UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE ENGENHARIA DA COMPUTAÇÃO

DOUGLAS FELIPE JOHANN

ESTUDO E DESENVOLVIMENTO DE ROBÔ PREDITOR PARA ANÁLISE DE ATIVOS FINANCEIROS UTILIZANDO MACHINE LEARNING

São Leopoldo
2020

DOUGLAS FELIPE JOHANN

ESTUDO E DESENVOLVIMENTO DE ROBÔ PREDITOR PARA ANÁLISE DE ATIVOS FINANCEIROS UTILIZANDO MACHINE LEARNING

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação, pelo Curso de Engenharia da Computação da Universidade do Vale do Rio dos Sinos - UNISINOS

Orientador: Prof. Dr. Lucio Renê Prade

São Leopoldo

2020

AGRADECIMENTOS

Gostaria de agradecer a todas as pessoas que me acompanharam durante a jornada de oito anos trilhados para a conclusão do curso de Engenharia da Computação.

Aos familiares e amigos aos quais me mantive ausente por longos períodos em que o cansaço e o excesso de trabalho fizeram por me isolar na resolução das tarefas.

Aos colegas de trabalho que por muitas vezes me ajudaram com atividades de aula, fomentando discussões e ampliando meus conhecimentos em diversos assuntos.

Aos professores da UNISINOS que auxiliaram em meu caminho com conhecimento, inspiração, visão e conselhos.

Aos professores e alunos que formaram o excelente Ensino Propulsor pelas horas e horas em que me ajudaram com conteúdos difíceis, e com sua amizade.

À minha companheira Betina, pelo seu amor, paciência e apoio incondicionais, sem os quais minha jornada teria sido vazia e apática.

Ao programa Universidade para Todos – ProUni, sem o qual seria impossível para alguém com as condições que eu tive, estudar e concluir uma graduação superior em Engenharia da Computação.

Em um país que valoriza cada vez menos a educação, me sinto extremamente grato por ter valores e princípios que me fazem acreditar no contrário. Educação é a chave para a mudança de cultura e direção que o Brasil precisa a fim de se tornar uma pátria de verdade para todos os brasileiros.

RESUMO

Prever o comportamento de ativos no mercado financeiro de renda variável é uma tarefa complexa. Desde o início dos pregões eletrônicos, métodos matemáticos e estatísticos são utilizados na tentativa de maximizar lucros dos investidores. Nos últimos anos, os avanços na área de inteligência artificial, principalmente no campo de *machine learning*, fomentaram o surgimento de modelos e estratégias de predição que já começam a ser adotados por grandes instituições financeiras. Nesse contexto, o presente projeto apresenta um estudo para o desenvolvimento de um robô preditor, que tem a finalidade de explorar os recursos supracitados como ferramenta de análise e predição do comportamento de ativos financeiros. A partir de dados históricos de cotações o robô constrói modelos de regressão baseados nos algoritmos *k-Nearest Neighbors* (KNN), *Random Forests Regressor* (RFR) e *Support Vector Regressor* (SVR), além de utilizar a técnica de *stacking* composta dos três algoritmos, com objetivo de prever o comportamento do ativo no próximo período. Após a criação dos modelos a sua performance é aferida através do indicador *root-mean square error* (RMSE) e da taxa de acerto em relação à direção do ativo, denominada ACC. O software é parametrizável através de um arquivo em formato *JavaScript Object Notation* (JSON) e foi construído na linguagem de programação *Python*, utilizando a biblioteca *sklearn*. Para validação do sistema como um todo, foram realizadas predições para os ativos da ITUB4, PETR4 e WEGE3, listados na Brasil Bolsa Balcão no mês de outubro de 2020, em que foram obtidos resultados satisfatórios com ACC de 76,1% para ITUB4 e PETR4, e 80,9% para WEGE3, à exceção da métrica RMSE que apresentou resultados insatisfatórios comparados ao período de teste.

Palavras-chave: Mercado Financeiro. Robô Preditor. Inteligência Artificial. *Machine Learning*.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

B3	Brasil Bolsa Balcão
FIX	*Financial Information eXchange* (Troca de Informações Financeiras)
HFT	*High-Frequence Trading* (Negociação de Alta Frequência)
IA	Inteligência Artificial
KNN	*k-Nearest Neighbors* (k-Vizinhos Próximos)
SVM	*Support Vector Machines* (Máquinas de Vetor de Suporte)
SVR	*Support Vector Regressor* (Regressor de Vetor de Suporte)
RF	*Random Forests* (Florestas Aleatórias)
RFR	*Random Forests Regressor* (Regressor de Florestas Aleatórias)
XGBoost	*eXtreme Gradient Boosting* (Aumento de Gradiente Extremo)
RMSE	*Root Mean Squared Error* (Raiz do Erro Quadrático Médio)
ACC	Acurácia Customizada
CSV	*Comma-Separated Values* (Valores Separados por Vírgula)

SUMÁRIO

1 INTRODUÇÃO

O mercado financeiro de renda variável apresenta uma grande volatilidade no valor de seus ativos. Essa característica é buscada por muitos investidores por permitir ganhos percentuais maiores do que meios mais tradicionais de investimento, como a renda fixa. Para obter sucesso, o investidor precisa prever qual a tendência que o ativo tem de se valorizar ou desvalorizar. Diversos recursos são utilizados para essa avaliação, entre eles a análise técnica, fundamentalista, métodos matemáticos e algoritmos avançados.

O uso de robôs investidores surgiu para automatizar esse processo utilizando algoritmos pré-definidos para tomada de decisões. Atualmente estima-se que 90% das operações realizadas na bolsa de valores americana sejam feitas de forma automática, por robôs (CNBC, 2017). Nos últimos anos, empresas passaram a utilizar conceitos de inteligência artificial nos seus robôs de operação, a exemplo do influente banco europeu JPMorgan (NOONAN, 2017). Ferramentas baseadas nos conceitos de IA são recentes no mercado financeiro, porém têm potencial enorme e já existem vantagens bem evidentes no uso das mesmas relatadas por empresas como Citigroup, KPMG e PanAgora (CoinTimes, 2019).

Buscando explorar os recursos da IA como ferramenta de análise e predição do comportamento de ativos, o presente trabalho tem por objetivo a construção de um robô preditor que empregue conceitos de *machine learning* para estimar valores de cotação futuras de ativos financeiros. Adicionalmente, busca através da análise de sua performance preditiva justificar o uso das ferramentas utilizadas como instrumentos de análise de ativos na tomada de decisões, seguindo a tendência das instituições financeiras nas quais os estudos em IA para melhoria de processos têm ganhado cada vez mais relevância.

1.1 Objetivos

O objetivo deste trabalho é desenvolver e verificar o desempenho um sistema capaz de prever cotações de ativos da B3 através de dados históricos, utilizando técnicas e algoritmos de *machine learning*.

1.1.1 Objetivos Específicos

A proposta de desenvolvimento deste trabalho é dividida nos seguintes objetivos específicos:

a) obtenção e pré-processamento de dados de cotações em período diário dos ativos analisados, históricos e atuais;
b) criação de modelos de predição para prever a cotação que o ativo terá no fim do próximo período;
c) avaliação da performance preditiva dos modelos criados.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma revisão de conceitos de investimento no mercado financeiro e de inteligência artificial, com foco no campo de *machine learning* e dos algoritmos supervisionados de predição.

2.1 Estratégias de operação no mercado financeiro

Diversas formas de trabalho são utilizadas pelos investidores diariamente, desde as mais conservadoras até as mais ousadas. Este item procura abordar as estratégias mais conhecidas com a finalidade de fornecer um contexto para que posteriormente sejam abordados métodos mais sofisticados de análise.

2.1.1 Formas de operação

O jeito mais tradicional de se investir em ativos de renda variável é comprar ações de empresas consideradas sólidas e obter retorno no longo prazo. Nessa forma de investimento, são observados os fundamentos financeiros e estruturais das empresas. Analisando através de um espectro amplo nota-se que historicamente o índice das bolsas de valores tende a se valorizar com o crescimento econômico de um país e suas referidas empresas. O comportamento pode ser visto na figura 1.

Figura 1 – Variação do índice Ibovespa em períodos mensais desde 1995

Fonte: Adaptado de Site TradingView, 2020.

Segundo Xavier (2009), essa forma de investimento é conhecida como *Position Trade*, em que o foco de retorno do investimento está no longo prazo. Da mesma forma, o investimento também está suscetível às quedas causadas por grandes crises. Na figura 1 nota-se que entre o período de 2008 a 2017 o índice registrou forte queda e veio a se recuperar no fim do intervalo. Pode-se associar esse período à grandes acontecimentos econômicos e políticos que afetaram o Brasil, dentre eles a crise econômica internacional de 2008 e 2009, a operação Lava Jato, protestos populares contra o governo e impeachment da atual presidente no cargo.

A volatilidade é característica marcante no ambiente de investimentos de renda variável, tanto a longo como a curto prazo. Diversas estratégias foram construídas de forma a explorar essa característica. É comum encontrarmos variações significativas no valor de um ativo durante períodos curtos. A figura 2 apresenta a volatilidade de um ativo da B3 em períodos diários, com destaque para as marcações nos dias em que houve alta volatilidade.

Figura 2 – Variação do ativo ITUB4 em dezembro de 2019

Data	Fechamento	Variação	Variação (%)	Abertura	Máxima	Mínima	Volume
30 Dez 2019	37,10	-0,20	-0,54%	37,34	37,59	36,86	9.225.800
27 Dez 2019	37,30	-0,10	-0,27%	37,50	37,69	36,91	13.455.300
26 Dez 2019	37,40	0,56	1,52%	36,80	37,40	36,65	16.333.100
25 Dez 2019	36,84	0,00	+0,00%	36,50	36,84	36,31	0
24 Dez 2019	36,84	0,00	+0,00%	36,50	36,84	36,31	0
23 Dez 2019	36,84	0,34	0,93%	36,50	36,84	36,31	11.691.500
20 Dez 2019	36,50	-0,16	-0,44%	36,63	36,70	36,30	24.863.500
19 Dez 2019	36,66	0,18	0,49%	36,16	36,71	36,11	22.053.300
18 Dez 2019	36,48	0,46	1,28%	35,92	36,75	35,73	40.451.900
17 Dez 2019	36,02	0,54	1,52%	35,75	36,28	35,54	19.250.800
16 Dez 2019	35,48	-0,83	-2,29%	36,44	36,64	35,48	21.663.700
13 Dez 2019	36,31	0,72	2,02%	35,80	36,37	35,35	27.255.500
12 Dez 2019	35,59	-0,01	-0,03%	35,95	36,09	35,35	23.355.500
11 Dez 2019	35,60	-0,42	-1,17%	36,50	36,67	35,56	37.043.600
10 Dez 2019	36,02	-0,53	-1,45%	36,55	36,77	35,78	21.527.300
09 Dez 2019	36,55	0,70	1,95%	36,00	36,88	35,77	26.700.900
06 Dez 2019	35,85	-0,66	-1,81%	36,50	36,68	35,85	28.794.200
05 Dez 2019	36,51	-0,05	-0,14%	36,37	37,07	36,30	29.077.100

Fonte: Adaptado de ADVFN, 2020.

