, o LGBM se mostrou melhor mesmo não passando por nenhuma otimização dos seus hiperparâmetros.

Palavras-chave: *Machine learning, Random forest, LightGBM,* séries temporais financeiras.

# ABSTRACT

SANTOS, J. C. **Machine learning approaches to financial time series**. 2022. 52 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2022.

New machine learning approaches (Machine Learning - ML) utilizing deep learning (Deep Learning - DL) outperforms time series models and generally proves to be more accurate than traditional ML algorithms. However, these same models (DL) have the disadvantage of a large amount of time spent to train them in a sophisticated task of customizing their hyperparameters. It is noticed that less attention is given to another powerful class of ML approaches namely: Random Forests (Random Forest - RF) and Gradient Boosting Machine (GBM) that uses the bagging technique (RF) and booster (GBM). They are less computationally expensive than time series models, in addition to the hyperparameter customization activity being much less complex. In view of this finding, the two of these approaches - Random Forest and LightGBM - were chosen as they represent methods that prove to be powerful and can efficiently capture complex nonlinear patterns in data. From the analysis of these techniques, we sought to establish a methodology to systematically obtain a tool capable of assisting the analyst in the decision-making process about investing, making profits, or waiting and drawing conclusions about the potential use of Machine Learning techniques in the Brazilian market, suggesting recommended practices and/or techniques to be avoided. First, the imported data was partitioned into three sets (training, validation and test) and for this division two approaches were adopted for separating the data: one using the temporal aspect of the data and the other a random division. The process followed the steps of collecting and storing data, processing and normalizing the price series, analyzing the properties of the series, creating a new attribute from the original attributes, using forecast models and analyzing the results. It is worth mentioning that the data were also labeled using a customization of the described method as time bars. In conclusion, considering the problem investigated, although random division presents more efficient measures, it is pertinent to use temporal division for a real system. Regarding the algorithms, the LGBM proved to be better even though it did not undergo any optimization of its hyperparameters.

Keywords: Machine Learning, Random Forest, LightGBM, financial time series.

# Lista de Figuras

# Lista de Tabelas

# Sumário

# 1. Introdução

O avanço da Inteligência Artificial e áreas correlatas têm permitido que novos modelos sejam utilizados no processamento de séries temporais de preços de ativos do mercado acionário [2]. Este tipo de processamento visa à previsão do comportamento dos preços, o que sempre foi objeto de estudo e de análise por parte de investidores e pesquisadores da área [2]; o intuito é maximizar os retornos (lucro) e diminuir os riscos de uma carteira de investimentos.

## 1.1. Problema

O problema consiste em prever o comportamento das séries de preços dos ativos para uma operação futura. Trata-se de um problema com caráter interdisciplinar pois envolve temas de áreas como finanças, matemática, estatística, economia e, mais recentemente, ciência de dados e aprendizado de máquina [2]. Atualmente, com os avanços computacionais e com o acesso a dados históricos dos preços dos ativos, qualquer investidor pode criar estratégias de atuação baseadas em modelos econométricos e/ou estatísticos.

Contudo, o problema ainda se mostra de difícil solução dadas as particularidades das séries em questão, e a influência de fatores exógenos nos preços dos ativos: por exemplo, os fatores de origem política, macroeconômica, microeconômica e os vieses comportamentais presentes nos gestores de investimentos, um tipo de viés típico do comportamento humano [1].

Mais especificamente, dada uma série temporal TA = <v1, v2, …, vt-1, vt>, com t observações, na qual o i-ésimo valor vi, corresponde ao preço de fechamento de um dado ativo de interesse A no i-ésimo dia de observação, deseja-se computar a probabilidade do ativo A entrar no t+1-ésimo dia com preços em alta.

## 1.2. Motivação

A previsão do comportamento de um ativo tem o potencial de aumentar os retornos das operações [1]. Apesar da existência de robôs que operam de forma automatizada por meio da análise técnica, o comportamento dos preços dos ativos ainda se mostra um problema de difícil previsão/identificação estimulando a investigação de outras abordagens. Um dos objetivos é reduzir o viés comportamental dos operadores utilizando-se de técnicas computacionais apoiadas em Inteligência Artificial.

## 1.3. Objetivos

A meta é estabelecer uma metodologia que faça uso de técnicas de Aprendizado de Máquina para, sistematicamente, aumentar a acurácia nas operações conduzidas no mercado de ações. A partir desta meta, espera-se:

- obter uma ferramenta capaz de auxiliar o analista no processo decisório sobre investir, realizar lucros, ou aguardar;
- traçar conclusões sobre o potencial de uso de técnicas de Aprendizado de Máquina no

mercado brasileiro, sugerindo práticas recomendadas e/ou técnicas a serem evitadas.

# 2. Fundamentação - Mercado Financeiro

No Brasil a Bolsa de Valores, Mercadorias e Futuros de São Paulo (B3) é responsável por administrar todas as negociações de títulos, valores mobiliários e contratos derivativos. Ela realiza serviços de registro, compensação, liquidação e atua como contraparte garantidora da liquidação financeira das operações realizadas pelos investidores. Em sua página Web (www.b3.com.br) disponibiliza arquivos de séries temporais de cotações históricas de preços dos títulos negociados desde 1986 até os dias atuais[1].

Essas cotações contemplam as principais informações dos ativos, como: nome e código da empresa, código da ação, código ISIN, tipo de mercado (a vista, termo, opções), especificação do tipo ordinárias (ON) ou preferenciais (PN), preços (anterior, abertura, mínimo, médio, máximo, fechamento), quantidade de negócios e volume negociado com o papel, dentre outros dados disponíveis. A B3 disponibiliza um arquivo de layout[2] com os detalhes sobre o formato e o conteúdo, além de também disponibilizar um tutorial para aqueles que queiram importá-lo no Excel.

Normalmente esses arquivos possuem o seguinte formato:

Nome do Arquivo: COTAHIST.AAAA.TXT
Tipos de Registros: Cada arquivo é composto por três tipos de registros.
• Registro - 00 - Header
• Registro - 01 - Cotações dos papéis por dia
• Registro - 99 - Trailer

E para cada tipo de registro existe uma tabela explicativa da seguinte forma:

| NOME DO CAMPO / DESCRIÇÃO | CONTEÚDO | TIPO E TAMANHO | POSIÇÃO INICIAL | POSIÇÃO FINAL |
|---|---|---|---|---|

1. NOME DO CAMPO / DESCRIÇÃO: descreve o nome do campo propriamente dito.
2. CONTEÚDO: Indica o formato, o significado e as tabelas anexas (páginas 4, 5 e 6) referentes ao campo.
3. TIPO E TAMANHO: Aponta qual o tipo e o tamanho do campo:
    - N: Numérico;
    - X: Alfanumérico;
    - V: Indica que o número possui vírgula;
        - ( ): Quantidade de caracteres antes da vírgula;

[1] https://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/historico/mercado-a-vista/series-historicas/

[2] https://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/historico/mercado-a-vista/cotacoes-historicas/

- (99): Quantidade de caracteres depois da vírgula

- Exemplos:
  - N(03): Campo Numérico com 3 dígitos;
  - X(10): Campo Alfanumérico com 10 caracteres;
  - N(11)V(99): Campo Numérico com 11 caracteres antes da vírgula e 2 após.

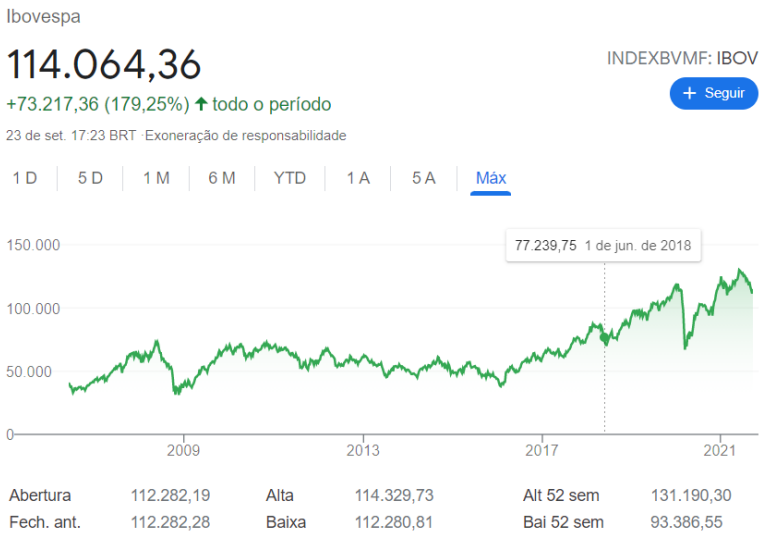
4. POSIÇÃO INICIAL E POSIÇÃO FINAL: Indicam onde começam e terminam os campos.

Como se pode perceber estes arquivos representam séries de preços ordenados no tempo, o que representa uma série temporal.

Uma série temporal pode ser definida como um conjunto de observações sequenciais ao longo do tempo [2]. Morettin (2006 apud [2]) cita que um dos principais objetivos ao estudá-las é poder realizar previsões de valores futuros da série, geralmente de curto prazo.

Nas mais variadas áreas de conhecimento pode-se citar exemplos de séries temporais: valores de temperatura ao longo do ano, valores diários de poluição, precipitação atmosférica em um local, registro de marés, registro de roubos de veículo em uma cidade, entre muitos outros. Pommeranzenbaum (2014 apud [2]) afirma que um grupo de séries que se destaca são as séries financeiras, geralmente compostas por cotações históricas de ativos da bolsa de valores.

Figura 1 - Índice da bolsa brasileira (IBOV)




Fonte: https://www.google.com/finance/quote/IBOV:INDEXBVMF?sa=X&ved=2ahUKEwjOlZekkff5AhWkr5UCHaHgAcoQ3ecFegQIGhAg

A *Figura 1 - Índice da bolsa brasileira (IBOV)* representa a cotação do índice da bolsa brasileira – índice Ibovespa (IBOV) - negociado no mercado financeiro brasileiro com recortes anuais.

## 2.1. Estruturas dos Dados Financeiros

A escolha pela informação vinda da B3 se justifica pois, geralmente, não se deseja trabalhar com um conjunto de dados já processados, pois eles provavelmente irão te mostrar/contar algo que alguém já sabe e não te fornecerão nenhuma vantagem [1]). Um dos pontos de partida de qualquer investigação é uma coleção de dados financeiros não estruturados e dos quais se deriva um conjunto de dados estruturados acessível a algoritmos de *Machine Learning* (ML).

Neste ponto, começam os desafios já que os dados poderão estar em várias formas e formatos. A partir desta constatação necessita-se identificar métodos de organização dos dados para a devida análise e emprego de técnicas com o intuito de escolher a melhor forma de organizar e tratar os dados para a devida atividade de previsão.

A seguir, serão descritos alguns métodos de organização dos dados financeiros presentes em [1] com o intuito de identificar a melhor forma para a devida atividade de previsão. Vale antecipar um conceito muito usado no mercado financeiro que são os gráficos de barra apresentados na próxima seção.

### 2.1.1. Barras - BARS

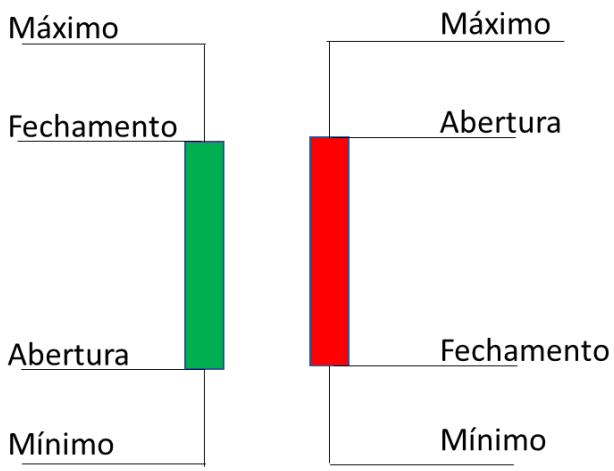
A maioria dos algoritmos de ML consideram uma forma estruturada (tabelas de dados) para as informações a serem analisadas. Os profissionais de finanças costumam se referir às linhas dessas tabelas como "barras". Podemos distinguir entre duas categorias de métodos de barra:

- métodos de barra padrão, que são comuns na literatura;
- métodos mais avançados, baseados em informações, que profissionais especializados usam, embora não possam ser encontrados (ainda) em artigos e/ou periódicos.

Neste ponto vale introduzir um conceito relacionado à forma tradicional de apresentar o movimento de ativo considerando um intervalo de tempo. A Figura 2 - Representação de *candlestick* mostra a representação de uma barra padrão de representação do preço de um ativo chamada *candlestick*. Esta representação toma um dado momento como referência (podendo ser minuto, dia, semana etc.). Para o melhor entendimento da figura, pode-se descrevê-la da seguinte forma:

- As cores representam *candlestick* de alta (verde) ou de baixa (vermelho).
- Abertura: preço de abertura do ativo para o momento em questão.
- Mínimo: preço mínimo que o ativo atingiu para o momento em questão.
- Máximo: Preço máximo que o ativo atingiu para o momento em questão.
- Fechamento: preço de fechamento do ativo para o momento em questão.

Figura 2 - Representação de *candlestick*




Fonte: Autor

Na Figura *3* - Gráfico de barras do índice Ibovespa é apresentada a cotação do índice Ibovespa (IBOV) com a representação de barras em intervalos diários.

Figura 3 - Gráfico de barras do índice Ibovespa


*Fonte:* TradingView - https://br.tradingview.com/chart/

A *Figura 3 - Gráfico de barras do índice Ibovespa* apresenta a cotação diária do índice Ibovespa.

### 2.1.2. Barras Padrão

Alguns métodos de construção de barras são muito populares no setor financeiro, a ponto de a maioria das APIs dos fornecedores de dados oferecerem vários deles. O objetivo desses métodos é transformar uma série de observações que chegam com frequência irregular (normalmente se refere a uma "série não homogênea") em uma série homogênea.

### 2.1.3 Barras de tempo

As barras de tempo são obtidas por amostragem de informações em intervalos de tempo fixos, por exemplo, uma vez a cada minuto. As informações coletadas geralmente incluem:

- *Timestamp*
- Volume - preço médio ponderado (VWAP)
- Preço aberto (ou seja, primeiro)
- Preço de fechamento (ou seja, último)
- Preço máximo
- Preço mínimo
- Volume negociado
- etc.

Embora os intervalos de tempo sejam talvez os mais populares entre profissionais e acadêmicos, eles devem ser evitados por dois motivos. Primeiro, os mercados não processam informações em um intervalo de tempo constante. A hora seguinte à abertura é muito mais ativa do que a hora ao redor do meio-dia (ou a hora por volta da meia-noite no caso de futuros). Se considerarmos a operação realizada por agentes humanos isso se explica pela forma como as pessoas se organizam.

Contudo os mercados de hoje são operados por algoritmos que negociam sem supervisão humana, para os quais os ciclos de processamento da CPU são muito mais relevantes do que intervalos cronológicos (Easley, López de Prado e O 'Hara [2011] apud [1]). Isso significa que o tempo não captura as informações de sobreamostragem durante os períodos de baixa atividade e as informações insuficientes durante os períodos de alta atividade.

Em segundo lugar, as séries com amostragem de tempo frequentemente exibem propriedades estatísticas pobres, como correlação serial, heterocedasticidade e não normalidade de retornos (Easley, López de Prado e O 'Hara [2012] apud [1]).

Como será visto a seguir, serão tratadas formas de agrupamento dos dados em barras como um processo subordinado à atividade de negociação para evitar esse problema inicial.

### 2.1.4. Barras de ordens/transações

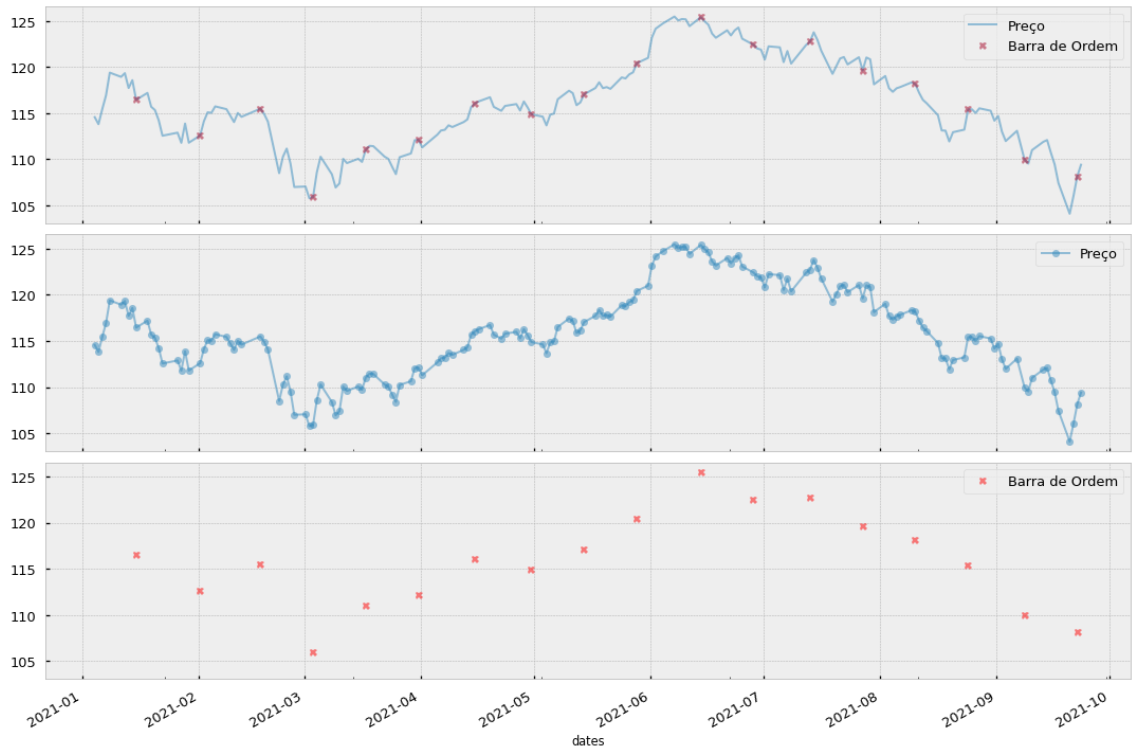
A ideia por trás das barras de ordens é simples: as variáveis de amostra listadas anteriormente (Timestamp, VWAP, preço de abertura etc.) serão extraídas cada vez que um número pré-definido de ordens ocorrer, por exemplo, 1.000 ordens. Isso permite sincronizar a amostragem baseada num valor arbitrário definido de ordens.