Trabalhar com a volatilidade significa explorar as quedas e altas de curto prazo para operar e realizar lucro. Operações curtas são divididas em duas modalidades: *swing trade e day trade*. A primeira comumente é utilizada para transações espaçadas em períodos de poucos dias, com análise dos preços das últimas semanas. Construir e encerrar uma operação no mesmo dia caracteriza a operação de *day trade* ou *intraday*. Para operar dessa forma, os investidores se valem de várias ferramentas de análise que buscam prever a tendência em tempo real de um ativo se valorizar ou desvalorizar nos próximos períodos, normalmente horas e minutos, no caso de investidores que utilizam análise técnica, podendo chegar no nível dos milésimos de segundo com o uso de robôs de alta frequência (XAVIER, 2009).

2.1.2 Análise fundamentalista

Segundo Rocha (2011), a estratégia, que é utilizada pelo investidor mais famoso do mundo, Warren Buffet, tornou-lhe a pessoa mais rica do mundo em 2008. Analisando indicadores das empresas e avaliando os balanços financeiros, bem como informações sobre plano de expansão e informes aos investidores, é possível ter uma boa noção sobre a situação da companhia e uma perspectiva sobre o futuro. Conhecida como análise por fundamentos ou fundamentalista, essa estratégia tem objetivo de lucro ao longo prazo com o crescimento das referidas empresas e os juros compostos. Além da valorização do papel, também é possível obter uma parte do lucro dessas empresas de volta através de dividendos e juros sob o capital próprio.

2.1.3 Análise técnica

Uma das análises mais conhecidas para operação é chamada de análise técnica e se baseia no estudo e visualização do gráfico de *candles* (do inglês, velas). A cada período de análise se forma um *candle*, que apresenta o valor de abertura, de fechamento, os picos de valor atingidos pelo ativo e se houve valorização ou desvalorização. Segundo Wawrzeniak (2014), a análise técnica descreve três pilares fundamentais. O primeiro deles aponta que o gráfico já desconta todos os

acontecimentos externos, ou seja, seus impactos já estão sendo considerados. O segundo aponta que os ativos se movem em tendências e, o terceiro, descreve que a história do ativo tende a se repetir, executar os mesmos padrões de movimento. A figura 3 apresenta um trecho do gráfico de *candles* da criptomoeda Bitcoin em período diário.

Figura 3 – Bitcoin, gráfico diário. Período de 4 de maio a 27 de junho de 2020

Fonte: Adaptado de Site TradingView, 2020.

Cada retângulo da figura 3 corresponde a um *candle*, que resume o que aconteceu com o ativo no último período definido. Um conjunto de *candles* mostra determinado comportamento que o ativo está tendo. De acordo com a figura formada pelos últimos períodos, são identificadas formas que segundo a análise técnica podem indicar o comportamento do ativo nos *candles* seguintes. A partir disso, o investidor toma decisões de compra e venda traçando preços alvo de ganho e perda (WAWRZENIAK, 2014).

Segundo Debastiani (2008), a partir da análise técnica derivam diversos outros indicadores, que podem ser divididos entre três grupos principais: seguidores de tendência, osciladores e operadores por canais. Os primeiros se referem a dados históricos e são utilizados para confirmar se uma tendência está consolidada. Nesse item encaixam-se indicadores como as médias móveis diversas. Os osciladores já trabalham com a predição de movimentos, indicando se uma tendência pode se reverter, como é o caso do Índice de Força Relativa. Por último, os operadores de canais são uma combinação dos dois primeiros para definir se o preço do ativo está próximo ou distante do valor considerado ideal, com o exemplo das Bandas de *Bollinger*.

2.2 Métodos computacionais e sistemas automatizados de investimento

O escopo deste trabalho não prevê a operação financeira dos ativos na B3, porém acredita-se ser relevantes os conteúdos referentes aos sistemas automatizados de investimento por serem comumente integrados a algoritmos de *machine learning* e outros métodos matemáticos e estatísticos para as tomadas de decisões.

Aldridge (2010) cita que os avanços tecnológicos no campo da computação aplicada provocaram mudanças no mundo das finanças. A execução de ordens, que antes levava um tempo considerável para ser feita devido a sistemas não informatizados foram comprimidas e atualmente ocorrem imperceptivelmente ao olho humano. Modalidades de investimento baseadas em explorar essa característica surgiram, aliados de conhecimentos na área de estatística e matemática. Simulações quantitativas envolvendo grandes conjuntos de dados se tornam agora possíveis e estratégias podem ser construídas com maior eficiência.

Fundos de investimento que se utilizam dos métodos supracitados são comuns, a exemplo do *Renaissance*. Esse fundo que é composto na maioria de cientistas e grandes estudiosos do campo das finanças, desde 1988 tem rendimento anual médio de 80% (ORDONES, 2014).

Aldridge (2010), propõe a criação de um modelo de trading automatizado em três etapas, conforme é apresentado na figura 4. A autora descreve com foco nos sistemas de HFT (do inglês, *High-Frequence Trading*).

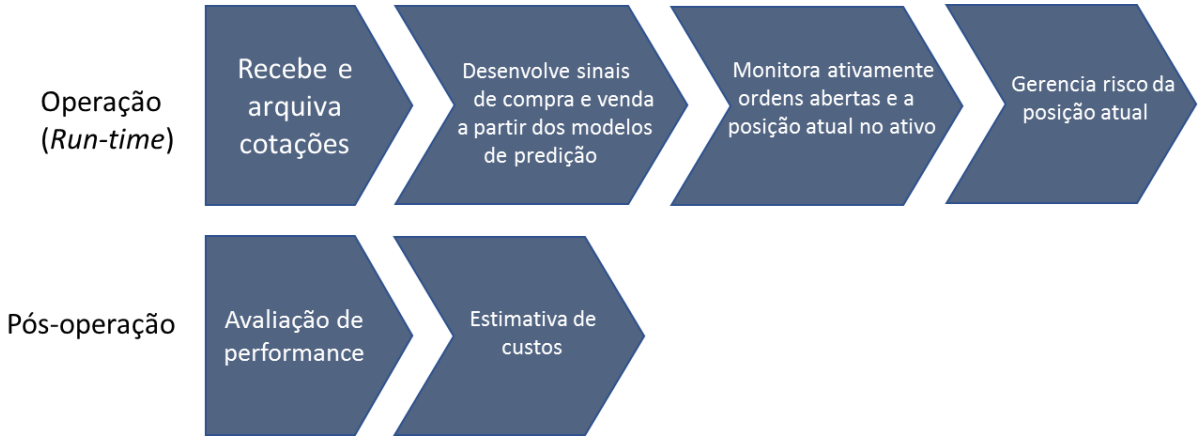
Figura 4 – Ciclo de desenvolvimento de um sistema de operação automatizado.

Fonte: Adaptado de Aldridge, 2010.

a) Modelos: consiste em modelos quantitativos para predição de curto prazo baseada em dados históricos do ativo. A autora elabora que modelos quantitativos são métodos matemáticos, estatísticos e algorítmicos de predição baseada em dados históricos do ativo. O desenvolvimento de tais métodos deve contar com testes em séries de dados temporais históricas do ativo simulando uma operação em tempo real. Tal processo é referenciado como *back-testing* e normalmente se requere que o modelo seja executado em pelo menos dois anos de dados compilados para validar o seu funcionamento.

b) Sistemas: poder de processamento computacional, programas conectados às corretoras e bolsas de valores de forma que rapidamente possam executar e monitorar ordens alimentadas pelos modelos quantitativos. Normalmente os modelos econométricos e algoritmos são criados em linguagens de análise de dados, como *MatLab* ou *Python*. Após esse desenvolvimento, os programas são traduzidos para uma linguagem mais poderosa e performática, como o C++. O desempenho é necessário principalmente para execução em tempo real, quando ao mesmo tempo que avalia dados históricos deverá identificar pontos de entrada e avaliar o risco das operações. Os sistemas ativamente monitoram as cotações do ativo para fornecer sinais de compra e venda. Quando assume uma posição, acompanha as ordens ativas e gerencia o risco da negociação, saindo quando receber o sinal correspondente. Após a execução a operação é avaliada e então gerada estimativa de custos. A figura 5 apresenta o fluxo de operação em detalhes.

Figura 5 – Fluxo de operação do sistema automatizado



Fonte: Adaptado de Aldridge, 2010.

c) Capital: é o módulo que compreende o capital aplicado e monitorado através de estratégias de gerenciamento de risco e de custos precisos e cautelosos. Considerado como o fator chave para o sucesso do sistema, parametriza os limites e tem como objetivo proteger o sistema, mitigando possíveis danos que possam ocorrer durante a operação.

A maioria dos sistemas automatizados de operação criados atualmente são feitos para operar independentemente de plataforma de operação. Isso é possível através de uma linguagem chamada FIX (do inglês, *Financial Information eXchange*), uma sequência otimizada e padronizada de instruções que é compatível com uma série de home brokers. Dessa forma, o código desenvolvido pode operar em qualquer ferramenta compatível com a linguagem, que atualmente é amplamente adotada (ALDRIDGE, 2010).

2.3 Inteligência Artificial

O termo inteligência artificial pode ser definido como a capacidade de uma máquina tomar decisões consideradas inteligentes. A partir disso, pode-se teorizar em cima do que de fato significa a inteligência. Historicamente, esse conceito pode ser analisado de quatro formas diferentes: pensamento humano, pensamento racional, ação humana e ação racional. Pelo lado humano, podemos considerar uma máquina inteligente se tomar decisões similares a que um ser humano tomaria. Pelo lado racional, podemos considerar inteligente um artefato que tome a melhor

decisão possível de acordo com a gama de dados disponível (RUSSEL; NORVIG, 2010).

2.4 Machine Learning

Segundo Chiavegatto (2018), inicialmente os algoritmos de IA baseavam-se em conjuntos de regras prontas e, portanto, tomavam decisões conhecidas de antemão. Era possível facilmente verificar se o algoritmo estava tomando a decisão correta, porém a capacidade dele era limitada ao conjunto de decisões pensado pelo programador. O advento do campo da inteligência artificial chamado *machine learning* trouxe mudanças para essa forma de operação. Segundo Géron (2017), pode ser definida como a ciência e arte de programar computadores para que possam aprender a partir de uma série de dados.

A grande diferença conceitual para os métodos já existentes é que a máquina é capaz de aprender as regras de tomada de decisão e se adaptar, modificando-as de acordo com suas próprias experiências. Baseado em uma entrada de dados e no resultado esperado, esses algoritmos têm a capacidade de compreender o que faz com que tais dados atinjam o resultado, ou não atinjam. Ao contrário do que era feito, portanto, o próprio algoritmo agora cria suas regras de decisão (CHIAVEGATTO, 2018).

O uso de *machine learning* constitui uma alternativa valiosa quando aplicada a problemas de decisão complexos. Como exemplo, Géron (2017) cita o desenvolvimento de um algoritmo para detectar *spam* em uma caixa de *e-mail*. Em uma abordagem sem *machine learning* teríamos que programar as regras uma a uma que farão o sistema detectar através das palavras do *e-mail* características que o tornam um possível *spam*, o que é uma tarefa complexa. O programa tornar-se-ia com o tempo um grande conjunto de regras complexas e necessitaria de manutenção constante para inclusão de novas regras. Com técnicas de *machine learning* o programa aprenderia automaticamente quais características que fazem o *e-mail* ser um *spam*, detectando padrões complexos de acordo com a amostragem de dados. O programa é mais curto, mais fácil de manter e provavelmente muito mais preciso.

2.4.1 Classificação quanto à forma de treinamento

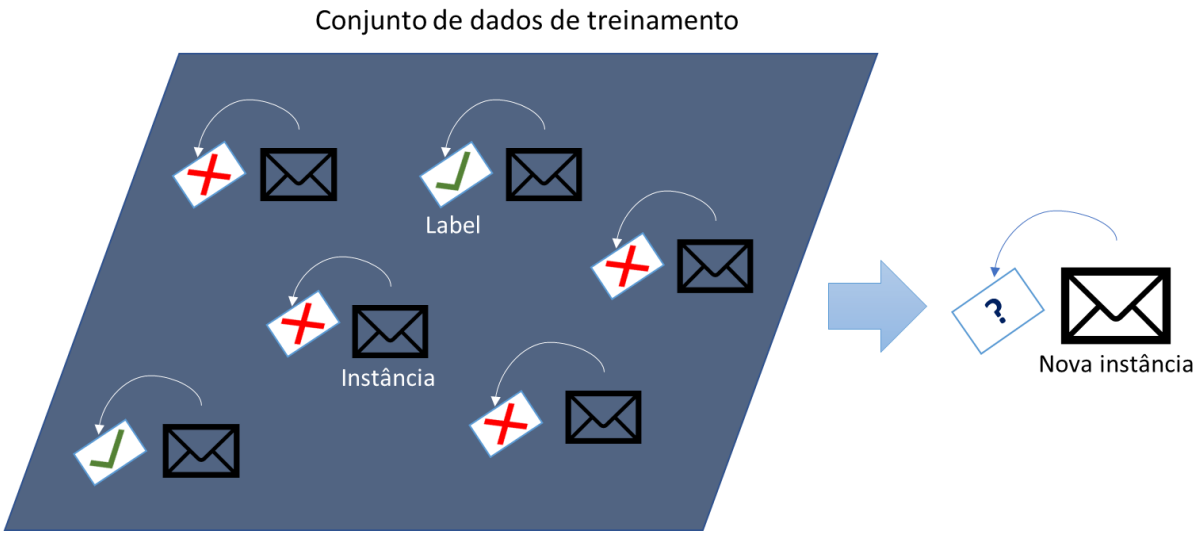
Os algoritmos de *machine learning* precisam ser treinados para identificar padrões do problema em que está inserido. Quanto ao treinamento, Géron (2017) cita que uma das formas de classificar os sistemas se refere ao tipo de supervisão que terão durante o treinamento, podendo ser supervisionada, não supervisionado, semi- supervisionado e aprendizado por reforço.

Chiavegatto (2018) cita as características de cada um dos tipos de treinamento. O treinamento supervisionado é indicado para tarefas de predição, por identificar padrões em conjuntos de dados que indicam a probabilidade de determinada resposta acontecer. Dos demais tipos de treinamento, o não supervisionado é comumente utilizado para clusterização e agrupamento de dados. O algoritmo semi-supervisionado apresenta uma junção dos dois primeiros métodos, utilizado por exemplo para agrupamento de dados similares e classificação em grupos. O conceito de aprendizado por reforço tem dado resultados surpreendentes na área da *machine learning*, podendo-se citar a vitória do sistema AlphaGo frente ao campeão mundial do jogo de estratégia Go. Esse último método se baseia em interações dinâmicas através de um sistema de feedbacks com punições e premiações.

2.4.2 Treinamento supervisionado

Conforme descrito por Géron (2017), no treinamento supervisionado os dados com os quais o algoritmo é alimentado já contém as respostas ou soluções corretas. Essas respostas são chamadas de *labels* (do inglês, etiquetas), ou *targets* (do inglês, alvos), conforme pode ser visto na figura 6.

Figura 6 – Exemplo de treinamento supervisionado em *machine learning*



Fonte: Adaptado de Géron (2017).

Exemplos de uso do método supervisionado incluem a predição de um valor numérico, como por exemplo, o valor de um automóvel baseado nas suas condições como idade, marca, modelo e quilometragem.

2.4.3 Modelos preditivos

Conforme visto na sessão anterior, nos algoritmos supervisionados de *machine learning* os dados utilizados para alimentar o algoritmo contêm o resultado esperado. Dessa forma, sua finalidade é identificar quais as características que fazem com que eles atinjam o resultado esperado. Modelos preditivos se beneficiam dessa característica diretamente. Chiavegatto (2018) cita que os métodos preditivos podem ser classificados em dois grandes grupos:

a) Algoritmos de classificação: quando o resultado da predição é qualitativo, a respeito de identificação de causas prováveis para acontecimentos. Como exemplo, podemos considerar a predição da causa de óbito para dado paciente no sistema de saúde, considerando as características apresentadas pelo mesmo em comparação com pacientes com dados similares.
b) Algoritmos de regressão: a variável a ser predita é quantitativa, um número a ser estimado. Como exemplo, qual será o IMC corporal de dado paciente daqui há dois anos baseado as características apresentadas.

Chiavegatto (2018) aborda ainda a diferença entre problemas de predição e de inferência, citando que problemas de predição se preocupam com performance preditiva, ou seja, com a taxa de acerto de determinada questão. Como exemplo temos a probabilidade de um ativo se valorizar 3% no próximo dia. Inferência, por outro lado, procura estabelecer relações entre variáveis e o desfecho, como identificar se a variação no preço do barril de petróleo está associada à variação de preços nas ações das empresas petrolíferas.

2.5 Algoritmos de predição com aprendizado supervisionado

Nesta sessão serão abordados os algoritmos mais comumente empregados para análise preditiva no campo de *machine learning*, com a utilização de métodos supervisionados conforme abordado por Géron (2017). Também são revisados conceitos citados por Chiavegatto (2018), Harrison (2018) e Dwivedi (2020), principalmente no que diz respeito à utilização prática e treinamento.

2.5.1 *k-Nearest Neighbors* (KNN)

Do inglês, vizinhos mais próximos, o algoritmo KNN possui uma abordagem simples e fácil de ser implementada, podendo ser usado tanto para algoritmos de classificação como de regressão. A ideia principal é de que objetos similares existem em proximidade. O modelo trabalha com o conceito de distância entre o valor do dado observado com todos os dados contidos no conjunto, selecionando os K vizinhos mais próximos de seu valor. No caso de um problema de classificação utiliza sistema de votos para eleger o *label* mais frequente, e em regressões realiza a média dos *labels* encontrados. K é definido como um parâmetro para execução do algoritmo, experimentalmente definida através de vários treinamentos (HARRISON, 2018).

A grande desvantagem desse método é o aumento substancial de processamento necessário quando há muitas variáveis preditoras. Nesse sentido, existem algoritmos mais eficientes e com maior taxa de acerto. Sua vantagem está no fato de ser simples e bastante eficaz para algoritmos de classificação, como por

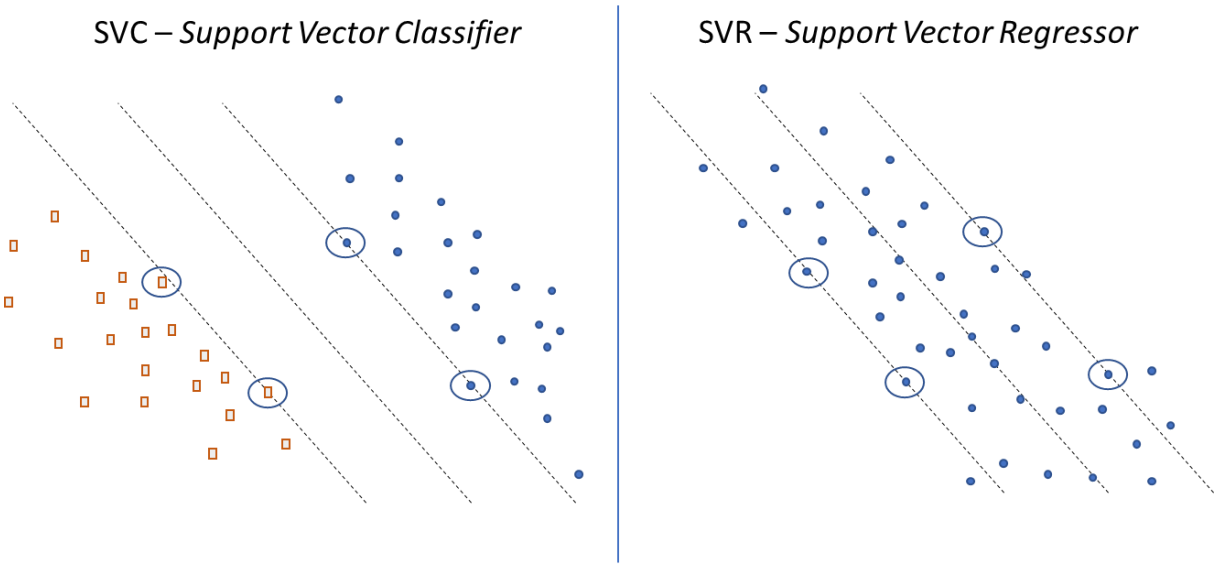
exemplo, identificar produtos com potencial de compra baseado nas informações de compras anteriores de clientes.