Mandelbrot e Taylor (1967 apud [1]) foram alguns dos primeiros a perceber que a amostragem em função do número de ordens exibia propriedades estatísticas desejáveis. ” As mudanças de preço em um número fixo de ordens podem ter uma distribuição gaussiana. Mudanças de preços ao longo de um período fixo podem seguir uma distribuição paretiana estável, cujo a variância é infinita. Uma vez que o número de ordens em qualquer período é aleatório, as declarações acima não estão necessariamente em desacordo.”

Intuitivamente, as barras de ordens permitem uma inferência melhor do que as barras de tempo. Ao construir barras de ordens, é preciso estar ciente dos valores discrepantes. Além disso, muitas bolsas realizam um leilão na abertura e um leilão no fechamento. Isso significa que, por um período, o livro de pedidos acumula lances e ofertas sem igualá-los. Isso quer dizer que várias ordens são lançadas tanto de compra como de venda sem que sejam ‘casadas’ para a efetiva liquidação das operações.

Quando o leilão termina, um grande negócio é publicado ao preço de compensação, por um valor descomunal. Este tipo de abordagem pode ser equivalente a milhares de ordens, embora seja relatado apenas como uma ordem, trazendo distorções na interpretação e na distribuição dos dados.

Figura 4 - Barras de ordens/transações do ativo BOVA11




Fonte: Autor

A *Figura 4: Barras de ordens/transações do ativo BOVA11* apresenta a cotação com o preço de fechamento para o ativo BOVA11 agrupados por número de dias (neste caso utilizou-

se o fator de acumulação de 10 dias de movimentação para a representação do gráfico assinalado com X (barra de ordens/transações).

. A partir deste ponto, sempre será apresentado, para efeito de simplificação, os gráficos plotados como linha já que não serão representados todos os preços (mínimo, máximo, abertura, fechamento) mostrados na *Figura 2 - Representação de candlesitck*.

### 2.1.5. Barras de Volume

Um problema com as barras de ordens é que a fragmentação da ordem introduz alguma arbitrariedade no número de ordens. Por exemplo, suponha que haja um pedido pendente na oferta, com um tamanho de dez. Se forem comprados dez lotes, isso seria representado por uma ordem de compra.

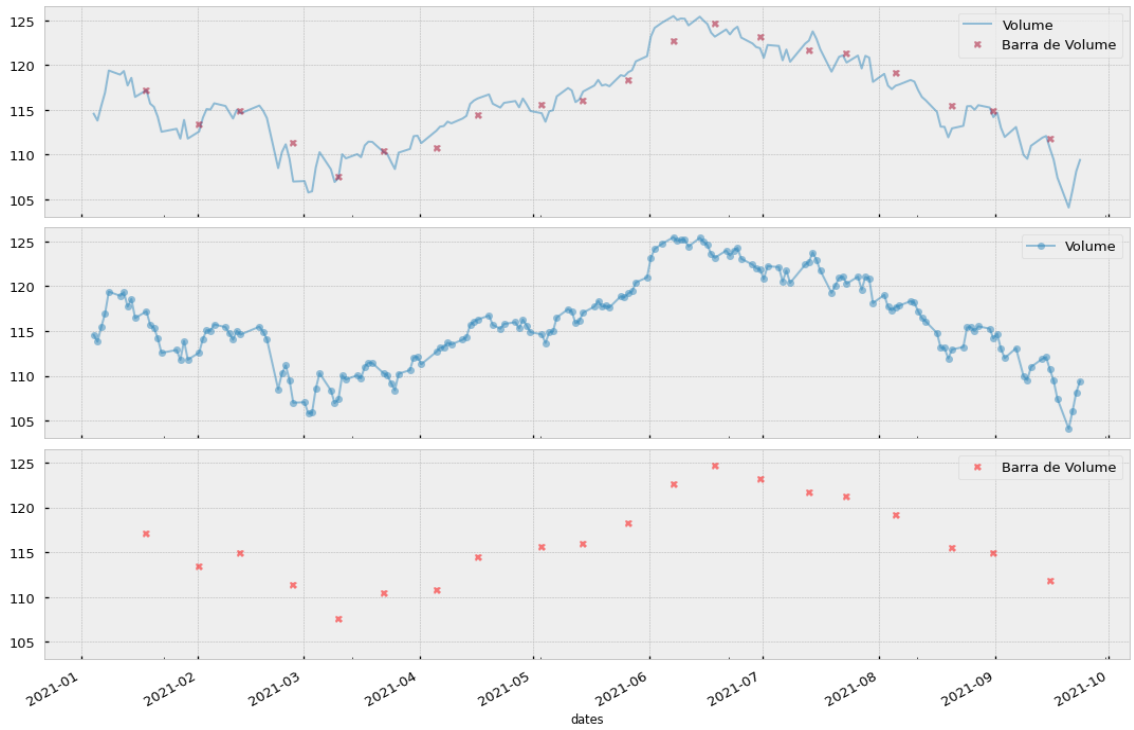
Se, em vez disso, houvesse dez pedidos de tamanho um na oferta, a compra seria registrada como dez ordens separadas. Além disso, por questões operacionais tais ordens podem ser divididas ainda por critérios aleatórios, o que representaria uma divisão artificial da ordem.

Para contornar tal problema as barras de volume sempre agrupam/amostram uma quantidade pré-definida de unidades (ações, contratos futuros etc.) que foi trocada. Por exemplo, poderíamos amostrar os preços sempre que um contrato futuro trocasse mil unidades, independentemente do número de ordens envolvidas.

É difícil imaginar hoje em dia, mas na década de 1960 os fornecedores raramente publicavam dados de volume, pois os clientes estavam mais preocupados com os preços presentes nas ordens.

Depois que o volume começou a ser relatado também, Clark (1973 apud [1]) percebeu que os retornos de amostragem por volume alcançaram propriedades estatísticas ainda melhores (ou seja, mais próximo de uma distribuição gaussiana) do que a amostragem por barras de ordens. Outra razão para preferir barras de volume em vez de barras de tempo ou barras de ordens é que várias teorias de microestrutura de mercado estudam a interação entre preço e volume.

Figura 5 - Barras de volume do ativo BOVA11




Fonte: Autor

A Figura 5 - Barras de volume do ativo BOVA11 apresenta a cotação com o preço de fechamento para o ativo BOVA11 agrupados por volume (neste caso utilizou-se o fator de acumulação de 400.000 ações para a representação do gráfico assinalado com X (barra de volume).

Após descritos alguns métodos, será necessário efetuar a escolha de um deles para a devida atividade de análise e previsão a ser desenvolvida. Contudo, não basta o método em si, se faz necessário também uma forma de rotulação/rotulagem dos dados para que seja possível a utilização de técnicas de ML de forma efetiva. A seguir serão tratadas algumas técnicas de rotulação presentes em [1].

# 3. Métodos de Rotulação/Rotulagem

## 3.1 MOTIVAÇÃO

As definições anteriores se fazem necessárias pois a partir delas será escolhido um formato de dado a ser produzido através das séries temporais dos conjuntos de dados disponíveis – serão utilizados dados de mercado disponibilizados pela B3.

Este tipo de dado permite que um conjunto de dados X seja associado a uma matriz de rótulos ou valores y, de modo que esses rótulos ou valores possam ser previstos em amostras de dados desconhecidos através de algoritmos de aprendizado supervisionado.

Na prática, o que se deseja com estes métodos é definir limites de realização de lucros e *stop-loss* que sejam uma função dos riscos envolvidos em uma aposta.

## 3.2. O MÉTODO DE HORIZONTE DE TEMPO FIXO

No que se refere a finanças, praticamente todos os métodos de ML rotulam as observações usando o método de horizonte de tempo fixo, definido a seguir [1].

Considere uma matriz de características X com L linhas, $\{X_i\}_{i=1,2,\ldots,L}$, extraídas de barras com índice t = 1,...,T, onde L <=T.

No exemplo mostrado na figura 2.4 pode-se ver que a matriz gerada foi composta por 18 pontos representados no eixo Y do terceiro gráfico e o intervalo T do índice de tempo composto por 20 pontos representado no eixo X.

Uma observação $\{X_i\}$ recebe um rótulo $y_i \in \{-1,0,1\}$, de acordo com a equação 3.1:

$$y_i = \begin{cases} -1 & \text{if } r_{t_{i,0},t_{i,0}+h} < -\tau \\ 0 & \text{if } |r_{t_{i,0},t_{i,0}+h}| \leq \tau \\ 1 & \text{if } r_{t_{i,0},t_{i,0}+h} > \tau \end{cases} \qquad (3.1)$$

onde $\tau$ é um limite constante pré-definido; $t_{i,0}$ é o índice da barra imediatamente após $X_i$ ocorrer; $t_{i,0}+h$ é o índice da h-ésima barra após $t_{i,0}$ e $r_{ti,0,\ ti,0+h}$ é o retorno do preço sobre uma barra horizontal h. Este limite usualmente utiliza-se do valor do desvio padrão exponencialmente ponderado dos retornos.

Considerando a condição (if $_{rti,0,\ ti,o+h}$) para rotulação apresentada na fórmula acima se faz necessária a definição de como tal condição será satisfeita. A fórmula abaixo representa a razão entre o preço do ativo no instante $t_i + h$ e $t_i$ subtraído de 1, como observado na Eq. 3.2.

$$r_{t_{i,0},t_{i,0}+h} = \frac{p_{t_{i,0}+h}}{p_{t_{i,0}}} - 1 \qquad (3.2)$$

Como a literatura quase sempre trabalha com barras de tempo, h implica um horizonte de tempo fixo. Um exemplo recente de aplicação deste método está descrito em (Dixon et al. [2016] apud [1]).

Apesar da popularidade, há vários motivos para evitar essa abordagem. Primeiramente porque as barras de tempo não apresentam boas propriedades estatísticas. Além disso, o mesmo limite r, por exemplo, r=1E-2 é aplicado independentemente da volatilidade observada no mercado o que poderia rotular uma observação $y_i = 1$ tanto para um período noturno de volatilidade (r=1E-4) quanto para um período diurno de volatilidade (r=1E-2). Em outras palavras, é um erro muito comum rotular as observações de acordo com um limite fixo nas barras de tempo.

Cada estratégia de investimento tem limites de *stop-loss*, sejam eles auto-impostos pelo gerente de portfólio, reforçados pelo departamento de risco ou acionados por uma chamada de margem. É simplesmente irrealista construir uma estratégia que lucre com posições que teriam sido interrompidas por esses limites na bolsa. O fato de que praticamente nenhuma publicação explica isso, ao rotular as observações, informa algo sobre o estado atual da literatura de investimento (LÓPEZ DE PRADO [2018]).

## 3.3. LIMIARES DINÂMICOS DE COMPUTAÇÃO

Este método de limiares dinâmicos busca calcular a volatilidade diária em pontos intradiários, aplicados a intervalos de tempo dentro do dia, usando uma função que poderia ser o desvio padrão móvel exponencialmente ponderado, por exemplo.

A título de ilustração segue uma sequência de valores aplicando a função ***EWM*** (*Exponential weighted functions*) do pacote pandas. Neste exemplo está sendo calculado o desvio padrão exponencialmente ponderado móvel considerando o intervalo de dois dias (span=2).

Figura 6 - Exemplo da função EWM para cálculo do desvio padrão exponencialmente ponderado

```
import pandas as pd
df = pd.DataFrame({'periodo': [1, 2, 3, 4, 5],
                   'preco': [66.3, 176.3, 186.3, 296.3, 266.3]})
df['2diaEWM'] = df['preco'].ewm(span=2).std()
df
```

| | periodo | preco | 2diaEWM |
|---|---|---|---|
| **0** | 1 | 66.3 | NaN |
| **1** | 2 | 176.3 | 77.781746 |
| **2** | 3 | 186.3 | 46.492348 |
| **3** | 4 | 296.3 | 85.512032 |
| **4** | 5 | 266.3 | 49.160725 |

Fonte: Autor

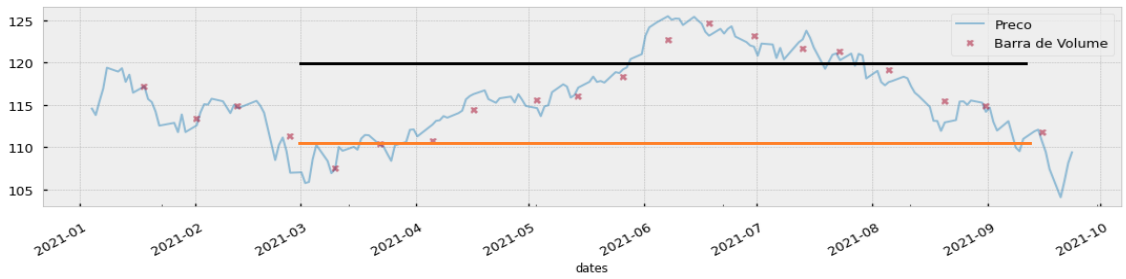
Esta abordagem pode ser utilizada para definir os limites padrão de realização de lucros e prejuízos, pois tem sido usada com sucesso em aplicações reais. Esta técnica pode ser compreendida como uma forma de operar um ativo através das oscilações de sua média móvel. Como exemplo de realização de lucros (*stop-gain*) ou de perdas (*stop-loss*) poderia ser adotado a variação de dois desvios padrões para cima (*stop-gain*) ou para baixo (*stop-loss*) para encerrar uma operação.

## 3.4. O MÉTODO DAS TRÊS BARREIRAS

Este método consiste em definir três barreiras, sendo duas horizontais que definirão um limite inferior (prejuízo) e outro superior (lucro potencial) e a barreira vertical que definirá o número de barras percorridas desde a tomada da posição (um limite de expiração da operação). O processo de rotulação de uma observação é feito de acordo com a primeira barreira que for cruzada pelo gráfico de linha de precificação de um ativo.

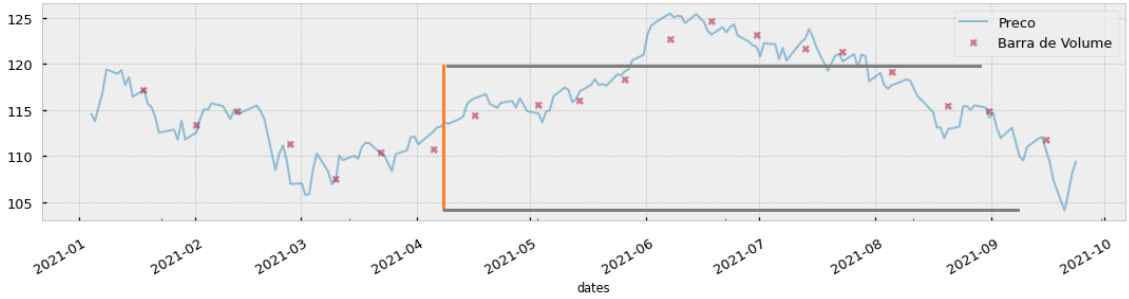
Se a barreira horizontal superior for tocada primeiro, rotula-se a observação como 1. Se a barreira inferior for tocada primeiro, rotula-se a observação como -1. Se a barreira vertical for tocada primeiro, rotula-se a observação com 0.