2.5.2 *Support Vector Machines* (SVM)

Thaller (2018) cita que os algoritmos *Support Vector Machines* (do inglês, máquina de vetores de suporte) são amplamente utilizados em séries históricas financeiras, com trabalhos acadêmicos referenciando o uso dessa técnica tanto de forma híbrida quanto isolada. Segundo Chiavegatto (2018) o objetivo de uma modelagem SVM é criar uma fronteira entre os dados analisados, chamada de hiperplano, que divide o espaço amostral em áreas similares. O conceito de vetores de suporte abrange pontos mais próximos de cada hiperplano.

O algoritmo pode ser utilizado tanto para classificação, quando recebe a designação de *Support Vector Classifier* (SVC), como para regressão, o chamado *Support Vector Regressor* (SVR). Para regressão, o algoritmo muda um pouco de conceito. Segundo Géron (2017), ao invés de buscar a separação dos hiperplanos, na regressão a ideia é preencher o maior número de pontos entre as margens, manipulando-se o tamanho da margem ou a distância máxima da localização dos planos através dos hiperparâmetros de configuração. Na figura 7 é apresentada a distinção de conceitos entre o SVM para classificação e regressão.

Figura 7 – Algoritmo SVM utilizado para classificação e regressão



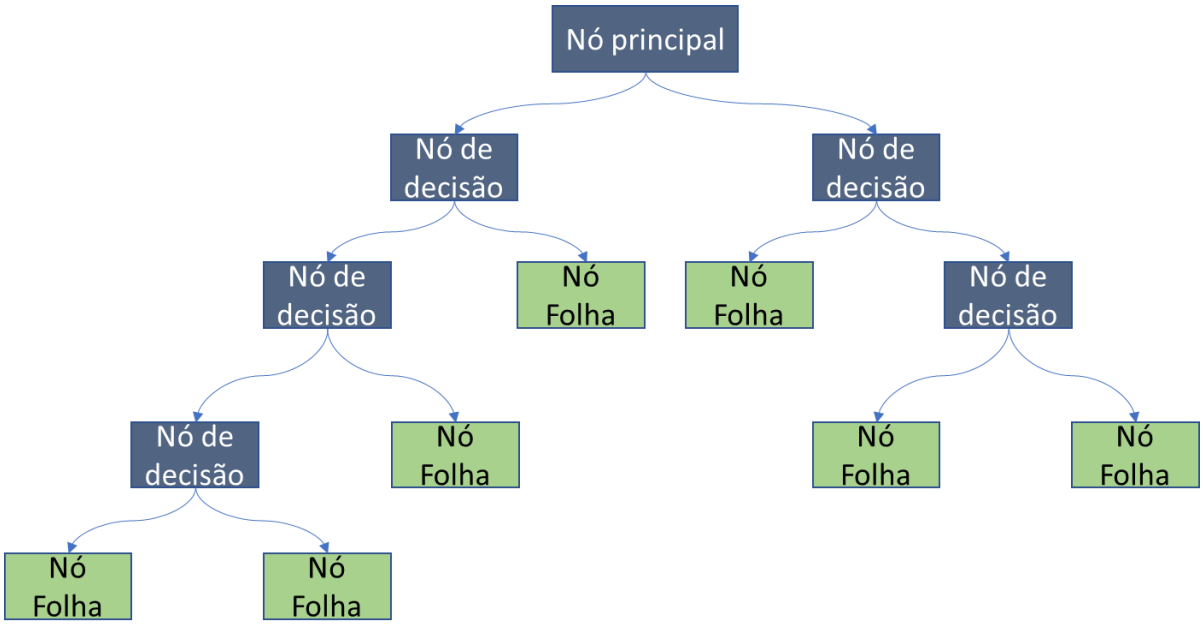
Fonte: Adaptado de Géron (2017).

Existem várias formas de implementação do algoritmo e o principal objetivo é a definição dos hiperplanos de forma que apresentem boa taxa de acerto em relação à separação dos dados, penalizando quando houver uma classificação errada. No caso da regressão, a manipulação das margens para encaixar a maior quantidade possível de pontos enquanto limita as violações as mesmas. Quanto maior o desvio, maior deve ser a penalização, comumente atribuída através do hiperparâmetro C (CHIAVEGATTO, 2018).

2.5.3 Árvores de decisão

Segundo Chiavegatto (2018) as árvores de decisão possuem o objetivo de separação das observações analisadas em grupos cada vez menores e homogêneos, atendendo ao desfecho de interesse. Utiliza o conceito de nós de decisão, iniciando por um nó principal e ramificando através de decisões até chegar no desfecho, chamado de folha. A figura 8 apresenta uma árvore de decisão conceitual simples. Nota-se que as decisões vão se aprofundando na árvore, ou ramificando.

Figura 8 – Árvore de decisão conceitual



Fonte: Elaborado pelo autor.

Entre as desvantagens apresentadas por esse tipo de algoritmo destacam-se a tendência de alta variância e necessidade forte de sobreajuste, fazendo com que

seu uso isolado não apresente bom resultado preditivo. Para solucionar essas dependências, existem métodos bastante eficazes de construção, manipulação dos preditores e amostras que garantem melhorar a performance preditiva consideravelmente, com maior destaque para as técnicas de *Random Forests* (do inglês, florestas randômicas) e *Gradient Boosted Trees* (do inglês, árvores com gradiente aumentado) (CHIAVEGATTO, 2018). A primeira é abordada no item 2.5.4.

2.5.4 *Random Forests* (RF)

Segundo Géron (2017), *Random Forests* é uma técnica de *ensemble* (do inglês, conjunto) que tem o objetivo de treinar várias árvores de decisão em amostras aleatórias de dados das *features*, para então calcular a média das suas predições. A ideia básica é obter o chamado senso comum, em que o conhecimento obtido a partir de uma larga população tende a superar o conhecimento de um único especialista. Os parâmetros de configuração são os mesmos das árvores de decisão, com a adição da variável *n_estimators*, que possibilita configurar o número máximo de árvores de decisão aleatórias que serão utilizadas na predição. O algoritmo é utilizado tanto para classificação como regressão, quando recebe o nome de *Random Forests Regressor* (RFR).

2.6 Pré-processamento de dados

A etapa de pré-processamento de dados é considerada de grande importância para o bom funcionamento de um modelo de *machine learning.* Sua aplicação envolve desde a validação dos valores da base dados quanto a divisão de subconjuntos homogêneos de dados, conforme será apresentado no item 2.6.1.

2.6.1 Estratificação dos dados

Segundo Géron (2017), é uma boa prática criar subconjuntos de dados para análise: treino e teste. Os dados de treino normalmente representam cerca de 80% dos dados da base e serão diretamente utilizados pelo algoritmo para aprendizado. Os 20% restantes são utilizados somente na etapa de teste, em que após gerado o modelo ele tenta prever os resultados desse subconjunto e tem sua performance

avaliada. A partir dos dados obtidos é possível verificar possíveis problemas e providenciar correções para otimizar os resultados. A técnica procura eliminar ou reduzir o problema de sobreajuste, que ocorre quando um modelo apresenta ótimos resultados em período de teste, porém ruins quando utilizado de forma produtiva.

O modo mais simples de efetuar a separação dos dados é por amostragem aleatória através de uma semente conhecida, porém isso pode gerar problemas de representatividade nos dados. Géron (2017) cita como exemplo uma técnica comumente empregada em pesquisas estatísticas. Caso uma pesquisa seja feita em uma cidade que possui mais mulheres do que homens, o conjunto das pessoas entrevistadas deve procurar manter essa proporção, tendo mais pessoas entrevistadas do sexo feminino.

No cenário desta pesquisa, isso significa que se distribuirmos os valores de cotação aleatoriamente corremos o risco de o modelo ter que prever respostas para valores que nunca enxergou no seu conjunto de treinamento. Em alguns algoritmos esse problema pode causar problemas consideráveis de performance. A estratificação busca então manter a proporção de valores próximas entre os subconjuntos criados, de forma a sanar esse risco.

2.6.2 Séries temporais e janelas

Nametala (2017) aborda em sua tese de mestrado o conceito das séries temporais, citando Moretin e Toloi (2006). Segundo os autores citados, qualquer conjunto de dados que possua amostras ordenadas ao longo do tempo pode ser considerada e analisada como uma série temporal. Esse conceito se aplica ao presente trabalho já que as cotações estão dispostas uniformemente ao longo do tempo.

No mercado financeiro é comum a ótica de análise de um ativo mudar de acordo com o período de análise determinado, que pode variar de segundos até meses. O conceito de janela propõe a criação de intervalos de tempo que serão considerados para a análise de um próximo período. Por exemplo, podemos analisar um ativo utilizando uma janela de oito períodos, o que significa que estão sendo observados os últimos oito registros (ou cotações) para determinar o valor do próximo registro. A figura 9 apresenta a diferença entre uma janela de apenas um período em comparação com uma janela de três. Nota-se que a janela x1 possui três

registros e cada um possui o seu *target* ou alvo, enquanto na janela x3, ou de três períodos, três registros são utilizados para a estimação de apenas um target.

Figura 9 – Comparação de janelas de um e três períodos.

Janela x1

feature			target
102168	56182800	24,34	24,29
101911	25170900	24,29	24,88
100721	81001700	24,88	24,96

→

Janela x3

feature			target
102168	56182800	24,34	
101911	25170900	24,29	24,96
100721	81001700	24,88	

Fonte: Elaborado pelo autor.

2.7 *Stacking*

A técnica de *stacking* (do inglês, empilhamento) baseia-se no conceito de agregação de preditores. De acordo com Géron (2017), o objetivo é treinar um novo modelo baseado em modelos previamente construídos, de forma que o último aprenda com os resultados obtidos pelos outros modelos. A técnica também é conhecida como a criação de uma segunda camada de predição, que busca otimizar os resultados previamente obtidos. No caso dos algoritmos de regressão, é comumente utilizada a regressão logística para a construção do modelo final, porém também são utilizados algoritmos mais complexos como o *XGBoost* (eXtreme Gradient Booosting).

XGBoost é uma biblioteca de *machine learning* desenhada para trabalhar com a técnica de *boosting*. Segundo Géron (2017), *boosting* se refere a combinar vários *weak learners* (aprendizes fracos) para gerar um modelo mais otimizado. É similar à técnica de *stacking*, porém o *XGBoost* se baseia no algoritmo de árvores de decisão implementando uma série de *boosted trees* (do inglês, árvores melhoradas), enquanto a primeira permite a utilização de algoritmos de outros tipos como os abordados anteriormente.

2.8 *Randomsearch*

Após a definição dos algoritmos inicia-se uma etapa também bastante importante para o processo de modelagem que é o *tunning* (do inglês, afinação) dos parâmetros. Cada algoritmo possui um conjunto de parâmetros específicos a ser considerado e alterá-los pode afetar os resultados da predição significativamente.

Para identificar os melhores parâmetros a serem utilizados no algoritmo existem técnicas como o *randomsearch* (do inglês, pesquisa aleatória).

Segundo Géron (2017), o algoritmo recebe como entrada uma série de valores possíveis para os hiperparâmetros de dado algoritmo, e cria combinações aleatórias desses valores que podem ser limitadas por um número máximo de interações. A pesquisa vai sendo efetuada no sentido de que os parâmetros que geram os melhores resultados são exibidos ao final e selecionados. Os resultados desejados devem ser especificados pelo usuário, que vão também depender das métricas relevantes para o algoritmo em questão. A técnica permite a separação do conjunto em n subconjuntos, para aferir de forma mais eficiente o resultado da modelagem e evitar o sobreajuste. Essa separação é chamada de *cross-validation.*

2.9 Métrica e validação de resultados

Algoritmos de regressão em *machine learning*, por preverem uma variável contínua, comumente possuem métricas de avaliação relacionadas ao desvio do valor real para o valor predito. Conforme abordado por Géron (2017), a métrica RMSE ou *Root Mean Squared Error* (do inglês, raiz do erro quadrático médio) é bastante utilizada nesse contexto por dar a ideia de qual a taxa de erro média do modelo quando faz uma predição. Quanto mais próximo zero, menor a taxa de erro associada ao modelo. A métrica se baseia no cálculo da raiz do quadrado da distância entre o valor predito e o valor real, e pela forma como é construída deixa mais evidentes grandes desvios entre esses valores, conforme equação apresentada na figura 10.

Figura 10 – Equação para cálculo da métrica da RMSE.

$$RMSE\,(X,h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m} h(x^{(i)}) - y^{(i)}\,^2}$$

Fonte: Adaptado de Gerón (2017).

2.10 Trabalhos Correlatos

Diversos trabalhos na área das finanças fazem uso de técnicas de *machine learning* para predição. Dentre os artigos e teses analisados destacam-se as seguintes consideradas de alta correlação com os assuntos envolvidos no presente trabalho.

Nametala (2017) apresenta a construção de um robô investidor que integra estratégias de negociação desenvolvidas a partir de redes neurais artificiais e indicadores econométricos para predição de ativos da B3. Através de seletores inteligentes que se atualizam progressivamente no tempo os métodos são otimizados, gerando cenários lucrativos. O robô criado pelo autor obteve taxa de acerto média de 58,3% no período simulado de 2012 a 2014.

Silva (2018) desenvolve em sua dissertação de pós-graduação um protótipo de robô investidor para operações de compra e venda de Minicontratos Futuros de Dólar. O sistema criado utiliza *machine learning*, especificamente a técnica SVM com indicadores de Análise Técnica para classificar e realizar predições sobre o movimento de ativos. O período de *backtesting* utilizada foi de quatro anos, entre 2014 e 2018, e o autor obteve taxas de acerto de até 78% nas predições.

Lu (2016) aplica diversos algoritmos de *machine learning* para medir performance preditiva no índice americano S&P 500. Entre eles regressão logística, SVM e variações dessas técnicas. O período de treinamento do sistema foi de dois anos (2014 e 2016) e o *backtesting* foi realizado em um período de três meses (fevereiro a maio de 2016). Apesar de considerar que o período de testes foi curto e seu aumento poderia melhorar a performance preditiva do sistema, foram obtidos aproximadamente 60% de taxa de acerto. O trabalho ainda conclui que a combinação de algoritmos mais simples de *machine learning*, e suas variações, como curvas de otimização ROC e métodos de *essemble* (do inglês, montagem) podem ter grande impacto nos resultados e aumentar a performance significativamente.

Ainda pode-se citar trabalhos como o de Nair et al. (2010), que utiliza árvores de decisão, redes neurais e lógica *fuzzy* para realização de predições no mercado de ações com taxa de acerto de 83%. Bao et al. (2017) elabora sobre a criação de um *framework* para predição de ativos financeiros utilizando *deep learning.* O modelo proposto foi treinado com período de avaliação de oito anos e *backtesting*

realizado em dados amostrais de seis anos, atingindo ganhos financeiros de até 63% na China e 45% na India. Um resumo das contribuições dos trabalhos citados é apresentado na tabela 1.

Tabela 1 – Trabalhos correlacionados e contribuições

Trabalho analisado	**Contribuição**
Nametala (2017): Construção de um Robô Investidor baseado em Redes Neurais Artificiais e Preditores Econométricos	Trabalha o conceito de redes neurais artificiais e o uso de diversos indicadores econométricos, envolvendo pré-processamento de dados, validação, *backtesting* e análise de performance preditiva.
Silva (2018): THALER - Um protótipo de robô investidor utilizando análise técnica e máquinas de vetores de suporte	Em seu sistema utiliza SVM com indicadores de análise técnica, descrevendo as etapas de treinamento e modelagem do sistema assim como a verificação e criação de estratégias de negociação baseadas também no retorno financeiro, a partir das decisões tomadas pelo algoritmo.
Lu (2016): A Machine Learning Approach to Automated Trading	Utiliza combinação de diversos algoritmos de *machine learning*, como regressão logística e SVM, além de técnicas de otimização identificando que podem melhorar consideravelmente a performance dos algoritmos preditores.
Nair et al. (2010): A Stock market trend prediction system using a hybrid decision tree-neuro-fuzzy system	Utiliza Sistema complexo de predição com a combinação de redes neurais e árvores de decisão obtendo taxa de acerto de 83% no período analisado.
Bao et al. (2017): A deep learning framework for financial time series using stacked autoencoders and longshort term memory	O trabalho apresentado possui foco nos métodos de *deep learning* com preparação e execução do *backtesting* em períodos extensos de dados atingindo ganhos financeiros consideráveis.

Fonte: Elaborado pelo autor.

Os trabalhos revisados contribuíram de forma significativa para o entendimento do complexo campo da análise preditiva utilizando *machine learning*. Seus conceitos serão diretamente aplicados no desenvolvimento da metodologia, apresentada a partir do próximo capítulo, com destaque para o conceito de séries temporais abordado por Nametala (2017) e a utilização de SVM citado por Silva (2018).

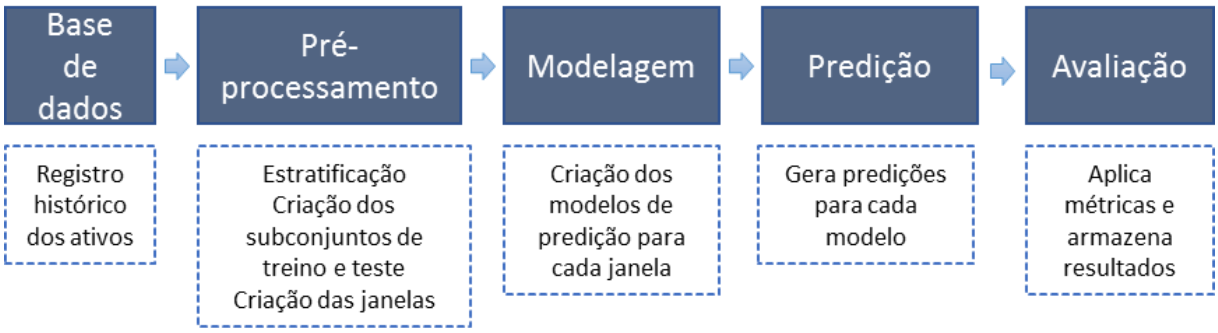
3 METODOLOGIA

A metodologia escolhida para realizar esse trabalho tem como base os objetivos iniciais e a bibliografia revisada no capítulo anterior. A solução proposta envolve a criação de um *software* que funcione como um robô preditor, analisando os dados históricos de cotação de ativos da B3 e fornecendo predições sobre o valor de suas cotações. Para o desenvolvimento foi escolhida a linguagem de programação *Python*, devido principalmente à biblioteca *sklearn* que é amplamente utilizada no meio acadêmico para a modelagem em *machine learning*. A seguir serão apresentados em detalhes os aspectos considerados para a construção do sistema, sua utilização e análise dos resultados obtidos.

3.1 Visão geral

O fluxo geral de funcionamento do sistema pode ser dividido nas etapas apresentadas na figura 11.

Figura 11 – Visão geral do sistema



Fonte: Elaborado pelo autor.

Na primeira etapa, cabe a verificação de integridade dos dados da base armazenados em arquivos de formato CSV (*Comma-Separated Values*). Os dados são lidos e pré-processados, de forma que fiquem prontos a serem modelados pelos algoritmos KNN, RFR, SVR e posteriormente utilizados no *stacking* com *XGBoost*.

Após gerados os modelos, são efetuadas as predições para o próximo período e armazenadas em arquivos também em formato CSV. Por fim, na etapa de avaliação, os arquivos com os resultados da predição são avaliados e verificada a performance dos modelos gerados.

3.2 Lista de materiais

Considerando que o projeto é constituído fundamentalmente de um *software*, a lista de materiais utilizados compreende:

a) computador com suporte a sistemas operacionais 64 bits Linux, Windows ou Mac;
b) ferramenta de edição de texto;
c) *python* versão 3.7.9 instalado e configurado;
d) bibliotecas do *python*, seguidas de suas versões: *pandas* 1.1.2, *numpy* 1.18.5, *matplotlib* 3.3.2, *sklearn* 0.23.2 e *xgboost* 1.2.1.

Quanto às bibliotecas utilizadas, cabe ressaltar que foram utilizadas também outras, porém não citadas por já serem nativas da instalação do *python* na versão 3.7.9.

3.3 Arquitetura da solução

Os blocos apresentados na visão geral do sistema indicam as principais etapas para operação do robô preditor, que foram incrementadas e são apresentadas nos itens a seguir.

3.3.1 Escolha dos ativos e construção da base de dados

A escolha dos ativos para análise teve como critério a participação dos mesmos na composição do índice Bovespa. Por esse motivo, foram escolhidos ITUB4, PETR4 e WEGE3, empresas com grande volume de negociações na B3 e colocadas entre as maiores empresas da bolsa brasileira. Dados históricos das operações foram consultados através do site *Yahoo* Finanças (2020), de onde foi possível obter registros que datam de 21/12/2000 até 30/09/2020, escolhida como data de corte para utilização no treinamento e teste dos modelos. Adicionalmente, o

mês de outubro de 2020 foi escolhido como período de validação para a execução do robô diariamente e verificação da sua performance, simulando o uso real. A tabela 2 apresenta a distribuição dos períodos quanto a sua utilização.