Figura 7 - Ocorrência de primeiro toque na barreira horizontal inferior



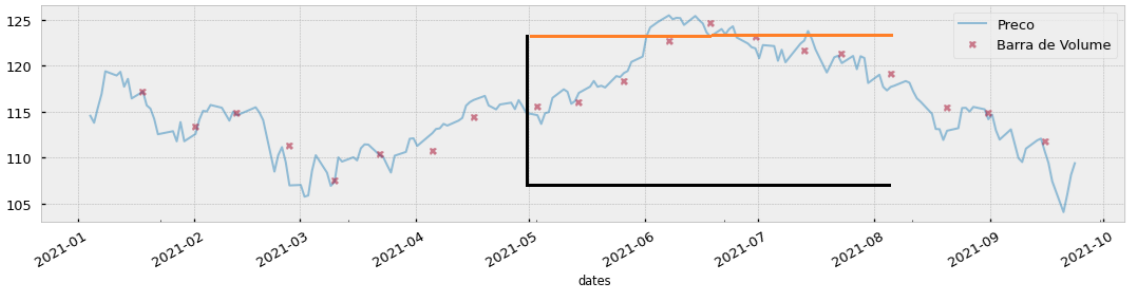

Fonte: Autor

Figura 8 - Ocorrência de primeiro toque na barreira vertical




Fonte: Autor

Figura 9 - Ocorrência de primeiro toque na barreira horizontal superior




Fonte: Autor

## 3.5. AVALIAÇÃO DE APIS

Foi realizada uma análise exploratória de pacotes contendo APIs encontradas na base de softwares livres GitHub, as quais implementam os conceitos presentes em no trabalho de López de Prado [1]:

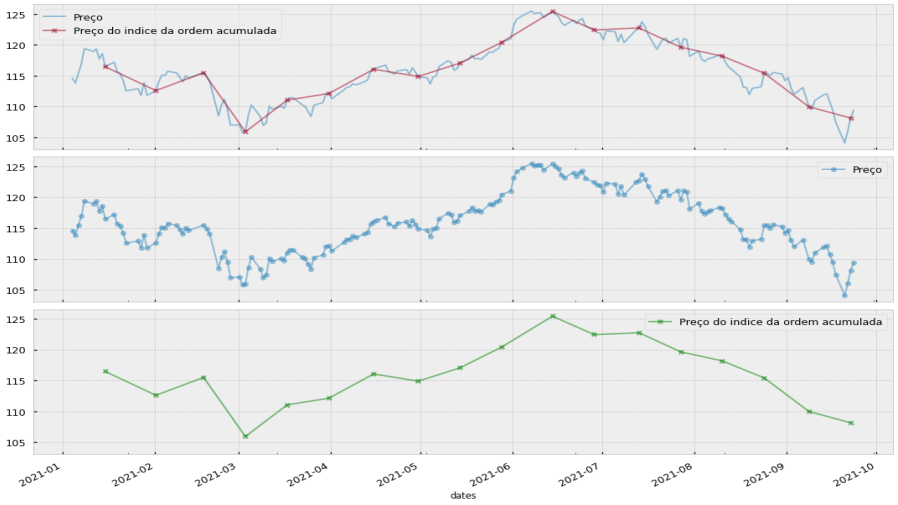
- https://github.com/jjakimoto/finance_ml
- https://github.com/boyboi86/AFML
- https://github.com/BlackArbsCEO/Adv_Fin_ML_Exercises

Considerando estas APIs, percebeu-se que nenhuma delas contém todos os métodos e conceitos definidos no livro e que os códigos são guiados pelos *snnippets* presentes no próprio livro. A API de *boyboi86* (AFML) foca nos exercícios propostos no final de cada capítulo do livro. Diante isso, pretende-se usar os códigos que sejam pertinentes ao projeto sem, contudo, ficar vinculado a apenas uma das APIs. A título de ilustração, a API de *jjakimoto* (finance_ml) possui uma implementação dos conceitos de Barras de Tempo e Barras de Volume conforme ilustrado no livro.

A

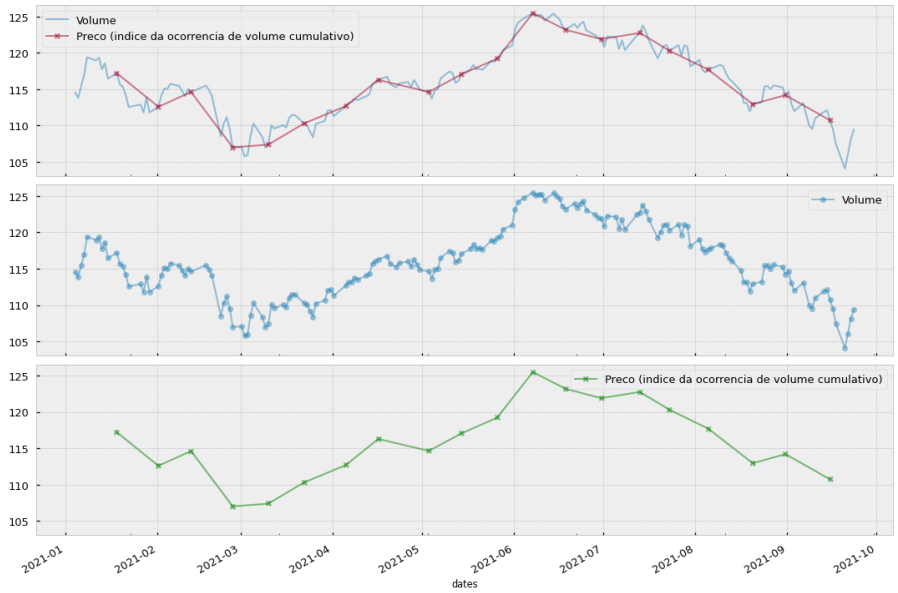
*Figura 10 - Barras de* tempo ilustra o conceito de Barras de tempo, e a *Figura 11 - Barras de Volume* representa o conceito Barras de Volume geradas pela API de *jjakimoto* (finance_ml).

Figura 10 - Barras de tempo




Fonte: Autor

Figura 11 - Barras de Volume




Fonte: Autor

Contudo, percebeu-se que o conceito foi implementado levando em consideração a ocorrência que satisfizesse os critérios mencionados a seguir:

- No contexto de Barras de Tempo, considerando que a acumulação foi representada por 10 dias, a figura destaca o preço da décima ocorrência de fato; isso pode ser percebido observando-se que, no gráfico, a indicação do

preço toca exatamente na série de preços – os itens marcados estão tocando a linha de fato.

- Acontece o mesmo com a Barra de volume. Considerando que a acumulação se deu por 400.000 ações, o preço destacado na figura representa exatamente quando a soma de volume atinge tal valor e o preço em destaque representa exatamente o preço deste evento – os itens marcados estão 'tocando' a linha de fato.

Diante desta constatação de que os preços estão sendo selecionados baseado na ocorrência de fato e não na média acumulada dos movimentos do ativo, foi feita uma extensão à implementação da API para mudar o critério de seleção do preço do ativo, ou seja, computar os preços médios ponderados que satisfizessem as condições escolhidas para acumulação de tempo (barras de tempo) ilustrada na *Figura 14 - Barras de Tempo acumulado* e acumulação de volume (barras de volume) ilustrada na *Figura 17 - Barras de Volume acumulado*.

Para efeito de justificativa de tal alteração no processo de cálculo do preço acumulado, buscou-se suavizar algum evento peculiar que o preço pudesse ter sofrido na observação que satisfizesse a condição de acumulação.

Figura 12 - Código original da função tick_bars

```
def tick_bars(df, price_column, m):
    '''
    compute tick bars

    # args
        df: pd.DataFrame()
        column: name for price data
        m: int(), threshold value for ticks
    # returns
        idx: list of indices
    '''
    t = df[price_column]
    ts = 0
    idx = []
    for i, x in enumerate(tqdm(t)):
        ts += 1
        if ts >= m:
            idx.append(i)
            ts = 0
            continue
    return idx
```

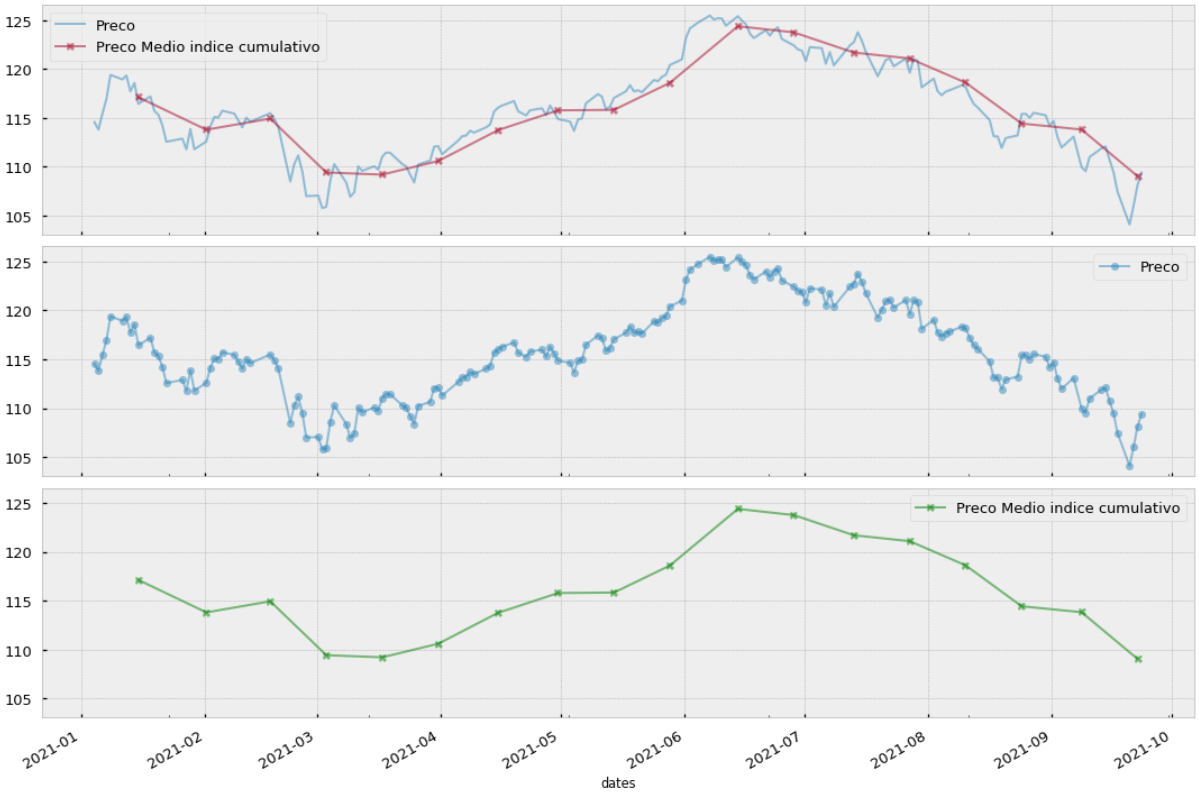
Fonte: Autor

Figura 13 - Extensão para cálculo do preço médio (acúmulo do preço)

```
def tick_bar_df_medio(df, price_column, m):
    idx = tick_bars(df, price_column, m)
    j=0
    lista=[]
    barra1h=df['preco'].mean()+df['preco'].std()*2
    barra2h=df['preco'].mean()- df['preco'].std()
    keys=['dates', 'preco', 'melhorcompra', 'melhorvenda', 'volume', 'v', 'barra1h', 'barrah2']
    for i, x in enumerate(idx):
        data=df.index[x]
        preco=df[j:x+1]['preco'].mean()
        melhorcompra=df[j:x+1]['melhorcompra'].mean()
        melhorvenda=df[j:x+1]['melhorvenda'].mean()
        volume=df[j:x+1]['volume'].sum()
        #barra1h=preco+df[j:x+1]['preco'].std()*2
        #barra2h=preco-df[j:x+1]['preco'].std()
        #indice=df[j:x+1]['Data']
        lista.append({'dates':data, 'preco':preco, 'melhorcompra':melhorcompra, 'melhorvenda':melhorvenda, 'volume':volume,
                      'v':volume, 'barra1h':barra1h, 'barra2h':barra2h})
        barra1h=preco+df[j:x+1]['preco'].std()*2
        barra2h=preco-df[j:x+1]['preco'].std()
        j=x+1
    df_resultado = pd.DataFrame(lista)
    return df_resultado
```

Fonte: Autor

Figura 14 - Barras de Tempo acumulado




Fonte: Autor

Para a justificativa no contexto de barras de tempo poderia ocorrer de a condição ser satisfeita num dia de muita volatilidade ou de um evento externo (macroeconômico) que fizesse com que o preço tivesse uma grande distorção para mais ou para menos.

Figura 15 - Código original da função volume_bars presente na API

```
def volume_bars(df, volume_column, m):
    '''
    compute volume bars

    # args
        df: pd.DataFrame()
        volume_column: name for volume data
        m: int(), threshold value for volume
    # returns
        idx: list of indices
    '''
    t = df[volume_column]
    ts = 0
    idx = []
    for i, x in enumerate(tqdm(t)):
        ts += x
        if ts >= m:
            idx.append(i)
            ts = 0
            continue
    return idx
```

Fonte: Autor

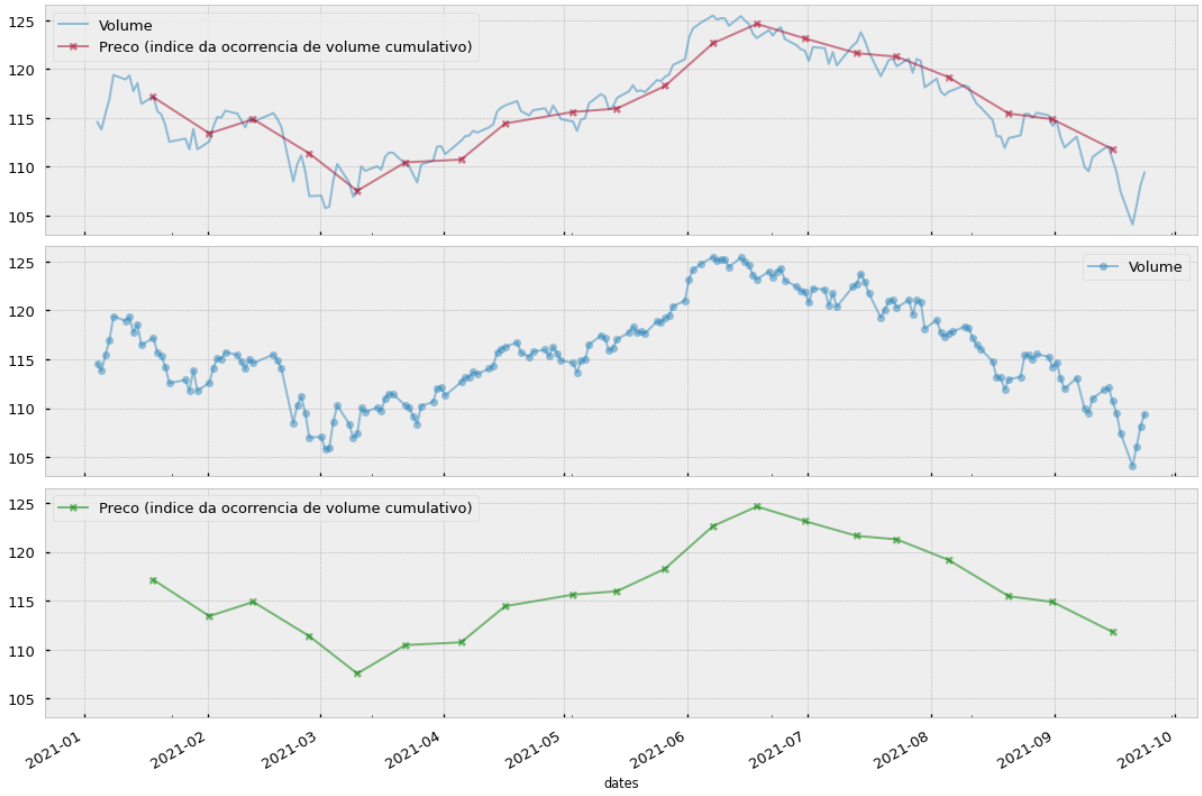
Foi feita a mesma modificação também para o conceito de Barras de Volume ilustrado na Figura *15* - Código original da função volume_bars presente na API conforme apresentado na Figura *16* - Extensão para cálculo do preço médio (Acúmulo de volume).

Figura 16 - Extensão para cálculo do preço médio (Acúmulo de volume)

```
def volume_bar_df(df, volume_column, m):
    idx = volume_bars(df, volume_column, m)
    j=0
    lista=[]
    barra1h=df['preco'][:9].mean()+df['preco'].std()*2
    barra2h=df['preco'][:9].mean()-df['preco'].std()
    keys=['dates', 'preco', 'melhorcompra', 'melhorvenda', 'volume', 'v', 'barra1h', 'barrah2']
    for i, x in enumerate(idx):
        #print(i, x, j, 'i, x, j', df[j:x+1]['volume'].sum(), df.index[x])
        data=df.index[x]
        preco=df[j:x+1]['preco'].mean()
        melhorcompra=df[j:x+1]['melhorcompra'].mean()
        melhorvenda=df[j:x+1]['melhorvenda'].mean()
        volume=df[j:x+1]['volume'].sum()
        #barra1h=preco+df[j:x+1]['preco'].std()*2
        #barra2h=preco-df[j:x+1]['preco'].std()
        #indice=df[j:x+1]['Data']
        lista.append({'dates':data, 'preco':preco, 'melhorcompra':melhorcompra, 'melhorvenda':melhorvenda, 'volume':volume,
                      'v':volume,'barra1h':barra1h, 'barra2h':barra2h})
       # barra1h=preco+df[j:x+1]['preco'].std()*2
       # barra2h=preco-df[j:x+1]['preco'].std()
        j=x+1
    df_resultado = pd.DataFrame(lista)
    return df_resultado
```

Fonte: Autor

Figura 17 - Barras de Volume acumulado




Fonte: Autor

Para a justificativa no contexto de barras de volume poderia ocorrer de a condição ser satisfeita numa ordem muito grande ou muito pequena que distorce o preço em questão, além de podermos ter o mesmo evento externo comentado no contexto de barras de tempo que levasse a uma grande distorção.

# 4. Abordagens de aprendizado de máquina para previsão em séries temporais financeiras

Nos últimos tempos os pesquisadores têm que se concentrado nas novas abordagens de aprendizado de máquina (Machine Learning – ML) utilizando aprendizado profundo (Deep Learning – DL) e percebendo que tal estrutura supera os modelos de séries temporais e geralmente tem se mostrado com maior precisão do que os algoritmos tradicionais de ML. Contudo, esses mesmos modelos (DL) tem como desvantagem a grande quantidade de tempo gasto para treiná-los em uma tarefa sofisticada de customização dos seus hiperparâmetros [3].