Tabela 2 – Períodos de operação quanto a sua finalidade

Período	Finalidade
21/12/2000 a 30/09/2020	Treino e testes
01/10/2020 a 31/10/2020	Validação

Fonte: Elaborado pelo autor.

Também foram obtidos registros referentes ao índice IBOV, para compor os dados dos três ativos, e ao preço do barril de petróleo cru para compor os dados de PETR4. O objetivo é que eles sirvam de análise para identificar possível correlação com os ativos da B3 que são os focos do estudo. Todos os dados foram obtidos com base no valor de fechamento diário e armazenados em arquivos com formato CSV, que podem ser lidos através da biblioteca *pandas* e manipulados dentro do *software*.

O período dos dados obtidos é diário, o que significa que cada registro da tabela corresponde a um dia de comportamento do ativo com informações sobre o valor de abertura, valor de fechamento e volume de negociação. Posteriormente os dados serão acrescidos dos valores de fechamento dos índices citados como possível correlação na análise, para identificar se a adição destes contribui para a performance preditiva dos modelos. A tabela 3 apresenta uma amostra da estrutura de dados da tabela do ativo WEGE3, com a inclusão do IBOV.

Tabela 3 – Registros WEGE3 com inclusão do IBOV

index	data	IBOV	volume	WEGE3
4849	01/09/2020	102168	56182800	24,34
4850	02/09/2020	101911	25170900	24,29
4851	03/09/2020	100721	81001700	24,88
4852	04/09/2020	101242	41679900	24,96
4853	08/09/2020	100050	29238000	24,40

Fonte: Elaborado pelo autor.

A variável *index* é um identificador sequencial de cada registro, para fins de manipulação dados no *software.* A coluna de data traz o dia das cotações analisadas, sendo importante reforçar que as cotações se referem ao dia especificado nessa coluna, e no caso do ativo ou do índice não possuir valor para a

data em questão é considerado o último valor válido de cotação. A coluna IBOV, como o nome indica, refere-se ao valor de fechamento do IBOV, enquanto volume e WEGE3 são os dados únicos de cotação do ativo analisado.

3.3.2 Validação dos dados

Os dados lidos do arquivo CSV são analisados quanto a sua integridade de valores. Datas que correspondem a feriados possuem valores nulos e por isso foram removidos das tabelas. A mesma ação foi tomada para os dias em que um ativo não operou. Também foi verificado se haviam campos com conteúdo inválido ou fora do *range* aceitável para aquele tipo de dado, não sendo encontradas ocorrências nos dados utilizados.

3.3.3 Períodos de análise

Conforme revisado na bibliografia, a análise de ativos da B3 trata da predição de valores futuros de uma série temporal. Com base nesse conceito, foram estabelecidas nove janelas temporais que serão utilizadas para construção dos modelos de predição, conforme pode ser observado na tabela 4.

Tabela 4 – Janelas de cotações

Período	**Tamanho da janela**
x1	1 cotação
x2	2 cotações
x3	3 cotações
x5	5 cotações
x10	10 cotações
x20	20 cotações
x30	30 cotações
x50	50 cotações
x80	80 cotações

Fonte: Elaborado pelo autor.

Cada cotação representa um dia de comportamento do ativo, dessa forma quando é utilizada a janela x80 estão sendo observados os últimos oitenta dias de cotação do ativo para se construir a previsão para o valor de fechamento no próximo período. Optou-se por limitar o número em oitenta devido à complexidade

computacional da geração dos modelos, que consomem uma grande quantidade de tempo para conclusão. Além disso, não foi encontrada melhora significativa na performance para janelas maiores durante os testes que justificassem a modelagem em mais tamanhos.

3.3.4 Pré-processamento

Antes de serem submetidos aos algoritmos de modelagem os dados são pré-processados. Segundo revisado na bibliografia, os dados devem ser separados em um conjunto de treino e um conjunto de este. Para este trabalho optou-se pela divisão de 80% para treino e 20% para teste. A divisão dos dados foi feita através da técnica de estratificação, de modo que os dois conjuntos mantenham a representatividade do total. A coluna *index* serve de referência para a montagem posterior das janelas, já que a estratificação monta os subconjuntos com amostras aleatórias e posteriormente é preciso recuperar a sequência inicial para montar os períodos anteriores. Após essa etapa, são eliminados os campos que não serão considerados na predição, como o identificador sequencial dos registros e a data. A tabela 5 apresenta como ficam os registros após a remoção dos dados não utilizados.

Tabela 5 – Dados do ativo WEGE3 considerados na predição.

IBOV	volume	WEGE3
102168	56182800	24,34
101911	25170900	24,29
100721	81001700	24,88
101242	41679900	24,96
100050	29238000	24,40

Fonte: Elaborado pelo autor.

Após esse passo, para cada registro da tabela, os dados são rearranjados em novos registros de acordo com a janela de dados estipulada para a geração do modelo, de forma que os últimos *n* registros sejam considerados um único valor de *feature*. O valor do *target* continua o mesmo, já que a única mudança é que registros anteriores são adicionados ao vetor da *feature.*

Na aplicação, essa conversão é feita para todos os registros da base. Além de acumulados, os registros precisam ser redimensionados antes de serem modelados, já que os algoritmos utilizados precisam que os dados tenham a mesma

dimensionalidade para todos os tamanhos de janela. A última etapa antes da modelagem trata da aplicação do *scaler*, que busca normalizar as *features* para que estejam compreendidas entre o *range* de valores 0 (zero) e 1 (um).

3.3.5 Modelagem em *machine learning*

A tarefa de predição caso de estudo desse trabalho pode ser analisada sob a ótica de algoritmos de regressão ou classificação. Optou-se pela utilização de algoritmos de regressão pelo objetivo de se estimar o valor de fechamento que o ativo irá assumir no próximo período. Para a construção dos modelos de predição optou-se pelo uso de três dos algoritmos de regressão revisados: KNN, SVR e RFR.

Adicionalmente, também é utilizada a técnica de *stacking* para a construção de uma segunda camada de predição, que utiliza o algoritmo *XGBoost*. Este é alimentado pelos modelos gerados nos três algoritmos citados anteriormente. Considerando que foram definidas no item 3.2.3 nove janelas de predição distintas e os quatro modelos baseados nos algoritmos definidos, cada modelagem produz o total de trinta e seis modelos de predição distintos. A organização dos modelos de predição pode ser observada na tabela 6.

Tabela 6 – Modelos de predição gerados pelo sistema

Modelos de predição			
KNN	**SVR**	**RFR**	***Stacking* (XGB)**
knn_x1	svr_x1	rfr_x1	stacked_x1
knn_x2	svr_x2	rfr_x2	stacked_x2
knn_x3	svr_x3	rfr_x3	stacked_x3
knn_x5	svr_x5	rfr_x5	stacked_x5
knn_x10	svr_x10	rfr_x10	stacked_x10
knn_x20	svr_x20	rfr_x20	stacked_x20
knn_x30	svr_x30	rfr_x30	stacked_x30
knn_x50	svr_x50	rfr_x50	stacked_x50
knn_x80	svr_x80	rfr_x80	stacked_x80

Fonte: Elaborado pelo autor.

Acrescenta-se ainda uma definição referente à periodicidade de treinamento dos modelos, sendo criadas três classes: treino diário, semanal e mensal. Os modelos com treino diário serão treinados todos os dias após o encerramento da

operação dos ativos analisados, contendo os dados das cotações fechadas do dia atual para prever o valor de fechamento no próximo período. Os modelos semanal e mensal terão os dados atualizados para realizar as predições, porém não serão treinados novamente com a inclusão desses valores até completarem a data de corte.

É importante ressaltar que a data da geração do modelo implica sempre que as predições realizadas serão válidas para a próxima data de operação do ativo na B3. Como o período de validação inicia em 01/10/2020, o modelo diário inicial deve ser gerado com as informações de fechamento do dia 30/09/2020 a fim de que gere as predições para o dia posterior. No caso dos modelos semanal e mensal, conforme citado, o que muda é que o modelo não será gerado com a mesma frequência, somente será atualizada a base de dados com a última data de fechamento e realizadas as predições para o próximo período. A tabela 7 apresenta a organização para a geração dos modelos e suas predições de acordo com a periodicidade do treinamento.

Tabela 7 – Modelos quanto a periodicidade de treinamento

Período de treino	**Datas de geração**
Diário	Dias de operação da B3 entre 30/09/2020 e 29/10/2020, inclusivo
Semanal	30/09/2020, 07/10/2020, 14/10/2020, 21/10/2020 e 28/10/2020
Mensal	30/09/2020

Fonte: Elaborado pelo autor.

3.3.6 Métricas de análise

Para verificação da performance preditiva dos modelos será utilizada a métrica RMSE. Como visto na bibliografia, algoritmos de regressão não possuem cálculo nativo da acurácia justamente por não preverem uma resposta binária, mas no caso específico desta pesquisa o seu uso pode trazer um dado bem relevante que indica se o algoritmo previu a subida ou a descida no valor de cotação.

Optou-se por utilizar uma métrica customizada baseada no movimento do ativo, cujos cenários são apresentados na tabela 8, considerando *vpred* como o valor predito pelo robô, *vlast* o valor da cotação no último período e *vreal* o valor real que o ativo assumiu.

Tabela 8 – Cenários para cálculo da acurácia customizada do sistema

Cenário	**Resultado**
(vpred > vlast) E (vreal > vlast)	Acerto
(vpred > vlast) E (vreal < vlast)	Erro
(vpred < vlast) E (vreal > vlast)	Erro
(vpred < vlast) E (vreal < vlast)	Acerto

Fonte: Elaborado pelo autor.

A acurácia customizada, denominada de ACC para referência posterior, é determinada pela taxa de acertos em relação ao número total de predições realizadas.

Serão considerados satisfatórios valores de RMSE que representarem até 2% do valor de cotação do ativo, por ser uma variação normal ocorrida durante a operação dos ativos na B3, conforme revisado na bibliografia. Para ACC, considera-se satisfatórias taxas de acerto superiores a 60%. Esse valor foi estipulado conforme os trabalhos correlatos revisados na bibliografia. Nametala (2017), obteve 58,3% de acerto nas predições realizadas, enquanto Lu (2016), 60%. Serão considerados resultados insatisfatórios de ACC valores abaixo de 60%.

3.3.7 Configuração do sistema

O sistema possui a capacidade de automatizar a sua operação através de parâmetros pré-determinados. Essa configuração foi criada em um arquivo no formato JSON, conforme apresentado na figura 12.

Figura 12 – Arquivo de configuração do software.

```
{
    "ativo" : "WEGE3",
    "debug" : true,
    "periodo" : "1d",
    "window" : [1, 2, 3, 5, 10, 20, 30, 50, 80],
    "operationstartdate" : "10/01/2020",
    "operationenddate" : "10/30/2020"
}
```

Fonte: Elaborado pelo autor.