Por outro lado, os modelos tradicionais de ML têm se mostrado com precisão comparável na previsão de séries temporais e de certa forma são mais 'leves' em termos computacionais, além da atividade de customização dos hiperparâmetros ser bem menos complexa [3].

Considerando a abordagem e uso dos algoritmos de ML na previsão financeira, os mais comuns são: Redes Neurais Artificiais - RNAs (*Artificial Neural Network* - ANN) de diversas arquiteturas, Máquinas de Vetor de Suporte (*Support Vector Machine* - SVM) e Lógica *Fuzzy*.

Vários estudos apresentam as RNAs com melhores propriedades preditivas do que outras abordagens de ML para prever séries temporais financeiras. Ao mesmo tempo, existem trabalhos de pesquisa que apresentam resultados em que as SVMs demonstram superar outras técnicas não lineares, incluindo técnicas de previsão não linear baseadas em redes neurais, como o *perceptron* multi-camada (*Multi-Layer Perceptron* - MLP) [3].

Além disso, percebe-se que menos atenção tem sido dada a outra classe poderosa de abordagens de ML a saber: Florestas aleatórias (*Random Forest* - RF) e Máquina de Aumento do Gradiente (*Gradient Boosting Machine* - GBM) que se utiliza de técnica *bagging* (RF) e *boosting* (GBM). Ambos os métodos se mostram poderosos e podem capturar com eficiência padrões não lineares complexos em dados.

Em estudos recentes, Varghade e Patel [4 apud 3] testaram o RF e o SVM para a previsão do índice do mercado de ações S&P CNX NIFTY. Eles observaram que o modelo de Árvores de Decisão supera o SVR, embora RF, às vezes, seja 'superajustada' (*overfitting*) aos dados.

## 4.1. Máquina de vetores de suporte (SVM)

A máquina de vetor de suporte (SVM) consiste numa extensão do classificador de vetor de suporte que resulta da ampliação do espaço de recursos de uma maneira específica usando a função kernels. A ideia principal do método SVM é mapear os vetores originais em um espaço de maior dimensão e buscar um hiperplano de separação com margem máxima neste espaço conforme mostrado na Figura *18* - SVM - Hiperplanos.

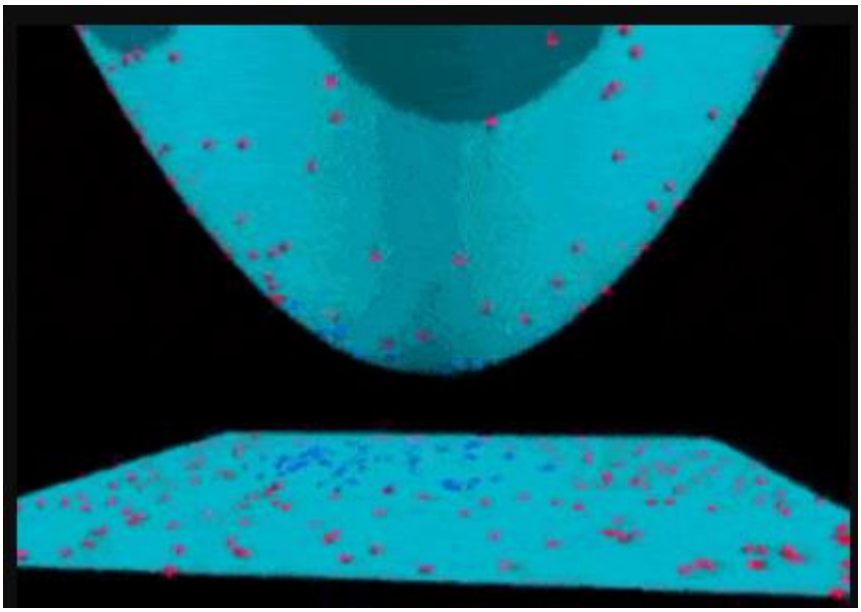
Dois hiperplanos paralelos são construídos em ambos os lados do hiperplano separando as classes. O hiperplano de separação será o hiperplano que maximiza a distância

a dois hiperplanos paralelos. O algoritmo trabalha sob a suposição de que a maior diferença ou distância entre esses hiperplanos paralelos (margem) fornece o menor erro médio do classificador.

*Support Vector Regression* (SVR) é o processo de regressão realizado pelo SVM que tenta identificar o hiperplano que maximiza a margem entre duas classes e minimiza o erro total. Para que um SVM eficiente seja construído, uma penalidade de complexidade também é introduzida, equilibrando a precisão da previsão e o desempenho computacional.

Ao contrário do problema clássico de regressão, o SVR busca coeficientes que minimizem um tipo diferente de perda, onde apenas os resíduos maiores em valor absolutos do que alguma constante positiva contribui para a função de perda. Esta é uma extensão da margem usada nos classificadores de vetor de suporte para a configuração de regressão.

Figura 18 - SVM - Hiperplanos

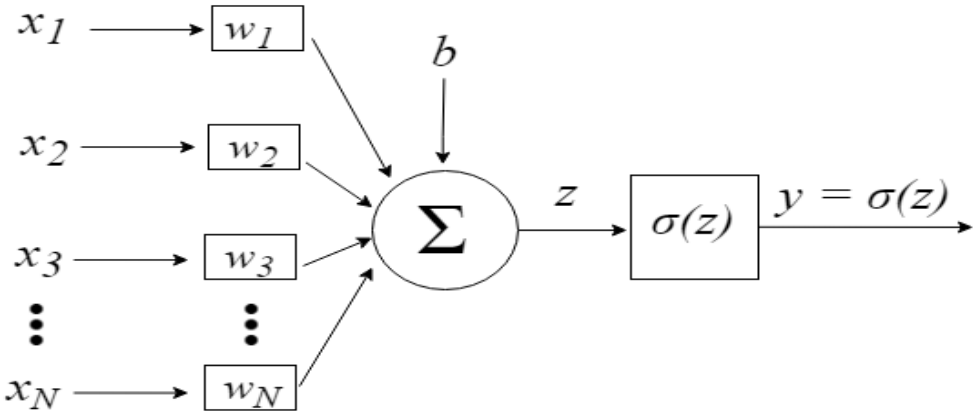


Fonte: https://miro.medium.com/max/333/0*aiT6AJL16jgGmjh_.gif

## 4.2. Rede Neural Artificial (RNA)

As RNAs são os métodos mais populares em ML e são modelos matemáticos que se assemelham às estruturas neurais biológicas e que têm capacidade computacional adquirida por meio de aprendizagem e generalização. Segundo Rosenblatt [7 apud 8] esses modelos almejam semelhança com o sistema nervoso dos seres vivos e a com sua capacidade de processar informações. Estabeleceu-se na área da IA um modelo computacional de um neurônio, conforme ilustrado na Figura *19* – Modelo computacional de um neurônio artificial.

Figura 19 – Modelo computacional de um neurônio artificial



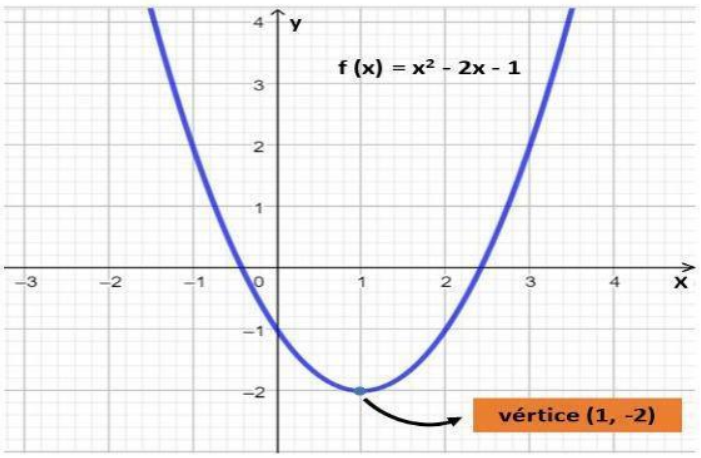

Fonte: Retirada de [8].

Numerosos estudos empíricos mostram a eficiência das RNAs nos diferentes campos tanto para problemas de classificação quanto de regressão: reconhecimento de padrões, controle de processos, séries temporais e assim por diante [9]. A tarefa de aprendizado da rede neural se utiliza de duas técnicas muito poderosas: o gradiente descendente e o *backpropagation*.

## 4.2.1. Gradiente descendente

O gradiente descendente representado na *Figura 20 - Gradiente descendente* é um algoritmo iterativo que, a partir de um ponto inicial, utiliza o gradiente da função como base para escolher uma direção de busca do ponto de mínimo. Para cada etapa da iteração, o algoritmo calcula o gradiente e desloca o ponto candidato a mínimo um passo na direção contrária. Dado tempo suficiente, com um passo pequeno o bastante, o algoritmo converge para, pelo menos, um mínimo local.

Figura 20 - Gradiente descendente




Fonte: https://static.todamateria.com.br/upload/ve/rt/verticeparabola-0.jpg

### 4.2.2. Backprogation

O processo de aprendizado das redes neurais consiste na aplicação de um algoritmo muito poderoso chamado de *backpropagation*, que consiste em, a partir do cálculo do gradiente da última camada, propagar as atualizações dos parâmetros para as camadas anteriores. Para que isso seja possível, é necessário definir uma função de custo na saída da rede.

A sequência de passos que faz a atualização dos pesos pode ser descrita da seguinte forma:

1. Aplica um conjunto de dados de entrada na rede, gerando saídas;
2. Calcula o erro comparando as predições com o target real: utiliza-se de uma função de custo (em regressão - *mse* e classificação – *log-loss*);
3. Aplica o gradiente descendente para atualizar os pesos da última camada;
4. Propaga o gradiente para as camadas anteriores, atualizando os pesos camada por camada.

## 4.3. Máquina de aumento de gradiente (GBM)

*Boosting*[3] é um procedimento para construir sequencialmente uma composição de algoritmos de aprendizado de máquina, quando cada um deles busca compensar as deficiências da composição de todos os algoritmos anteriores. Em contraste com o *bagging*[4], o *boosting* não usa votação simples, mas ponderada. As principais atrações do *boosting* são que é fácil projetar classificadores fracos computacionalmente eficientes (como regra, são usadas árvores de decisão rasas). *Boosting* sobre árvores de decisão é considerado um dos métodos mais eficientes em termos de qualidade de classificação. Uma implementação existente desta técnica é o *LightGBM* (LGBM) distribuição gratuita e de código aberto para aprendizado de máquina desenvolvida originalmente pela Microsoft.

## 4.4. Floresta Aleatória (Random Forest - RF)

O conceito principal do RF é que uma composição de classificadores fracos pode dar bons resultados tanto para problemas de classificação quanto para problemas de regressão. Proposto por Breiman em 1996 ([5, 6] apud [3]) a RF é baseada na técnica de *bagging* (agregação de *bootstrap*) sobre árvores de decisão.

---

3 - Método que tem como procedimento geral a construção de estimadores de forma sequencial, de modo que estimadores posteriores tentam reduzir o viés do estimador conjunto, que leva em consideração estimadores anteriores.

4- Método que tem como procedimento geral construir diversos estimadores independentes, e tomar a média de suas predições como a predição final. O principal objetivo do método é reduzir variância, de modo que o modelo final seja melhor que todos os modelos individuais.

O *bagging* reduz a variância dos algoritmos básicos se eles estiverem fracamente correlacionados. Em RF a correlação entre árvores é reduzida por randomização em duas direções. Primeiramente, cada árvore é treinada em um subconjunto de *bootstrap*. Em segundo lugar, o recurso pelo qual a divisão é realizada em cada nó não é selecionado de todos os recursos possíveis, mas apenas de seu subconjunto aleatório de tamanho *m*.

A principal distinção entre *bagging* e RF é a escolha desses subconjuntos desses recursos. A RF funciona bem quando todos os recursos são pelo menos marginalmente relevantes, pois o número de recursos selecionados para qualquer árvore é pequeno. Usar um valor pequeno de recursos possíveis de um subconjunto aleatório normalmente será útil quando tivermos um grande número de preditores correlacionados.

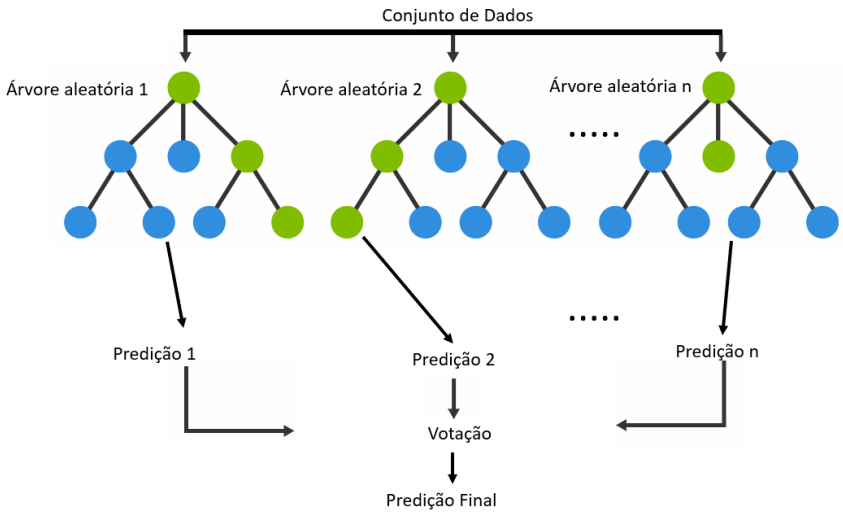
O algoritmo de RF gera cada uma das N árvores de forma independente, o que facilita muito a paralelização. Para cada árvore, ele constrói uma árvore binária completa de profundidade máxima. O conceito principal é que os classificadores (árvores) não corrigem os erros uns dos outros, mas os compensam ao votar. Os classificadores básicos devem ser independentes e podem ser baseados em diferentes grupos de métodos ou treinados em conjuntos de dados independentes. *Bagging* nos permite reduzir o erro de previsão no caso em que a variância do método base de erro é alta.

Assim, a eficiência do desempenho de RF é alcançada, mesmo que algumas árvores consultem recursos inúteis e façam previsões aleatórias. Mas algumas das árvores irão consultar bons recursos e farão boas previsões (porque as folhas são estimadas com base nos dados de treinamento).

Se tivermos árvores suficientes, as aleatórias serão eliminadas como ruído e apenas as árvores “boas” terão efeito no resultado (classificação ou previsão).

O processo descrito anteriormente pode ser ilustrado conforme *Figura 21 - Floresta Aleatória (Random Forest)*:

Figura 21 - Floresta Aleatória (Random Forest)




Fonte: https://www.tibco.com/pt-br/reference-center/what-is-a-random-forest

# 5. Metodologia a ser utilizada

Nesta seção, antes de serem tratadas as etapas da metodologia propriamente dita, vale ressaltar que, das abordagens de aprendizado de máquina para previsão em séries temporais financeiras apresentadas no capítulo 4, serão avaliadas as técnicas de *Random Forest* e LGBM em experimentos usando dados de diferentes ativos do mercado.

## 5.1. Critérios de particionamento dos dados

Primeiramente, o conjunto de dados foi dividido em duas partes (conjunto de treino/validação e conjunto de teste) na proporção de oitenta por cento e vinte por cento respectivamente. O conjunto de testes foi isolado da técnica de treinamento e validação para posterior checagem da validade efetiva do modelo treinado. A primeira partição foi novamente dividida na mesma proporção em conjunto de treino e conjunto de validação e, a partir daí, se deu o treinamento dos modelos. Foram adotadas duas abordagens de particionamento para avaliação dos modelos de previsão, a saber:

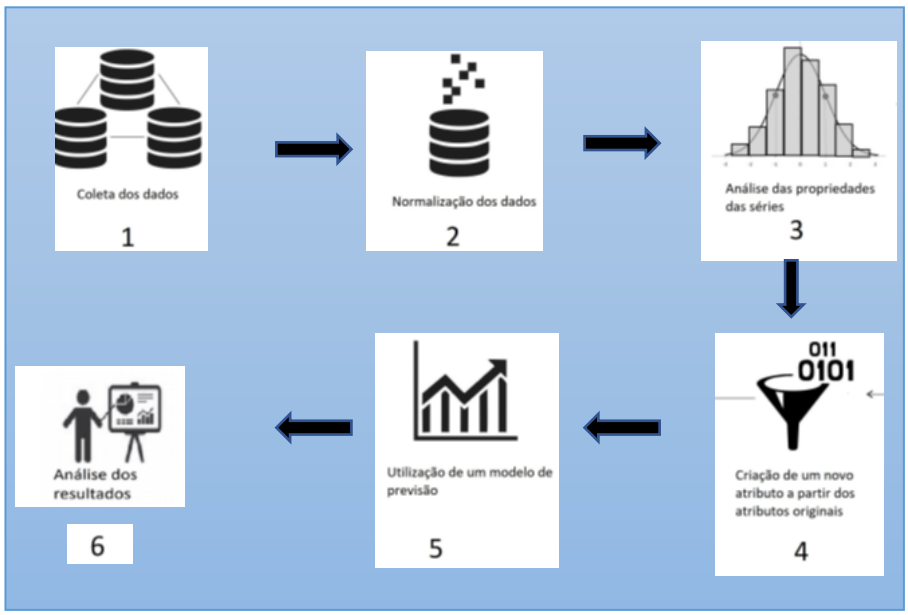
- Separação dos dados utilizando o aspecto temporal (divisão temporal);
- Separação dos dados sem critério (divisão aleatória).

Além disso, utilizou-se também a abordagem descrita no capítulo 3, a qual trata dos acúmulos de preço descritos na seção 3.5. AVALIAÇÃO DE APIS onde se apresenta o conceito de barras de tempo. Para esta abordagem foram utilizados cinco valores de acúmulos (3, 6, 9, 12 e 15 dias).

Depois dessas considerações, a

*Figura 22 - Etapas a serem seguidas na metodologia* do trabalho ilustra as etapas previstas.

Figura 22 - Etapas a serem seguidas na metodologia do trabalho




Fonte: Adaptada de MESQUITA C.M.H.S. R. (2019)

As 6 etapas apresentadas na Figura *22* - Etapas a serem seguidas na metodologia do trabalho são:

- Coleta e armazenamento dos dados;
- Tratamento e normalização das séries de preço;
- Análise das propriedades das séries;
- Criação de um novo atributo a partir dos atributos originais;
- Utilização de um modelo de previsão e
- Análise dos resultados.

## 5.2. Coleta e armazenamento dos dados

Nesta etapa são utilizados dados históricos referentes à bolsa de valores B3 com coleta manual por meio da página Web da B3. Considerando a literatura a ser seguida, será necessário realizar uma filtragem pelos campos de código do ativo, preço médio, preço da melhor compra, preço da melhor venda, e volume de armazenamento deles.

## 5.3. Tratamento e normalização dos dados

Nesta etapa, os dados são tratados buscando-se obter séries consistentes. É realizada uma verificação com a finalidade de remover valores nulos, incorretos ou inconsistentes, além da definição de uma escala de valores homogênea. Após tais tratamentos, as séries de dados são normalizadas. É utilizado o logaritmo do retorno financeiro ao invés dos preços, já que as séries do logaritmo do retorno financeiro são estacionárias em relação à média e possuem propriedades estatísticas mais fáceis de se analisar ao contrário das séries originais.

Seguindo a metodologia de Taylor, foi adotado o logaritmo do retorno financeiro, o qual pode ser definido como visto na Eq. 5.1:

$$r_t = \log(p_t) - \log(p_{t-1}) \quad (5.1)$$

onde $p_t$ representa o preço no dia t.

Nesta etapa, considerando as fontes de dados mencionadas na introdução do capítulo 2, percebeu-se que os dados eram estruturados e com suficiente qualidade, apesar da existência de mais atributos de interesse que o desejado.

Diante disso foi feita a filtragem dos atributos de interesse - ligados ao conceito de preços e volume - e o tratamento de dados executado foi a filtragem dos ativos correspondentes a um dado tipo de mercado (mercado à vista e leilão).

No processo de normalização, foi usada exatamente a abordagem do logaritmo aplicado a todos os dados relacionados aos atributos relativos a preços por meio da função disponível na biblioteca Python *numpy*.

No restante do trabalho as séries na forma de logaritmo do retorno financeiro serão referenciadas apenas como séries log-retorno financeiro para simplificação.

## 5.4. Análise das propriedades das séries

Nesta etapa, são analisadas as propriedades estatísticas das séries log-retorno financeiro. São analisados os valores dos 4 primeiros momentos (média, desvio padrão, assimetria e curtoses) e valores extremos de forma a tentar compreender melhor a distribuição das séries utilizando-se uma amostra grande de dados com o intuito de se obter melhores estimativas sobre tais propriedades estatísticas.

## 5.5. Criação de um novo atributo a partir dos atributos originais

Nesta etapa, a partir dos métodos de rotulação, foi feita uma escolha para a criação de novo atributo a partir dos atributos originais que tenha potencial analítico considerando um modelo de aprendizado de máquina.

Nas abordagens práticas existem técnicas baseadas apenas em fluxo do ativo (chamadas de acumulação de volumes) com o objetivo de encontrar a direção/tendência do mercado. Neste sentido, uma base com a segregação dos volumes poderia também ser uma abordagem interessante a ser analisada/pesquisada.

Para a criação deste atributo indicador alvo (classe) será utilizado o seguinte método baseado no retorno financeiro, conforme descreve a equação 5.2:

$$Classe = \{0,\ se\ retorno\ financeiro\ \leq 0;\ \ 1,\ se\ retorno\ financeiro > \ 0\} \quad (5.2)$$

## 5.6. Utilização de modelo de previsão

Nesta etapa, o modelo de previsão apoiado em algoritmo de aprendizado de máquina supervisionado será utilizado. Serão utilizadas as etapas abaixo para a implementação do algoritmo:

- Modelagem em um problema de classificação binário: para cada dia futuro o algoritmo procura classificar aquele dia como uma classe de alta ou uma classe de baixa. A classe de alta corresponde a valores de log-retorno financeiro positivo e a classe de baixa a valores de log-retorno negativos. Essa classificação é realizada através do aprendizado obtido na etapa de treinamento por meio de classificações passadas;
- Definição do conjunto de treinamento/validação e conjunto de teste: se faz necessária a obtenção de um conjunto grande o suficiente para o treinamento do algoritmo; assim, conjuntos de validação e testes serão definidos para avaliar as métricas de classificação do modelo de predição;
- Configuração dos parâmetros do algoritmo: escolha dos parâmetros utilizados para o algoritmo de classificação.

## 5.7. Análise dos resultados

A análise dos resultados é comumente feita através de medidas de desempenho, adicionalmente, visto que em problemas de investimento é importante validar modelos do ponto de vista financeiro, pois estes nem sempre se correlacionam com a qualidade preditiva do modelo; utilizam-se também medidas financeiras.

### 5.7.1 Medidas de desempenho

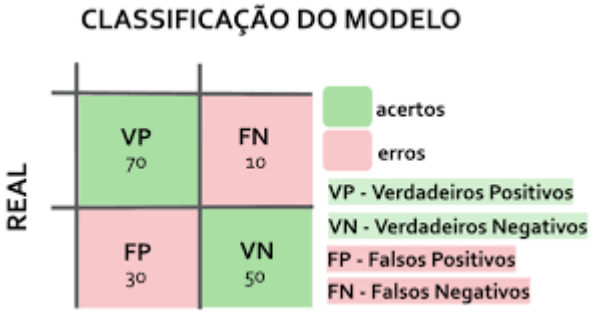
Essas medidas têm como objetivo realizar a avaliação da qualidade dos classificadores sendo constituídas de fórmulas matemáticas e estatísticas. Com a predição dos classificadores é possível extrair a matriz de confusão e assim gerar métricas de desempenho e risco.

A matriz de confusão é utilizada para organizar e exibir as informações utilizadas para avaliar o desempenho de um algoritmo.

Cada coluna da matriz representa as instâncias de uma classe prevista, enquanto cada linha representa uma classe real. Verdadeiros positivos (VP) são exemplos rotulados corretamente como positivos, falsos positivos (FP) são exemplos negativos incorretamente rotulados como positivos; verdadeiros negativos (VN) correspondem a negativos rotulados corretamente como negativos e falsos negativos (FN) referem-se a exemplos positivos incorretamente rotulados como negativos.

A *Figura 23 - Exemplo de Matriz de Confusão* ilustra a matriz de confusão para duas classes:

Figura 23 - Exemplo de Matriz de Confusão




As métricas mais utilizadas no processo de análise são detalhadas abaixo:

- **Acurácia**: é a quantidade de amostras positivas e negativas classificadas corretamente dividida pelo total de amostras da série avaliada em percentual, representada pela Eq. 5.3. No problema tratado, qual porcentagem de predições do algoritmo estava correta.

$$\frac{VP+VN}{VP+FP+VN+FN} \quad (5.3)$$

- **Revocação/sensibilidade**: é a quantidade de amostras positivas (VP) classificadas corretamente sobre o total de amostras classificadas como falsas negativas (FN) mais amostras positivas (VP) em percentual, representada pela Eq. 5.5. No problema tratado, dentre todos os casos de alta, qual porcentagem foi identificada.

$$\frac{VP}{VP+FN} \quad (5.5)$$

- **Precisão**: é a quantidade de amostras positivas classificadas corretamente sobre o total de amostras classificadas como falsas positivas acrescida das amostras positivas em percentual, representada pela Eq. 5.4. No problema tratado, dentre todos os casos identificados como sendo de alta, qual porcentagem realmente era de alta.

$$\frac{VP}{VP+FP} \quad (5.4)$$

- **Especificidade**: é quantidade de amostras negativas (VN) classificadas corretamente sobre total de amostras negativas (VN) mais total de amostras falsos positivos (FP) em percentual, representada pela Eq. 5.6. No problema tratado, dentre todos os casos de baixa, qual porcentagem foi identificada.

$$\frac{VN}{VN+FP} \quad (5.6)$$

### 5.7.2 Medidas financeiras

A métrica financeira mais importante é o retorno financeiro propriamente dito.

- **Retorno Financeiro**: é o percentual de ganho financeiro de um ativo durante um determinado período conforme apresentado na *Figura 23 - Exemplo de Matriz de Confusão*.

Outras métricas financeiras associadas a riscos também são utilizadas, mas não serão objeto de análise neste trabalho, por exemplo, volatilidade, índice Sharpe, entre outras.

# 6. Análise dos Resultados

Este capítulo tem o objetivo de apresentar os resultados considerando as etapas descritas na metodologia utilizada. Para atender às etapas descritas no capítulo 4, foi feito um módulo em Python contendo funções e estrutura de dados presentes no apêndice I. Foi escolhida uma lista de ativos para a avaliação do desempenho dos algoritmos: BBDC4, BOVA11, ITUB4, MEAL3, PETR4, SHOW3, VALE3.

Foram realizados experimentos considerando, os algoritmos *Random Forest* e LGBM considerando a divisão aleatória e a divisão temporal. Na divisão aleatória foi feita a estratificação das classes mantendo a proporção nos conjuntos de treino, validação e teste. Na divisão temporal foi utilizado o atributo de data para efetuar a devida divisão nos conjuntos de dados na proporção definida no capítulo de metodologia (oitenta/vinte). Os resultados de cada sequência de experimentos estão descritos nas seções 6.1 a 6.4, e na seção 6.5, os resultados são todos são comparados e discutidos.

## 6.1. Abordagem de particionamento divisão temporal – *Random Forest*

Neste cenário, foi feito o particionamento dos dados; foi utilizado o critério temporal, tendo modelo treinado para cada ativo. A *Tabela 1 - Random Forest - Treino e Validação (divisão temporal)* apresenta os valores para as medidas (acurácia, precisão, recall e especificidade) tanto para os conjuntos de treino quanto para o conjunto de validação.

Tabela 1 - Random Forest - Treino e Validação (divisão temporal)

| | Treino | | | | Validação | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Ativo** | **Acurácia** | **Precisão** | **Recall** | **Especificidade** | **Acurácia** | **Precisão** | **Recall** | **Especificidade** | **Acúmulo** |
| BBDC4 | 83,81% | 83,79% | 83,81% | 79,14% | 65,82% | 77,55% | 65,82% | 39,53% | 3 |
| BBDC4 | 93,59% | 93,74% | 93,59% | 95,71% | 62,50% | 74,42% | 62,50% | 39,13% | 6 |
| BBDC4 | 96,15% | 96,49% | 96,15% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 9 |
| BBDC4 | 100,00% | 100,00% | 100,00% | 100,00% | 70,00% | 82,86% | 70,00% | 50,00% | 12 |
| BBDC4 | 100,00% | 100,00% | 100,00% | 100,00% | 75,00% | 84,09% | 75,00% | 55,56% | 15 |
| BOVA11 | 87,94% | 87,94% | 87,94% | 85,04% | 83,54% | 84,05% | 83,54% | 83,33% | 3 |
| BOVA11 | 96,15% | 96,15% | 96,15% | 95,08% | 70,00% | 72,81% | 70,00% | 83,33% | 6 |
| BOVA11 | 98,08% | 98,17% | 98,08% | 100,00% | 73,08% | 83,55% | 73,08% | 100,00% | 9 |
| BOVA11 | 98,70% | 98,73% | 98,70% | 96,77% | 65,00% | 82,50% | 65,00% | 100,00% | 12 |
| BOVA11 | 96,72% | 96,72% | 96,72% | 95,45% | 93,75% | 94,79% | 93,75% | 100,00% | 15 |
| ITUB4 | 86,35% | 86,54% | 86,35% | 88,11% | 81,01% | 81,11% | 81,01% | 82,61% | 3 |
| ITUB4 | 96,79% | 96,81% | 96,79% | 97,14% | 82,50% | 83,00% | 82,50% | 90,91% | 6 |
| ITUB4 | 98,08% | 98,16% | 98,08% | 100,00% | 96,15% | 96,50% | 96,15% | 93,75% | 9 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 95,00% | 95,56% | 95,00% | 91,67% | 12 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 68,75% | 84,38% | 68,75% | 54,55% | 15 |
| MEAL3 | 83,81% | 83,95% | 83,81% | 81,48% | 70,89% | 71,95% | 70,89% | 58,97% | 3 |
| MEAL3 | 92,95% | 92,96% | 92,95% | 92,41% | 70,00% | 78,26% | 70,00% | 52,17% | 6 |
| MEAL3 | 96,15% | 96,22% | 96,15% | 98,04% | 65,38% | 73,99% | 65,38% | 46,67% | 9 |
| MEAL3 | 97,40% | 97,53% | 97,40% | 94,74% | 95,00% | 95,45% | 95,00% | 100,00% | 12 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 89,84% | 90,36% | 89,84% | 92,68% | 77,22% | 77,24% | 77,22% | 80,49% | 3 |
| PETR4 | 96,79% | 96,81% | 96,79% | 96,61% | 95,00% | 95,43% | 95,00% | 100,00% | 6 |
| PETR4 | 96,15% | 96,37% | 96,15% | 89,19% | 100,00% | 100,00% | 100,00% | 100,00% | 9 |
| PETR4 | 100,00% | 100,00% | 100,00% | 100,00% | 90,00% | 92,22% | 90,00% | 100,00% | 12 |
| PETR4 | 100,00% | 100,00% | 100,00% | 100,00% | 81,25% | 82,03% | 81,25% | 85,71% | 15 |
| SHOW3 | 81,59% | 81,61% | 81,59% | 82,25% | 58,23% | 71,11% | 58,23% | 29,55% | 3 |
| SHOW3 | 97,44% | 97,46% | 97,44% | 98,78% | 65,00% | 80,81% | 65,00% | 39,13% | 6 |
| SHOW3 | 98,08% | 98,15% | 98,08% | 100,00% | 57,69% | 77,08% | 57,69% | 15,38% | 9 |
| SHOW3 | 96,10% | 96,14% | 96,10% | 97,30% | 50,00% | 77,78% | 50,00% | 16,67% | 12 |
| SHOW3 | 100,00% | 100,00% | 100,00% | 100,00% | 62,50% | 78,57% | 62,50% | 25,00% | 15 |
| VALE3 | 89,84% | 89,86% | 89,84% | 89,51% | 74,68% | 82,97% | 74,68% | 96,67% | 3 |
| VALE3 | 97,44% | 97,44% | 97,44% | 96,92% | 92,50% | 92,47% | 92,50% | 84,62% | 6 |
| VALE3 | 97,12% | 97,12% | 97,12% | 95,45% | 69,23% | 78,85% | 69,23% | 87,50% | 9 |
| VALE3 | 96,10% | 96,17% | 96,10% | 96,30% | 70,00% | 86,36% | 70,00% | 100,00% | 12 |
| VALE3 | 98,36% | 98,40% | 98,36% | 96,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |

Fonte: Autor

Pela *Tabela 2 - Random Forest – treino e validação (acúmulo)* pode-se notar que o modelo apresenta *Overfitting* em boa parte do conjunto de dados de treinamento e em alguns casos no conjunto de dados de validação para o maior valor de acúmulo de dias utilizado. Os ativos BOVA11 e VALE3 não apresentaram este comportamento para o conjunto de treino, contudo no conjunto de validação isso não se verificou para o ativo VALE3.