De forma básica o que o arquivo de configuração está especificando é o ativo que será analisado, se terá saída de informações em modo *debug*, o período de análise (no caso desta pesquisa período diário), o tamanho da janelas de análise, a data inicial de validação e a data final de validação.

3.3.8 Logs de operação

A execução automática do robô predito produz uma série de informações desde as etapas iniciais até as predições, que incluem os parâmetros utilizados pelos modelos de predição, as métricas de avaliação e o resultado efetivo das predições. Para registro e posterior avaliação dessas informações foi criado um arquivo de texto que é gravado constantemente durante a execução da aplicação e armazena todas essas informações para conferência.

3.3.9 Avaliação dos resultados

A análise das métricas de avaliação para cada modelo gerado, bem como a comparação desses valores com os demais modelos apresentados, para cada ativo e cada janela temporal representa a análise de resultados deste trabalho. Além do período de teste, como citado no item 3.3.1, também é efetuado um período de validação no mês de outubro de 2020. As métricas para esse último período de também são comparadas com as métricas do período de teste para que seja possível estimar o funcionamento dos modelos de predição depois que foram gerados, e assim verificada a sua performance em uso real.

4 RESULTADOS

No presente capítulo serão apresentados os resultados obtidos através da análise de dados das cotações da bolsa de valores para os ativos ITUB4, PETR4 e WEGE3, para os períodos de teste e validação estipulados na metodologia.

4.1 Construção dos modelos de predição

A modelagem do sistema no período de treinamento e testes é realizada para cada ativo estipulado com a data de corte de 30/09/2020. Foi avaliada a performance para cada um dos modelos gerados considerando as métricas RMSE e acurácia customizada (ACC) conforme apresentado no item 3.3.6. Para cada ativo, verificou-se a influência do índice IBOV e do preço do barril de petróleo cru (no caso de PETR4) nos dados analisados.

4.1.1 Modelagem para ITUB4

O total de dias operados no período de modelagem para o ativo ITUB4 corresponde a 4855 dias. Considerando a divisão de 80% para treino e 20% para teste, o período de teste para qual foram geradas as métricas equivale a 971 dias. As tabelas 9, 10, 11 e 12 apresentam os resultados de cada modelo. A coluna ITUB4 traz os resultados quando considerados somente os valores de fechamento do ITUB4 para análise, enquanto a coluna +IBOV traz as métricas quando considerados adicionalmente aos dados de fechamento do ativo também os dados de fechamento do índice IBOV.

Tabela 9 – Resultados da modelagem com KNN para ITUB4

(continua)

Modelo	ITUB4	+IBOV	ITUB4	+IBOV
	RMSE		ACC	
knn_x1	0,373	0,393	0,512	0,550
knn_x2	0,381	0,422	0,543	0,551
knn_x3	0,394	0,440	0,526	0,551
knn_x5	0,422	0,433	0,522	0,579
knn_x10	0,485	0,424	0,547	0,645

(conclusão)

Modelo	ITUB4	+IBOV	ITUB4	+IBOV
	RMSE		ACC	
knn_x20	0,375	0,337	0,658	0,695
knn_x30	0,339	0,319	0,687	0,712
knn_x50	0,331	0,306	0,710	0,718
knn_x80	0,308	0,305	0,738	0,735
média	0,401	0,375	0,604	0,637

Fonte: Elaborado pelo autor.

Tabela 10 – Resultados da modelagem com SVR para ITUB4

Modelo	ITUB4	+IBOV	ITUB4	+IBOV
	RMSE		ACC	
svr_x1	0,391	0,363	0,499	0,493
svr_x2	0,393	0,363	0,496	0,502
svr_x3	0,389	0,363	0,505	0,495
svr_x5	0,393	0,362	0,513	0,510
svr_x10	0,392	0,360	0,514	0,522
svr_x20	0,389	0,359	0,513	0,532
svr_x30	0,388	0,367	0,504	0,529
svr_x50	0,396	0,375	0,508	0,534
svr_x80	0,396	0,375	0,508	0,525
média	0,391	0,365	0,507	0,516

Fonte: Elaborado pelo autor.

Tabela 11 – Resultados da modelagem com RFR para ITUB4

Modelo	ITUB4	+IBOV	ITUB4	+IBOV
	RMSE		ACC	
rfr_x1	0,464	0,470	0,547	0,545
rfr_x2	0,401	0,377	0,545	0,538
rfr_x3	0,416	0,377	0,529	0,527
rfr_x5	0,414	0,396	0,531	0,548
rfr_x10	0,433	0,404	0,540	0,546
rfr_x20	0,451	0,418	0,560	0,565
rfr_x30	0,480	0,417	0,573	0,577
rfr_x50	0,482	0,430	0,571	0,590
rfr_x80	0,491	0,436	0,570	0,590
média	0,448	0,414	0,552	0,558

Fonte: Elaborado pelo autor.

Tabela 12 – Resultados da modelagem com modelo *stacked* para ITUB4

Modelo	ITUB4	+IBOV	ITUB4	+IBOV
	RMSE		ACC	
stacked_x1	0,417	0,403	0,538	0,507
stacked_x2	0,416	0,398	0,526	0,513
stacked_x3	0,427	0,381	0,510	0,537
stacked_x5	0,410	0,396	0,514	0,509
stacked_x10	0,423	0,376	0,523	0,536
stacked_x20	0,398	0,359	0,554	0,572
stacked_x30	0,379	0,364	0,584	0,625
stacked_x50	0,366	0,340	0,619	0,649
stacked_x80	0,357	0,348	0,653	0,659
média	0,399	0,373	0,558	0,567

Fonte: Elaborado pelo autor.

A inclusão do índice IBOV causou efeitos positivos nas métricas avaliadas, considerando todas as janelas. Analisando o modelo knn_x80, verifica-se que este apresentou os melhores resultados, atingindo RMSE de 0,305 e acurácia de 73,5% na previsão de subida ou descida na cotação do ativo. O valor de RMSE corresponde a 1,36% do valor de cotação do ativo ao fim do período, que foi R$ 22,50.

No caso do algoritmo SVR a taxa de acurácia mostra-se consideravelmente menor do que nos modelos com KNN, por outro lado, houve uma pequena queda no valor médio do RMSE. Os modelos com RFR apresentaram aumento no valor de RMSE comparado aos demais algoritmos. Sua acurácia também não obteve resultados próximos aos obtidos pelo algoritmo KNN.

O modelo *stacked*, por sua vez, também apresentou melhora após a inclusão do índice IBOV na modelagem dos dados. Os melhores resultados foram obtidos pelo modelo *stacked_x80*, que atingiu aproximadamente 65,9% de taxa de acerto e RMSE de 0,373. Apesar de ser um modelo mais complexo, os resultados não superaram a utilização do algoritmo KNN nos períodos de teste avaliados.

4.1.2 Modelagem para PETR4

Para o ativo PETR4 foram utilizados registros de 4858 dias de operação na B3, o que corresponde a um período de teste contendo 971 dias pela divisão

realizada na aplicação. As tabelas 13, 14, 15 e 16 apresentam os resultados das métricas colhidas nesse período para cada algoritmo utilizado. A coluna PETR4 traz os resultados quando considerados somente as informações de fechamento do próprio ativo. A coluna +IBOV já considera além dos dados do ativo também as cotações do índice. A coluna +PETROL inclui nos dados os valores de cotação do barril de petróleo cru, adicionalmente aos dados do ativo e do índice IBOV.

Tabela 13 – Resultados da modelagem com KNN para PETR4

Modelo	**PETR4**	**+IBOV**	**+PETROL**	**PETR4**	**+IBOV**	**+PETROL**
	RMSE			**ACC**		
knn_x1	0,598	0,606	0,626	0,535	0,557	0,603
knn_x2	0,609	0,636	0,664	0,529	0,583	0,585
knn_x3	0,658	0,676	0,659	0,526	0,572	0,600
knn_x5	0,697	0,694	0,625	0,529	0,606	0,633
knn_x10	0,775	0,612	0,513	0,568	0,673	0,727
knn_x20	0,592	0,516	0,485	0,674	0,728	0,726
knn_x30	0,550	0,494	0,473	0,709	0,727	0,73
knn_x50	0,486	0,487	0,494	0,731	0,739	0,715
knn_x80	0,525	0,520	0,520	0,739	0,736	0,733
média	0,610	0,582	0,562	0,616	0,658	0,672

Fonte: Elaborado pelo autor.

Tabela 14 – Resultados da modelagem com SVR para PETR4

Modelo	**PETR4**	**+IBOV**	**+PETROL**	**PETR4**	**+IBOV**	**+PETROL**
	RMSE			**ACC**		
svr_x1	0,605	0,605	0,605	0,510	0,510	0,518
svr_x2	0,603	0,604	0,605	0,533	0,533	0,527
svr_x3	0,608	0,609	0,610	0,526	0,515	0,520
svr_x5	0,604	0,605	0,605	0,523	0,530	0,518
svr_x10	0,613	0,614	0,615	0,510	0,532	0,541
svr_x20	0,604	0,605	0,608	0,501	0,516	0,510
svr_x30	0,612	0,613	0,616	0,496	0,533	0,528
svr_x50	0,603	0,606	0,616	0,516	0,536	0,545
svr_x80	0,613	0,621	0,628	0,503	0,516	0,505
média	0,607	0,609	0,612	0,513	0,525	0,523

Fonte: Elaborado pelo autor.

Tabela 15 – Resultados da modelagem com RFR para PETR4

Modelo	PETR4	+IBOV	+PETROL	PETR4	+IBOV	+PETROL
	RMSE			ACC		
rfr_x1	0,764	0,884	0,632	0,543	0,534	0,535
rfr_x2	0,634	0,652	0,664	0,519	0,538	0,556
rfr_x3	0,661	0,653	0,662	0,509	0,533	0,542
rfr_x5	0,675	0,706	0,696	0,509	0,545	0,530
rfr_x10	0,722	0,729	0,730	0,492	0,512	0,530
rfr_x20	0,746	0,743	0,747	0,512	0,531	0,534
rfr_x30	0,748	0,735	0,730	0,532	0,546	0,553
rfr_x50	0,765	0,756	0,734	0,529	0,557	0,565
rfr_x80	0,789	0,747	0,733	0,543	0,559	0,573
média	0,722	0,733	0,703	0,520	0,539	0,546

Fonte: Elaborado pelo autor.