Tabela 2 - Random Forest – treino e validação (acúmulo)

| | Treino | | | | Validação | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Ativo | Acurácia | Precisão | Recall | Especificidade | Acurácia | Precisão | Recall | Especificidade | Acúmulo |
| BBDC4 | 100,00% | 100,00% | 100,00% | 100,00% | 75,00% | 84,09% | 75,00% | 55,56% | 15 |
| BOVA11 | 96,72% | 96,72% | 96,72% | 95,45% | 93,75% | 94,79% | 93,75% | 100,00% | 15 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 68,75% | 84,38% | 68,75% | 54,55% | 15 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 100,00% | 100,00% | 100,00% | 100,00% | 81,25% | 82,03% | 81,25% | 85,71% | 15 |
| SHOW3 | 100,00% | 100,00% | 100,00% | 100,00% | 62,50% | 78,57% | 62,50% | 25,00% | 15 |
| VALE3 | 98,36% | 98,40% | 98,36% | 96,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |

Fonte: Autor

A *Tabela 3 - Random Forest – Teste (divisão temporal)* apresenta os dados para o conjunto de testes:

Tabela 3 - Random Forest – Teste (divisão temporal)

| | Testes | | | | |
|---|---|---|---|---|---|
| **Ativo** | **Acurácia** | **Precisão** | **Recall** | **Especificidade** | **Acúmulo** |
| BBDC4 | 57,58% | 68,83% | 57,58% | 20,00% | 3 |
| BBDC4 | 46,00% | 21,16% | 46,00% | 0,00% | 6 |
| BBDC4 | 90,91% | 91,08% | 90,91% | 88,24% | 9 |
| BBDC4 | 76,00% | 84,47% | 76,00% | 57,14% | 12 |
| BBDC4 | 55,00% | 77,50% | 55,00% | 18,18% | 15 |
| BOVA11 | 84,85% | 85,02% | 84,85% | 82,69% | 3 |
| BOVA11 | 56,00% | 76,60% | 56,00% | 100,00% | 6 |
| BOVA11 | 66,67% | 79,76% | 66,67% | 100,00% | 9 |
| BOVA11 | 52,00% | 78,18% | 52,00% | 100,00% | 12 |
| BOVA11 | 95,00% | 95,50% | 95,00% | 90,91% | 15 |
| ITUB4 | 82,83% | 83,29% | 82,83% | 79,25% | 3 |
| ITUB4 | 90,00% | 90,06% | 90,00% | 88,00% | 6 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 9 |
| ITUB4 | 84,00% | 88,27% | 84,00% | 71,43% | 12 |
| ITUB4 | 85,00% | 88,46% | 85,00% | 70,00% | 15 |
| MEAL3 | 60,61% | 75,73% | 60,61% | 29,63% | 3 |
| MEAL3 | 52,00% | 77,60% | 52,00% | 17,24% | 6 |
| MEAL3 | 57,58% | 77,37% | 57,58% | 17,65% | 9 |
| MEAL3 | 96,00% | 96,25% | 96,00% | 100,00% | 12 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 79,80% | 81,64% | 79,80% | 88,10% | 3 |
| PETR4 | 96,00% | 96,26% | 96,00% | 90,48% | 6 |
| PETR4 | 87,88% | 87,88% | 87,88% | 84,62% | 9 |
| PETR4 | 96,00% | 96,40% | 96,00% | 100,00% | 12 |
| PETR4 | 95,00% | 95,50% | 95,00% | 100,00% | 15 |
| SHOW3 | 57,58% | 65,67% | 57,58% | 20,41% | 3 |
| SHOW3 | 68,00% | 71,60% | 68,00% | 36,36% | 6 |
| SHOW3 | 54,55% | 77,27% | 54,55% | 16,67% | 9 |
| SHOW3 | 48,00% | 77,39% | 48,00% | 13,33% | 12 |
| SHOW3 | 60,00% | 77,78% | 60,00% | 20,00% | 15 |
| VALE3 | 46,46% | 21,59% | 46,46% | 100,00% | 3 |
| VALE3 | 94,00% | 94,12% | 94,00% | 95,00% | 6 |
| VALE3 | 42,42% | 18,00% | 42,42% | 100,00% | 9 |
| VALE3 | 36,00% | 12,96% | 36,00% | 100,00% | 12 |
| VALE3 | 95,00% | 95,42% | 95,00% | 88,89% | 15 |

Fonte: Autor

Pela tabela Tabela *4* - Random Forest – Teste (acúmulo), o mesmo comportamento citado para os conjuntos de dados de treino e validação se observou com o conjunto de dados de teste ao se aumentar o valor de acúmulo de dias contudo, isso se verificou apenas para o ativo MEAL3.

Tabela 4 - Random Forest – Teste (acúmulo)

| | Testes | | | | |
|---|---|---|---|---|---|
| Ativo | Acurácia | Precisão | Recall | Especificidade | Acúmulo |
| BBDC4 | 55,00% | 77,50% | 55,00% | 18,18% | 15 |
| BOVA11 | 95,00% | 95,50% | 95,00% | 90,91% | 15 |
| ITUB4 | 85,00% | 88,46% | 85,00% | 70,00% | 15 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 95,00% | 95,50% | 95,00% | 100,00% | 15 |
| SHOW3 | 60,00% | 77,78% | 60,00% | 20,00% | 15 |
| VALE3 | 95,00% | 95,42% | 95,00% | 88,89% | 15 |

Fonte: Autor

## 6.2. Abordagem de particionamento divisão aleatória – *Random Forest*

Neste cenário foi feito o particionamento dos dados de forma aleatória; em seguida, foi treinado o modelo para cada ativo. A *Tabela 5 - Random Forest - Treino e Validação (divisão aleatória)* apresenta os valores para as medidas (acurácia, precisão e recall e especificidade) tanto para os dados de treino quanto para o conjunto de validação.

Tabela 5 - Random Forest - Treino e Validação (divisão aleatória)

| | Treino | | | | Validação | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Ativo** | **Acurácia** | **Precisão** | **Recall** | **Especificidade** | **Acurácia** | **Precisão** | **Recall** | **Especificidade** | **Acúmulo** |
| BBDC4 | 84,44% | 84,66% | 84,44% | 87,42% | 81,01% | 81,10% | 81,01% | 79,41% | 3 |
| BBDC4 | 94,23% | 94,24% | 94,23% | 94,52% | 82,50% | 83,63% | 82,50% | 78,26% | 6 |
| BBDC4 | 96,15% | 96,24% | 96,15% | 97,83% | 92,31% | 92,31% | 92,31% | 92,86% | 9 |
| BBDC4 | 100,00% | 100,00% | 100,00% | 100,00% | 95,00% | 95,50% | 95,00% | 100,00% | 12 |
| BBDC4 | 100,00% | 100,00% | 100,00% | 100,00% | 87,50% | 87,50% | 87,50% | 87,50% | 15 |
| BOVA11 | 87,30% | 87,29% | 87,30% | 84,29% | 82,28% | 83,92% | 82,28% | 85,19% | 3 |
| BOVA11 | 96,15% | 96,17% | 96,15% | 93,65% | 87,50% | 87,59% | 87,50% | 90,00% | 6 |
| BOVA11 | 98,08% | 98,08% | 98,08% | 97,44% | 88,46% | 88,76% | 88,46% | 85,71% | 9 |
| BOVA11 | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 12 |
| BOVA11 | 100,00% | 100,00% | 100,00% | 100,00% | 68,75% | 67,95% | 68,75% | 50,00% | 15 |
| ITUB4 | 83,49% | 83,49% | 83,49% | 82,31% | 84,81% | 85,03% | 84,81% | 84,78% | 3 |
| ITUB4 | 96,79% | 96,80% | 96,79% | 95,89% | 82,50% | 82,58% | 82,50% | 85,00% | 6 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 9 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 95,00% | 95,50% | 95,00% | 90,91% | 12 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 93,75% | 94,64% | 93,75% | 100,00% | 15 |
| MEAL3 | 82,86% | 82,86% | 82,86% | 82,28% | 78,48% | 78,59% | 78,48% | 80,43% | 3 |
| MEAL3 | 91,03% | 91,03% | 91,03% | 91,57% | 85,00% | 85,28% | 85,00% | 90,48% | 6 |
| MEAL3 | 98,08% | 98,08% | 98,08% | 98,04% | 92,31% | 93,21% | 92,31% | 100,00% | 9 |
| MEAL3 | 98,70% | 98,73% | 98,70% | 100,00% | 80,00% | 81,62% | 80,00% | 88,89% | 12 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 89,52% | 89,82% | 89,52% | 91,85% | 75,95% | 76,12% | 75,95% | 70,00% | 3 |
| PETR4 | 96,15% | 96,20% | 96,15% | 97,01% | 95,00% | 95,00% | 95,00% | 92,31% | 6 |
| PETR4 | 98,08% | 98,13% | 98,08% | 94,44% | 84,62% | 85,62% | 84,62% | 78,57% | 9 |
| PETR4 | 97,40% | 97,58% | 97,40% | 100,00% | 85,00% | 89,50% | 85,00% | 100,00% | 12 |
| PETR4 | 98,36% | 98,41% | 98,36% | 96,15% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| SHOW3 | 82,54% | 82,53% | 82,54% | 86,05% | 77,22% | 80,35% | 77,22% | 91,89% | 3 |
| SHOW3 | 94,23% | 94,24% | 94,23% | 95,06% | 90,00% | 91,67% | 90,00% | 100,00% | 6 |
| SHOW3 | 98,08% | 98,08% | 98,08% | 98,11% | 76,92% | 76,92% | 76,92% | 80,00% | 9 |
| SHOW3 | 97,40% | 97,40% | 97,40% | 97,50% | 90,00% | 91,54% | 90,00% | 100,00% | 12 |
| SHOW3 | 100,00% | 100,00% | 100,00% | 100,00% | 87,50% | 90,00% | 87,50% | 100,00% | 15 |
| VALE3 | 87,30% | 87,39% | 87,30% | 88,44% | 84,81% | 86,31% | 84,81% | 89,29% | 3 |
| VALE3 | 96,79% | 96,81% | 96,79% | 96,61% | 85,00% | 85,00% | 85,00% | 84,21% | 6 |
| VALE3 | 95,19% | 95,54% | 95,19% | 87,18% | 92,31% | 93,41% | 92,31% | 85,71% | 9 |
| VALE3 | 98,70% | 98,73% | 98,70% | 96,00% | 90,00% | 91,43% | 90,00% | 75,00% | 12 |
| VALE3 | 100,00% | 100,00% | 100,00% | 100,00% | 68,75% | 75,78% | 68,75% | 80,00% | 15 |

Fonte: Autor

Pela tabela Tabela *6* - Random Forest - treino e validação (acúmulo), observou-se também o mesmo problema de *Overfitting* ao se aumentar o valor do parâmetro de acúmulo de dias como já comentado utilizando a técnica de divisão temporal e apenas o ativo PETR4 não apresentou este comportamento para o conjunto de treino, contudo no conjunto de validação isso não se verificou.

Tabela 6 - Random Forest - treino e validação (acúmulo)

| | Treino | | | | Validação | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Ativo | Acurácia | Precisão | Recall | Especificidade | Acurácia | Precisão | Recall | Especificidade | Acúmulo |
| BBDC4 | 100,00% | 100,00% | 100,00% | 100,00% | 87,50% | 87,50% | 87,50% | 87,50% | 15 |
| BOVA11 | 100,00% | 100,00% | 100,00% | 100,00% | 68,75% | 67,95% | 68,75% | 50,00% | 15 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 93,75% | 94,64% | 93,75% | 100,00% | 15 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 98,36% | 98,41% | 98,36% | 96,15% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| SHOW3 | 100,00% | 100,00% | 100,00% | 100,00% | 87,50% | 90,00% | 87,50% | 100,00% | 15 |
| VALE3 | 100,00% | 100,00% | 100,00% | 100,00% | 68,75% | 75,78% | 68,75% | 80,00% | 15 |

Fonte: Autor

A Tabela *7* - Random Forest - Teste (divisão aleatória) apresenta os valores para as medidas (acurácia, precisão e recall e especificidade) para os dados de teste.

Tabela 7 - Random Forest - Teste (divisão aleatória)

| | Testes | | | | |
|---|---|---|---|---|---|
| **Ativo** | **Acurácia** | **Precisão** | **Recall** | **Especificidade** | **Acúmulo** |
| BBDC4 | 87,88% | 88,38% | 87,88% | 80,85% | 3 |
| BBDC4 | 96,00% | 96,00% | 96,00% | 95,83% | 6 |
| BBDC4 | 93,94% | 93,94% | 93,94% | 93,33% | 9 |
| BBDC4 | 96,00% | 96,29% | 96,00% | 91,67% | 12 |
| BBDC4 | 75,00% | 76,79% | 75,00% | 55,56% | 15 |
| BOVA11 | 88,89% | 89,46% | 88,89% | 92,86% | 3 |
| BOVA11 | 90,00% | 90,00% | 90,00% | 85,71% | 6 |
| BOVA11 | 87,88% | 88,15% | 87,88% | 76,92% | 9 |
| BOVA11 | 88,00% | 88,44% | 88,00% | 90,00% | 12 |
| BOVA11 | 85,00% | 85,05% | 85,00% | 75,00% | 15 |
| ITUB4 | 87,88% | 87,88% | 87,88% | 87,76% | 3 |
| ITUB4 | 86,00% | 86,46% | 86,00% | 79,17% | 6 |
| ITUB4 | 96,97% | 97,15% | 96,97% | 100,00% | 9 |
| ITUB4 | 96,00% | 96,31% | 96,00% | 92,31% | 12 |
| ITUB4 | 80,00% | 80,88% | 80,00% | 90,91% | 15 |
| MEAL3 | 78,79% | 78,86% | 78,79% | 82,35% | 3 |
| MEAL3 | 82,00% | 83,59% | 82,00% | 74,07% | 6 |
| MEAL3 | 84,85% | 85,97% | 84,85% | 94,12% | 9 |
| MEAL3 | 92,00% | 92,00% | 92,00% | 92,31% | 12 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 84,85% | 84,92% | 84,85% | 82,93% | 3 |
| PETR4 | 86,00% | 88,72% | 86,00% | 66,67% | 6 |
| PETR4 | 84,85% | 84,78% | 84,85% | 76,92% | 9 |
| PETR4 | 76,00% | 80,82% | 76,00% | 88,89% | 12 |
| PETR4 | 95,00% | 95,38% | 95,00% | 87,50% | 15 |
| SHOW3 | 74,75% | 76,14% | 74,75% | 88,68% | 3 |
| SHOW3 | 94,00% | 94,06% | 94,00% | 96,15% | 6 |
| SHOW3 | 90,91% | 92,27% | 90,91% | 100,00% | 9 |
| SHOW3 | 88,00% | 88,21% | 88,00% | 92,31% | 12 |
| SHOW3 | 95,00% | 95,45% | 95,00% | 100,00% | 15 |
| VALE3 | 87,88% | 88,02% | 87,88% | 81,82% | 3 |
| VALE3 | 92,00% | 92,00% | 92,00% | 90,00% | 6 |
| VALE3 | 87,88% | 89,90% | 87,88% | 69,23% | 9 |
| VALE3 | 76,00% | 75,81% | 76,00% | 37,50% | 12 |
| VALE3 | 85,00% | 85,05% | 85,00% | 75,00% | 15 |

Fonte: Autor

Pela tabela Tabela *8* - Random Forest - Teste (acúmulo), observou-se mesmo comportamento citado para os conjuntos de dados de treino e validação se observou com o conjunto de dados de teste ao se aumentar o valor de acúmulo de dias contudo, isso se verificou apenas para o ativo MEAL3.

Tabela 8 - Random Forest - Teste (acúmulo)

| | Testes | | | | |
|---|---|---|---|---|---|
| Ativo | Acurácia | Precisão | Recall | Especificidade | Acúmulo |
| BBDC4 | 75,00% | 76,79% | 75,00% | 55,56% | 15 |
| BOVA11 | 85,00% | 85,05% | 85,00% | 75,00% | 15 |
| ITUB4 | 80,00% | 80,88% | 80,00% | 90,91% | 15 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 95,00% | 95,38% | 95,00% | 87,50% | 15 |
| SHOW3 | 95,00% | 95,45% | 95,00% | 100,00% | 15 |
| VALE3 | 85,00% | 85,05% | 85,00% | 75,00% | 15 |

Fonte: Autor

## 6.3. Abordagem de particionamento divisão temporal – LGBM

Neste cenário foi feito o particionamento dos dados utilizando o critério temporal e foi treinado o modelo para cada ativo. A Tabela *9* - LGBM - Treino e Validação (divisão temporal) apresenta os valores para as medidas (acurácia, precisão, recall e especificidade) tanto para os dados de treino quanto para o conjunto de validação.

Tabela 9 - LGBM - Treino e Validação (divisão temporal)

| | Treino | | | | Validação | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Ativo** | **Acurácia** | **Precisão** | **Recall** | **Especificidade** | **Acurácia** | **Precisão** | **Recall** | **Especificidade** | **Acúmulo** |
| BBDC4 | 96,19% | 96,19% | 96,19% | 95,68% | 64,56% | 74,52% | 64,56% | 39,53% | 3 |
| BBDC4 | 96,15% | 96,46% | 96,15% | 100,00% | 90,00% | 90,00% | 90,00% | 91,30% | 6 |
| BBDC4 | 95,19% | 95,70% | 95,19% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 9 |
| BBDC4 | 100,00% | 100,00% | 100,00% | 100,00% | 70,00% | 82,86% | 70,00% | 50,00% | 12 |
| BBDC4 | 96,72% | 96,90% | 96,72% | 92,31% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| BOVA11 | 97,78% | 97,80% | 97,78% | 98,43% | 83,54% | 84,05% | 83,54% | 83,33% | 3 |
| BOVA11 | 99,36% | 99,37% | 99,36% | 98,36% | 85,00% | 85,21% | 85,00% | 77,78% | 6 |
| BOVA11 | 97,12% | 97,33% | 97,12% | 100,00% | 96,15% | 96,47% | 96,15% | 100,00% | 9 |
| BOVA11 | 96,10% | 96,16% | 96,10% | 96,77% | 95,00% | 95,36% | 95,00% | 85,71% | 12 |
| BOVA11 | 91,80% | 91,93% | 91,80% | 90,91% | 93,75% | 94,79% | 93,75% | 100,00% | 15 |
| ITUB4 | 95,24% | 95,24% | 95,24% | 94,41% | 81,01% | 81,11% | 81,01% | 82,61% | 3 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 82,50% | 83,00% | 82,50% | 90,91% | 6 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 96,15% | 96,50% | 96,15% | 93,75% | 9 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 12 |
| ITUB4 | 96,72% | 96,72% | 96,72% | 96,67% | 93,75% | 94,79% | 93,75% | 90,91% | 15 |
| MEAL3 | 95,56% | 95,58% | 95,56% | 96,91% | 73,42% | 73,58% | 73,42% | 76,92% | 3 |
| MEAL3 | 94,23% | 94,30% | 94,23% | 92,41% | 90,00% | 90,00% | 90,00% | 91,30% | 6 |
| MEAL3 | 96,15% | 96,22% | 96,15% | 98,04% | 96,15% | 96,39% | 96,15% | 100,00% | 9 |
| MEAL3 | 94,81% | 94,81% | 94,81% | 94,74% | 95,00% | 95,45% | 95,00% | 100,00% | 12 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 98,10% | 98,18% | 98,10% | 100,00% | 79,75% | 79,75% | 79,75% | 80,49% | 3 |
| PETR4 | 98,72% | 98,76% | 98,72% | 100,00% | 87,50% | 88,45% | 87,50% | 80,95% | 6 |
| PETR4 | 95,19% | 95,27% | 95,19% | 89,19% | 100,00% | 100,00% | 100,00% | 100,00% | 9 |
| PETR4 | 100,00% | 100,00% | 100,00% | 100,00% | 90,00% | 92,22% | 90,00% | 100,00% | 12 |
| PETR4 | 98,36% | 98,40% | 98,36% | 95,65% | 87,50% | 87,50% | 87,50% | 85,71% | 15 |
| SHOW3 | 94,60% | 94,63% | 94,60% | 94,08% | 58,23% | 71,11% | 58,23% | 29,55% | 3 |
| SHOW3 | 100,00% | 100,00% | 100,00% | 100,00% | 82,50% | 82,71% | 82,50% | 82,61% | 6 |
| SHOW3 | 100,00% | 100,00% | 100,00% | 100,00% | 88,46% | 88,69% | 88,46% | 84,62% | 9 |
| SHOW3 | 94,81% | 94,94% | 94,81% | 97,30% | 90,00% | 90,00% | 90,00% | 91,67% | 12 |
| SHOW3 | 98,36% | 98,41% | 98,36% | 100,00% | 93,75% | 94,44% | 93,75% | 100,00% | 15 |
| VALE3 | 98,10% | 98,10% | 98,10% | 97,90% | 81,01% | 83,66% | 81,01% | 90,00% | 3 |
| VALE3 | 98,72% | 98,76% | 98,72% | 100,00% | 92,50% | 92,76% | 92,50% | 92,31% | 6 |
| VALE3 | 100,00% | 100,00% | 100,00% | 100,00% | 96,15% | 96,36% | 96,15% | 87,50% | 9 |
| VALE3 | 97,40% | 97,58% | 97,40% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 12 |
| VALE3 | 98,36% | 98,40% | 98,36% | 96,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |

Fonte: Autor

Pela Tabela *10* - LGBM - Treino e Validação (acúmulo), pode-se notar que ao aumentar o número de acúmulo de dias, o modelo acaba por apresentar *Overfitting* no conjunto de dados de treinamento do ativo MEAL3 e nos conjuntos de dados de validação dos ativos BBDC4, MEAL3 e VALE3.

Tabela 10 - LGBM - Treino e Validação (acúmulo)

| | Treino | | | | Validação | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Ativo | Acurácia | Precisão | Recall | Especificidade | Acurácia | Precisão | Recall | Especificidade | Acúmulo |
| BBDC4 | 96,72% | 96,90% | 96,72% | 92,31% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| BOVA11 | 91,80% | 91,93% | 91,80% | 90,91% | 93,75% | 94,79% | 93,75% | 100,00% | 15 |
| ITUB4 | 96,72% | 96,72% | 96,72% | 96,67% | 93,75% | 94,79% | 93,75% | 90,91% | 15 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 98,36% | 98,40% | 98,36% | 95,65% | 87,50% | 87,50% | 87,50% | 85,71% | 15 |
| SHOW3 | 98,36% | 98,41% | 98,36% | 100,00% | 93,75% | 94,44% | 93,75% | 100,00% | 15 |
| VALE3 | 98,36% | 98,40% | 98,36% | 96,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |

Fonte: Autor

A Tabela *11* - LGBM - Teste (divisão temporal) apresenta os valores para as medidas (acurácia, precisão e recall e especificidade) para os dados de teste.

Tabela 11 - LGBM - Teste *(divisão temporal)*

| | Testes | | | | |
|---|---|---|---|---|---|
| **Ativo** | **Acurácia** | **Precisão** | **Recall** | **Especificidade** | **Acúmulo** |
| BBDC4 | 55,56% | 70,30% | 55,56% | 14,00% | 3 |
| BBDC4 | 92,00% | 92,00% | 92,00% | 92,59% | 6 |
| BBDC4 | 90,91% | 91,08% | 90,91% | 88,24% | 9 |
| BBDC4 | 48,00% | 76,17% | 48,00% | 7,14% | 12 |
| BBDC4 | 90,00% | 91,82% | 90,00% | 81,82% | 15 |
| BOVA11 | 84,85% | 85,02% | 84,85% | 82,69% | 3 |
| BOVA11 | 92,00% | 92,27% | 92,00% | 96,00% | 6 |
| BOVA11 | 84,85% | 85,00% | 84,85% | 82,35% | 9 |
| BOVA11 | 92,00% | 93,33% | 92,00% | 100,00% | 12 |
| BOVA11 | 95,00% | 95,50% | 95,00% | 90,91% | 15 |
| ITUB4 | 82,83% | 83,29% | 82,83% | 79,25% | 3 |
| ITUB4 | 90,00% | 90,06% | 90,00% | 88,00% | 6 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 9 |
| ITUB4 | 96,00% | 96,33% | 96,00% | 92,86% | 12 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| MEAL3 | 79,80% | 79,92% | 79,80% | 79,63% | 3 |
| MEAL3 | 82,00% | 83,86% | 82,00% | 75,86% | 6 |
| MEAL3 | 87,88% | 88,40% | 87,88% | 94,12% | 9 |
| MEAL3 | 96,00% | 96,25% | 96,00% | 100,00% | 12 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 71,72% | 75,48% | 71,72% | 85,71% | 3 |
| PETR4 | 96,00% | 96,26% | 96,00% | 90,48% | 6 |
| PETR4 | 90,91% | 90,95% | 90,91% | 84,62% | 9 |
| PETR4 | 96,00% | 96,40% | 96,00% | 100,00% | 12 |
| PETR4 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| SHOW3 | 56,57% | 64,41% | 56,57% | 18,37% | 3 |
| SHOW3 | 90,00% | 90,72% | 90,00% | 95,45% | 6 |
| SHOW3 | 84,85% | 85,06% | 84,85% | 83,33% | 9 |
| SHOW3 | 84,00% | 84,00% | 84,00% | 86,67% | 12 |
| SHOW3 | 95,00% | 95,45% | 95,00% | 100,00% | 15 |
| VALE3 | 74,75% | 77,01% | 74,75% | 86,96% | 3 |
| VALE3 | 94,00% | 94,12% | 94,00% | 95,00% | 6 |
| VALE3 | 87,88% | 88,24% | 87,88% | 78,57% | 9 |
| VALE3 | 100,00% | 100,00% | 100,00% | 100,00% | 12 |
| VALE3 | 95,00% | 95,42% | 95,00% | 88,89% | 15 |

Fonte: Autor

Pela Tabela *12*- LGBM - Testes (acúmulo), pode-se notar que ao aumentar o número de acúmulo de dias, o modelo acaba por apresentar *Overfitting* no conjunto de dados de teste dos ativos ITUB4, MEAL3 e PETR4.

Tabela 12- LGBM - Testes (acúmulo)

| | Testes | | | | |
|---|---|---|---|---|---|
| Ativo | Acurácia | Precisão | Recall | Especificidade | Acúmulo |
| BBDC4 | 90,00% | 91,82% | 90,00% | 81,82% | 15 |
| BOVA11 | 95,00% | 95,50% | 95,00% | 90,91% | 15 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| SHOW3 | 95,00% | 95,45% | 95,00% | 100,00% | 15 |
| VALE3 | 95,00% | 95,42% | 95,00% | 88,89% | 15 |

Fonte: Autor

## 6.4. Abordagem de particionamento divisão aleatória – LGBM

Tabela 13 - LGBM - Treino e Validação (divisão aleatória)

| | Treino | | | | Validação | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Ativo | Acurácia | Precisão | Recall | Especificidade | Acurácia | Precisão | Recall | Especificidade | Acúmulo |
| BBDC4 | 92,70% | 92,75% | 92,70% | 94,04% | 78,48% | 78,58% | 78,48% | 76,47% | 3 |
| BBDC4 | 97,44% | 97,57% | 97,44% | 100,00% | 85,00% | 85,58% | 85,00% | 82,61% | 6 |
| BBDC4 | 96,15% | 96,24% | 96,15% | 97,83% | 92,31% | 92,31% | 92,31% | 92,86% | 9 |
| BBDC4 | 100,00% | 100,00% | 100,00% | 100,00% | 95,00% | 95,50% | 95,00% | 100,00% | 12 |
| BBDC4 | 95,08% | 95,12% | 95,08% | 93,10% | 93,75% | 94,44% | 93,75% | 100,00% | 15 |
| BOVA11 | 98,73% | 98,77% | 98,73% | 100,00% | 81,01% | 81,69% | 81,01% | 77,78% | 3 |
| BOVA11 | 96,79% | 96,79% | 96,79% | 95,24% | 92,50% | 92,61% | 92,50% | 95,00% | 6 |
| BOVA11 | 96,15% | 96,26% | 96,15% | 97,44% | 88,46% | 88,62% | 88,46% | 92,86% | 9 |
| BOVA11 | 96,10% | 96,47% | 96,10% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 12 |
| BOVA11 | 91,80% | 91,91% | 91,80% | 91,67% | 87,50% | 87,50% | 87,50% | 83,33% | 15 |
| ITUB4 | 93,65% | 93,66% | 93,65% | 93,88% | 84,81% | 85,42% | 84,81% | 82,61% | 3 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 80,00% | 80,00% | 80,00% | 80,00% | 6 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 92,31% | 93,49% | 92,31% | 86,67% | 9 |
| ITUB4 | 97,40% | 97,54% | 97,40% | 95,00% | 100,00% | 100,00% | 100,00% | 100,00% | 12 |
| ITUB4 | 96,72% | 96,72% | 96,72% | 97,06% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| MEAL3 | 93,33% | 93,33% | 93,33% | 93,67% | 79,75% | 80,00% | 79,75% | 80,43% | 3 |
| MEAL3 | 96,79% | 96,80% | 96,79% | 97,59% | 85,00% | 85,28% | 85,00% | 90,48% | 6 |
| MEAL3 | 98,08% | 98,08% | 98,08% | 98,04% | 88,46% | 88,83% | 88,46% | 86,67% | 9 |
| MEAL3 | 98,70% | 98,74% | 98,70% | 97,56% | 85,00% | 85,50% | 85,00% | 88,89% | 12 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 98,10% | 98,18% | 98,10% | 100,00% | 74,68% | 74,42% | 74,68% | 63,33% | 3 |
| PETR4 | 99,36% | 99,37% | 99,36% | 100,00% | 95,00% | 95,67% | 95,00% | 100,00% | 6 |
| PETR4 | 98,08% | 98,08% | 98,08% | 97,22% | 88,46% | 90,77% | 88,46% | 78,57% | 9 |
| PETR4 | 98,70% | 98,75% | 98,70% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 12 |
| PETR4 | 96,72% | 96,72% | 96,72% | 96,15% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| SHOW3 | 92,38% | 92,39% | 92,38% | 94,19% | 77,22% | 82,93% | 77,22% | 97,30% | 3 |
| SHOW3 | 98,08% | 98,08% | 98,08% | 98,77% | 95,00% | 95,45% | 95,00% | 100,00% | 6 |
| SHOW3 | 99,04% | 99,06% | 99,04% | 98,11% | 76,92% | 76,92% | 76,92% | 80,00% | 9 |
| SHOW3 | 97,40% | 97,40% | 97,40% | 97,50% | 90,00% | 91,54% | 90,00% | 100,00% | 12 |
| SHOW3 | 100,00% | 100,00% | 100,00% | 100,00% | 87,50% | 90,00% | 87,50% | 100,00% | 15 |
| VALE3 | 94,60% | 94,61% | 94,60% | 94,56% | 79,75% | 83,15% | 79,75% | 89,29% | 3 |
| VALE3 | 98,08% | 98,17% | 98,08% | 100,00% | 92,50% | 92,59% | 92,50% | 89,47% | 6 |
| VALE3 | 94,23% | 94,24% | 94,23% | 89,74% | 96,15% | 96,45% | 96,15% | 92,86% | 9 |
| VALE3 | 100,00% | 100,00% | 100,00% | 100,00% | 95,00% | 95,56% | 95,00% | 100,00% | 12 |
| VALE3 | 98,36% | 98,41% | 98,36% | 96,15% | 93,75% | 94,27% | 93,75% | 80,00% | 15 |

Fonte: Autor

Pela Tabela *13* - LGBM - Treino e Validação (divisão aleatória) pode-se notar que ao aumentar o número de acúmulo de dias, o modelo acaba por apresentar *Overfitting* no conjunto de dados de treinamento dos ativos MEAL3 e SHOW3 e nos conjuntos de dados de validação dos ativos ITUB4, MEAL3 E PETR4.

Tabela 14- LGBM - Treino e Validação (acúmulo)

| | Treino | | | | Validação | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Ativo | Acurácia | Precisão | Recall | Especificidade | Acurácia | Precisão | Recall | Especificidade | Acúmulo |
| BBDC4 | 95,08% | 95,12% | 95,08% | 93,10% | 93,75% | 94,44% | 93,75% | 100,00% | 15 |
| BOVA11 | 91,80% | 91,91% | 91,80% | 91,67% | 87,50% | 87,50% | 87,50% | 83,33% | 15 |
| ITUB4 | 96,72% | 96,72% | 96,72% | 97,06% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 96,72% | 96,72% | 96,72% | 96,15% | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| SHOW3 | 100,00% | 100,00% | 100,00% | 100,00% | 87,50% | 90,00% | 87,50% | 100,00% | 15 |
| VALE3 | 98,36% | 98,41% | 98,36% | 96,15% | 93,75% | 94,27% | 93,75% | 80,00% | 15 |

Fonte: Autor

A Tabela *15* - LGBM - Testes (divisão aleatória) apresenta os valores para as medidas (acurácia, precisão e recall e especificidade) para os dados de teste.

Tabela 15 - LGBM - Testes (divisão aleatória)

| | Testes | | | | |
|---|---|---|---|---|---|
| **Ativo** | **Acurácia** | **Precisão** | **Recall** | **Especificidade** | **Acúmulo** |
| BBDC4 | 82,83% | 83,08% | 82,83% | 76,60% | 3 |
| BBDC4 | 92,00% | 92,00% | 92,00% | 91,67% | 6 |
| BBDC4 | 93,94% | 93,94% | 93,94% | 93,33% | 9 |
| BBDC4 | 100,00% | 100,00% | 100,00% | 100,00% | 12 |
| BBDC4 | 90,00% | 90,00% | 90,00% | 88,89% | 15 |
| BOVA11 | 81,82% | 83,09% | 81,82% | 88,10% | 3 |
| BOVA11 | 92,00% | 92,00% | 92,00% | 90,48% | 6 |
| BOVA11 | 96,97% | 97,11% | 96,97% | 92,31% | 9 |
| BOVA11 | 88,00% | 88,44% | 88,00% | 90,00% | 12 |
| BOVA11 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| ITUB4 | 85,86% | 85,92% | 85,86% | 87,76% | 3 |
| ITUB4 | 90,00% | 90,05% | 90,00% | 87,50% | 6 |
| ITUB4 | 96,97% | 97,15% | 96,97% | 100,00% | 9 |
| ITUB4 | 100,00% | 100,00% | 100,00% | 100,00% | 12 |
| ITUB4 | 95,00% | 95,50% | 95,00% | 90,91% | 15 |
| MEAL3 | 81,82% | 81,85% | 81,82% | 84,31% | 3 |
| MEAL3 | 84,00% | 85,08% | 84,00% | 77,78% | 6 |
| MEAL3 | 81,82% | 82,23% | 81,82% | 88,24% | 9 |
| MEAL3 | 92,00% | 92,00% | 92,00% | 92,31% | 12 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 79,80% | 79,82% | 79,80% | 68,29% | 3 |
| PETR4 | 90,00% | 90,42% | 90,00% | 80,95% | 6 |
| PETR4 | 93,94% | 94,49% | 93,94% | 84,62% | 9 |
| PETR4 | 96,00% | 96,24% | 96,00% | 88,89% | 12 |
| PETR4 | 95,00% | 95,38% | 95,00% | 87,50% | 15 |
| SHOW3 | 69,70% | 70,81% | 69,70% | 84,91% | 3 |
| SHOW3 | 92,00% | 92,25% | 92,00% | 96,15% | 6 |
| SHOW3 | 96,97% | 97,14% | 96,97% | 100,00% | 9 |
| SHOW3 | 88,00% | 88,21% | 88,00% | 92,31% | 12 |
| SHOW3 | 95,00% | 95,45% | 95,00% | 100,00% | 15 |
| VALE3 | 84,85% | 84,83% | 84,85% | 81,82% | 3 |
| VALE3 | 98,00% | 98,10% | 98,00% | 100,00% | 6 |
| VALE3 | 93,94% | 93,94% | 93,94% | 92,31% | 9 |
| VALE3 | 88,00% | 87,87% | 88,00% | 75,00% | 12 |
| VALE3 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |

Fonte: Autor

Pela Tabela *16*- LGBM - testes (acúmulo) pode-se notar que ao aumentar o número de acúmulo de dias, o modelo acaba por apresentar *Overfitting* no conjunto de dados de testes dos ativos BOVA11, MEAL3 e VALE3.

Tabela 16- LGBM - testes (acúmulo)

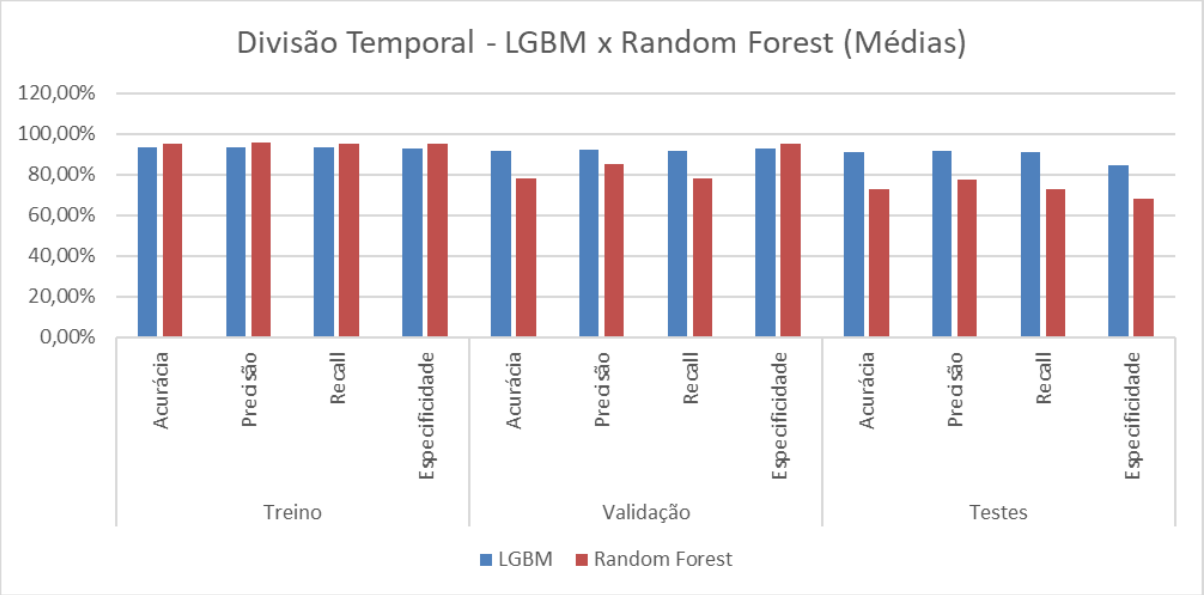
| | Testes | | | | |
|---|---|---|---|---|---|
| Ativo | Acurácia | Precisão | Recall | Especificidade | Acúmulo |
| BBDC4 | 90,00% | 90,00% | 90,00% | 88,89% | 15 |
| BOVA11 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| ITUB4 | 95,00% | 95,50% | 95,00% | 90,91% | 15 |
| MEAL3 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |
| PETR4 | 95,00% | 95,38% | 95,00% | 87,50% | 15 |
| SHOW3 | 95,00% | 95,45% | 95,00% | 100,00% | 15 |
| VALE3 | 100,00% | 100,00% | 100,00% | 100,00% | 15 |

Fonte: Autor

## 6.5. Comparação dos Modelos desenvolvidos - *RF X LGBM*

Considerando os dados coletados, pôde-se perceber que, na divisão temporal as medidas para o conjunto de treino se mostraram similares para os dois algoritmos, contudo para o conjunto de testes o algoritmo LGBM se mostrou melhor considerando as médias de todos os ativos escolhidos conforme *Figura 24 - Divisão temporal - LGBM x Random Forest (Médias)* Figura *24* - Divisão temporal - LGBM x Random Forest (Médias).

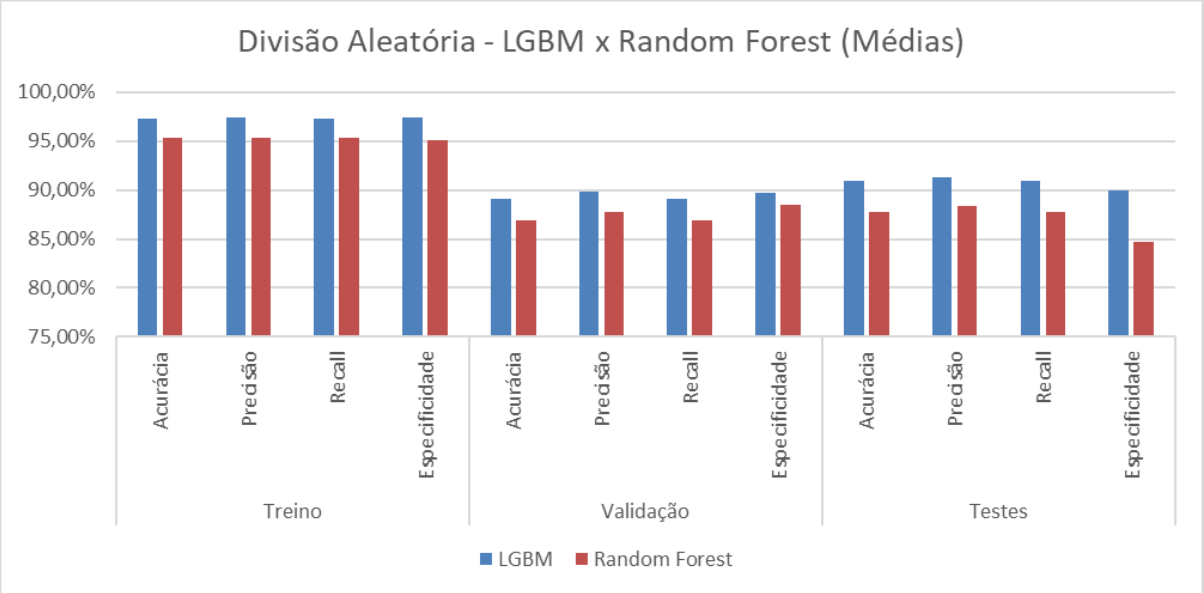
Figura 24 - Divisão temporal - LGBM x Random Forest (Médias)




Fonte: Autor

Pôde-se perceber que, na divisão aleatória, o algoritmo LGBM se mostrou melhor basicamente em todos os conjuntos de dados (treino, validação e testes) considerando as médias de todos os ativos escolhidos conforme *Figura 25 - Divisão aleatória - LGBM x Random Forest (Médias)*.

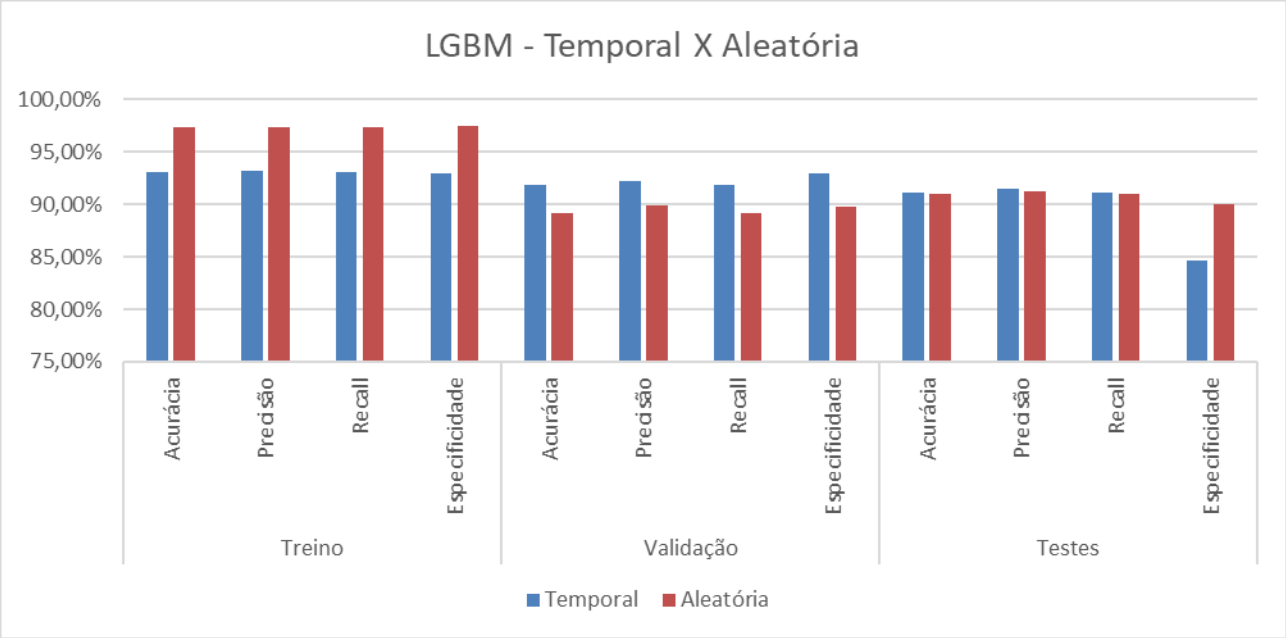
Figura 25 - Divisão aleatória - LGBM x Random Forest (Médias)




Fonte: Autor

Pôde-se perceber que, para o algoritmo LGBM conforme *Figura 26 - LGBM - temporal x aleatória*, a abordagem aleatória apresentou uma média maior para as medidas no conjunto de dados de treino, mas ao analisar o conjunto de validação e testes esse cenário não se confirma. Apesar disso, não dá para afirmar que a divisão temporal é melhor pois ambas as abordagens apresentaram o mesmo número de ativos com o problema de *overfitting* para os dois conjuntos de dados validação e testes.

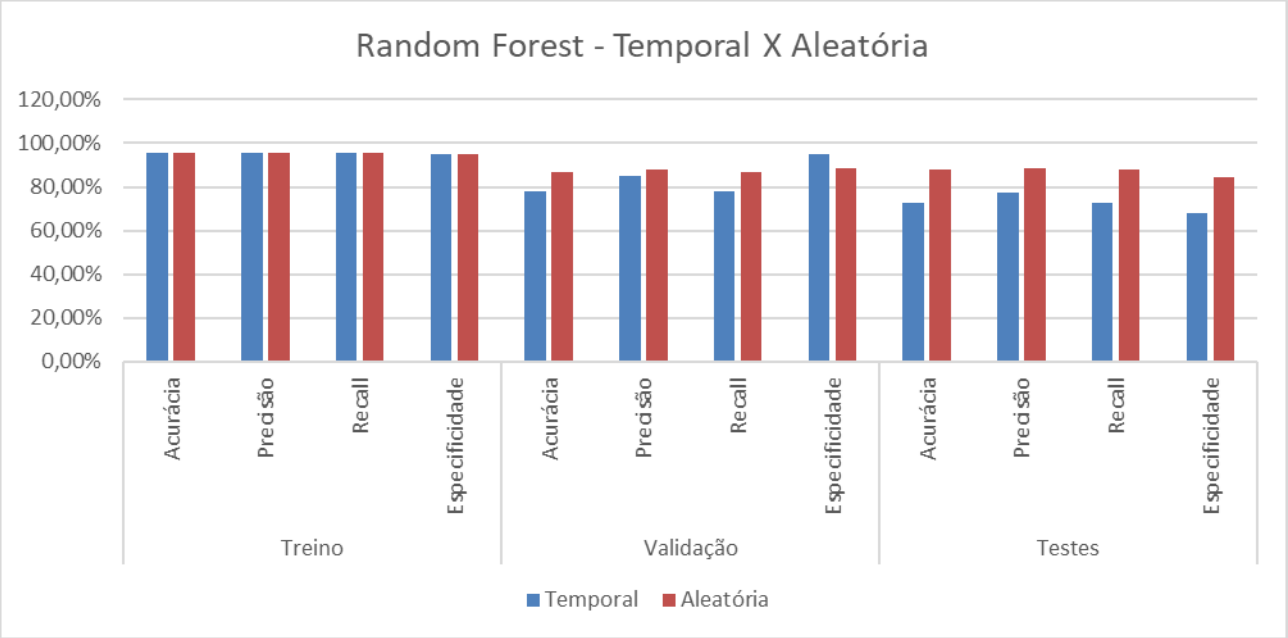
Figura 26 - LGBM - temporal x aleatória




Fonte: Autor

Pôde-se perceber que, para o algoritmo *Random Forest* conforme *Figura 27- Random Forest - temporal x aleatória*, a abordagem aleatória e a temporal apresentaram médias similares para as medidas no conjunto de dados de treino, contudo para os demais conjuntos de dados - validação e testes - a abordagem aleatória se mostrou mais eficiente.

Figura 27- Random Forest - temporal x aleatória




Fonte: Autor

# 7. Conclusão

O trabalho procurou apresentar algumas abordagens de aprendizado de máquina para previsão em séries temporais financeiras e, a partir destas, foram escolhidos dois algoritmos *Random Forest* e LGBM. Foram apresentadas diferentes técnicas de manipulação e validação dos dados e foi escolhida a técnica de barras de preço (acúmulo de ocorrências) ilustrada na seção 3.5.

Conforme ilustrado na seção **6.5. Comparação dos Modelos desenvolvidos - *RF X LGBM*** e considerando o problema a ser investigado, apesar da divisão aleatória apresentar medidas mais eficientes é pertinente que se utilize a divisão temporal para um sistema real.

Diante dos algoritmos avaliados o LGBM se mostrou melhor mesmo não passando por nenhuma otimização dos seus hiperparâmetros.

## 7.1. Trabalhos Futuros

Como o intuito de aprimorar os resultados obtidos com os algoritmos, algumas abordagens futuras podem ser exploradas, a saber:

- Aumento de características derivadas na série de dados atuais, ou até mesmo informações de contexto macroeconômico como por exemplo inflação, taxa de juros entre outras;
- Aprimoramento dos algoritmos através da otimização dos hiper parâmetros presentes em cada uma das técnicas;
- Utilização de outros algoritmos e comparação entre eles para escolher aquele de melhor desempenho;
- Utilização de técnicas de processamento de linguagem natural para avaliar as publicações dos órgãos reguladores do mercado para identificar tendências nas comunicações.

# 8. Apêndice

## 8.1. Lista de funções implementadas/customizadas

| Função | Descrição |
|---|---|
| importa_cotacao | Função utilizada para a importação dos dados oriundos da B3. |
| trata_dados | Função implementada para tratar os dados (retirar os atributos não utilizadas) e para transformar os dados de preço na função log presente na biblioteca numpy. |
| particiona_conjunto_alvo | Função para efetuar o processo de acúmulo de ocorrêcnias/dias a partir dos dados tratados. |
| rotula_cotacao | Função implementada para efetuar a rotulagem do atributo criado para a predição dos modelos. |
| particiona_dados_teste | Função para particionar o conjunto de dados rotulados em dois conjuntos (treino/validação e testes). |
| particiona_dados_validacao | Função para particionar o primeiro conjunto de dados (treino/validação) em dois conjuntos de treino e validação. |
| calc_acc_prec_rec | Função para calcular as medidas acurácia, precisão e recall, especificidade para o conjunto de dados. |
| retorna_score | Função que reotorna a lista das medidas de desempenho. |
| tick_bar_df_medio | Função adaptada para calcular a média de preço tomando como base as funções tick_bars e tick_bar_df da API utlizada. |
| volume_bar_df_medio | Função adaptada para calcular a média de volume tomando como base as funções volume_bars e volume_bar_df da API utlizada. |
| tick_bars | Funções presentes na API citada na seçao 3.5 (https://github.com/jjakimoto/finance_ml) |
| tick_bar_df | |
| volume_bars | |
| volume_bar_df | |

Essas funções serão disponibilizadas no repositório do GitHub.
https://github.com/JCS972/TCC_IA_BigData20_22

# 9. Bibliografia


1. LÓPEZ DE PRADO M. (2018) – **Advances in financial machine learning.** 1ed. New Jersey: John Wiley & Sons, Inc.

2. MESQUITA C.M.H.S. R. (2019) – **Ciência de dados e aprendizado de máquina para predição em séries temporais financeiras**. Dissertação de Mestrado - Belo Horizonte.

3. DERBENTSEV V., MATVIYCHUK A., DATSENKO N., BEZKOROVAINYI V. e AZARYAN A. (2020) - **Machine learning approaches for financial time series forecasting**. *Proceedings of the Selected Papers of the Special Edition of International Conference on Monitoring, Modeling & Management of Emergent Economy* (M3E2-MLPEED 2020), Odessa, Ukraine, July 13-18, 2020, Published on CEUR-WS: 26-Oct-2020. http://ceur-ws.org/Vol-2713/paper47.pdf

4. VARGHADE P., PATEL R.: **Comparison of SVR and Decision Trees for Financial Series Prediction**. IJACTE 1(1), 101–105 (2012).

5. Breiman, L., Friedman, H., Olshen, R.A., Stone, C.J.: **Classification and Regression Trees.** Chapman & Hall/CRC, Boca Raton (1984)

6. Breiman, L.: **Random Forests. Machine Learning** 45, 5–32 (2001). doi:10.1023/A:1010933404324.

7. Rosenblatt, F. (1958) - **The perceptron: A probabilistic model for information storage and organization in the brain**. Psychological Review, p. 65–386. Citado na página 25.

8. ASSIS, C. A. S. de. (2019) - **Predição de tendências em séries financeiras utilizando meta-classificadores** - Tese de Doutorado – CEFET – Centro Federal de Educação Tecnológica de Minas Gerais.

9. Barreto, J. M. (2002) - **Introdução às Redes Neurais Artificiais, Laboratório de Conexionismo e Ciências Cognitivas** - UFSC -Departamento de Informática e de Estatística - Florianópolis – SC.