Tabela 16 – Resultados da modelagem com modelo *stacked* para PETR4

Modelo	PETR4	+IBOV	+PETROL	PETR4	+IBOV	+PETROL
	RMSE			ACC		
stacked_x1	0,697	0,670	0,683	0,503	0,522	0,538
stacked_x2	0,670	0,684	0,683	0,520	0,541	0,544
stacked_x3	0,660	0,677	0,652	0,545	0,543	0,529
stacked_x5	0,684	0,670	0,665	0,525	0,541	0,530
stacked_x10	0,657	0,617	0,639	0,514	0,581	0,608
stacked_x20	0,625	0,597	0,568	0,593	0,624	0,633
stacked_x30	0,613	0,546	0,559	0,578	0,622	0,631
stacked_x50	0,589	0,595	0,601	0,638	0,647	0,634
stacked_x80	0,585	0,598	0,597	0,650	0,640	0,634
média	0,642	0,628	0,627	0,563	0,585	0,589

Fonte: Elaborado pelo autor.

Na modelagem com o algoritmo KNN a inclusão do IBOV e dos valores de cotação do barril de petróleo cru trouxeram melhora nos números, chegando ao interessante valor de 73,3% de ACC. O modelo knn_x30 apresentou o melhor resultado de RMSE, com o valor de 0,473. Esse valor, porém, corresponde a 2,41% do valor de cotação de PETR4 ao fim do período, que foi R$ 19,61.

O algoritmo SVR não apresentou melhora perceptível de performance com a inclusão dos dados de IBOV e petróleo na modelagem, pelo contrário, apresentou ligeira piora na métrica RMSE. Para a modelagem com RFR, apesar da inclusão dos

dados correlacionados ao ativo reduzirem os erros, o ativo PETR4 apresentou resultados bem abaixo dos demais algoritmos, atingindo valor médio de RMSE igual a 0,703.

O modelo *stacked* para o ativo também não trouxe bons resultados, sendo superado em grande margem pelo modelo KNN. De forma geral, os resultados obtidos pioraram consideravelmente quando comparados com os valores gerados para o ativo ITUB4.

4.1.3 Modelagem para WEGE3

O ativo WEGE3 apresenta um diferencial dos outros dois analisados quanto a quantidade de registros. Foram utilizados registros de 3403 dias de operação na B3, o que corresponde a um período de teste contendo 680 dias pela divisão realizada na aplicação, cerca de 300 dias menor do que os ativos que foram analisados anteriormente.

As tabelas 17, 18, 19 e 20 apresentam os resultados das métricas colhidas nesse período para cada modelo. Assim como nas tabelas anteriores, a coluna com o nome do ativo apresenta os resultados somente com os seus dados de cotação, enquanto a coluna +IBOV apresenta os dados com a inclusão do índice na modelagem.

Tabela 17 – Resultados da modelagem com KNN para WEGE3

Modelo	**WEGE3**	**+IBOV**	**WEGE3**	**+IBOV**
	RMSE		**ACC**	
knn_x1	0,478	0,584	0,552	0,561
knn_x2	0,519	0,528	0,545	0,580
knn_x3	0,565	0,578	0,532	0,554
knn_x5	0,568	0,565	0,530	0,568
knn_x10	0,507	0,495	0,570	0,631
knn_x20	0,545	0,477	0,581	0,691
knn_x30	0,536	0,415	0,619	0,716
knn_x50	0,430	0,475	0,679	0,725
knn_x80	0,435	0,442	0,681	0,753
média	0,509	0,507	0,589	0,642

Fonte: Elaborado pelo autor.

Tabela 18 – Resultados da modelagem com SVR para WEGE3

Modelo	WEGE3	+IBOV	WEGE3	+IBOV
	RMSE		ACC	
svr_x1	0,493	0,502	0,516	0,495
svr_x2	0,477	0,530	0,518	0,501
svr_x3	0,488	0,531	0,511	0,490
svr_x5	0,484	0,471	0,527	0,533
svr_x10	0,486	0,537	0,509	0,518
svr_x20	0,492	0,397	0,534	0,561
svr_x30	0,507	0,549	0,512	0,529
svr_x50	0,497	0,378	0,514	0,492
svr_x80	0,506	0,528	0,53	0,535
média	0,492	0,491	0,519	0,517

Fonte: Elaborado pelo autor.

Tabela 19 – Resultados da modelagem com RFR para WEGE3

Modelo	WEGE3	+IBOV	WEGE3	+IBOV
	RMSE		ACC	
rfr_x1	0,611	0,742	0,542	0,544
rfr_x2	0,525	0,542	0,531	0,546
rfr_x3	0,515	0,532	0,530	0,538
rfr_x5	0,513	0,556	0,537	0,556
rfr_x10	0,491	0,527	0,558	0,556
rfr_x20	0,525	0,561	0,549	0,559
rfr_x30	0,557	0,571	0,553	0,566
rfr_x50	0,569	0,537	0,563	0,585
rfr_x80	0,527	0,463	0,574	0,606
média	0,537	0,559	0,549	0,562

Fonte: Elaborado pelo autor.

Tabela 20 – Resultados da modelagem com modelo *stacked* WEGE3

(continua)

Modelo	WEGE3	+IBOV	WEGE3	+IBOV
	RMSE		ACC	
stacked_x1	0,541	0,607	0,537	0,539
stacked_x2	0,620	0,560	0,563	0,561
stacked_x3	0,583	0,575	0,519	0,533
stacked_x5	0,602	0,583	0,512	0,512

(conclusão)

Modelo	WEGE3	+IBOV	WEGE3	+IBOV
	RMSE		ACC	
stacked_x10	0,585	0,568	0,513	0,539
stacked_x20	0,535	0,526	0,505	0,562
stacked_x30	0,563	0,542	0,551	0,599
stacked_x50	0,479	0,505	0,575	0,633
stacked_x80	0,490	0,568	0,609	0,638
média	0,555	0,559	0,543	0,568

Fonte: Elaborado pelo autor.

Para o ativo WEGE3 não houve melhora significativa na métrica RMSE com a inclusão do IBOV, porém a métrica ACC teve um aumento considerável, atingindo o valor de 75,3% no modelo knn_x80. A modelagem utilizando SVR não apresentou alteração significativa nos resultados, analisando o resultado geral dos modelos. O modelo svr_x50 apresentou o melhor valor de RMSE, com 0,378, que corresponde a 0,58% do valor de cotação do ativo no fim de período, R$ 65,70. Conforme aconteceu com os outros ativos analisados, o melhor desempenho foi identificado na modelagem utilizando KNN.

4.1.4 Análise de performance no período de teste

De forma geral, avalia-se que a inclusão das cotações do índice IBOV e do barril de petróleo cru, no caso de PETR4, provocaram melhora significativa nas métricas avaliadas. Além disso, entre os algoritmos analisados, o KNN apresentou os melhores resultados de acurácia e RMSE durante o período de testes dentro os três ativos analisados, com exceção de alguns modelos em que a métrica RMSE apresentou valores menores nos modelos com SVR. Por fim, nota-se também que o aumento no tamanho da janela utilizada para modelagem traz considerável melhora nos resultados de cada algoritmo. Apesar da melhora, porém, o melhor valor de RMSE obtido para o ativo PETR4 representou 2,41% do valor de cotação do ativo no fim do período, caracterizando o cenário como não satisfatório.

4.2 Validação dos modelos construídos

Conforme definido no item 3.3.1, os modelos construídos foram utilizados para a realização de predições no mês de outubro de 2020. O período é constituído de vinte e um dias de operação, que vão do dia 01/10/2020 a 30/10/2020. Os modelos utilizados possuem a inclusão do IBOV e das cotações do barril de petróleo cru, no caso de PETR4, devido apresentarem esses melhor performance verificada no item 4.1. Assim como para a modelagem, foi avaliada a performance de cada modelo considerando as métricas RMSE e ACC conforme apresentado no item 3.3.6. Adicionalmente, serão apresentadas as métricas para os três períodos de treinamento estipulados no item 3.3.5.

4.2.1 Resultados do período de validação para ITUB4

As tabelas 21, 22, 23 e 24 apresentam os resultados do período de validação para o ativo ITUB4.

Tabela 21 – Resultados do modelo KNN para período de validação ITUB4

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
knn_x1	0,592	0,635	0,601	0,619	0,619	0,666
knn_x2	0,682	0,677	0,688	0,619	0,666	0,666
knn_x3	0,785	0,742	0,749	0,761	0,666	0,666
knn_x5	0,880	0,897	0,943	0,666	0,666	0,714
knn_x10	0,943	1,158	1,112	0,714	0,619	0,619
knn_x20	1,228	1,140	1,124	0,714	0,619	0,761
knn_x30	1,272	1,125	1,053	0,666	0,666	0,666
knn_x50	1,243	1,022	1,048	0,666	0,666	0,714
knn_x80	1,060	1,277	1,114	0,761	0,666	0,619
média	0,965	0,964	0,937	0,687	0,650	0,677

Fonte: Elaborado pelo autor.

Tabela 22 – Resultados do modelo SVR para período de validação ITUB4

(continua)

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		

(conclusão)

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
svr_x1	0,577	0,579	0,587	0,523	0,523	0,523
svr_x2	0,576	0,660	0,590	0,476	0,571	0,571
svr_x3	0,577	0,655	0,577	0,476	0,523	0,523
svr_x5	0,577	0,579	0,597	0,666	0,619	0,571
svr_x10	0,562	0,569	0,640	0,666	0,619	0,571
svr_x20	0,566	0,560	0,640	0,523	0,476	0,428
svr_x30	0,555	0,624	0,667	0,714	0,523	0,476
svr_x50	0,584	0,672	0,678	0,619	0,476	0,476
svr_x80	0,618	0,598	0,643	0,619	0,619	0,666
média	0,577	0,611	0,624	0,587	0,550	0,534

Fonte: Elaborado pelo autor.

Tabela 23 – Resultados do modelo RFR para período de validação ITUB4

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
rfr_x1	1,796	1,697	1,748	0,476	0,476	0,523
rfr_x2	0,872	0,852	0,740	0,428	0,428	0,571
rfr_x3	0,707	0,672	0,647	0,571	0,619	0,619
rfr_x5	0,820	0,832	0,816	0,523	0,476	0,428
rfr_x10	0,880	0,805	0,767	0,619	0,571	0,571
rfr_x20	1,009	0,868	0,881	0,571	0,523	0,476
rfr_x30	1,032	0,878	0,865	0,571	0,571	0,476
rfr_x50	0,910	0,815	0,774	0,571	0,523	0,523
rfr_x80	1,023	0,921	0,908	0,523	0,476	0,476
média	1,005	0,927	0,905	0,539	0,518	0,518

Fonte: Elaborado pelo autor.

Tabela 24 – Resultados do modelo *stacked* para período de validação ITUB4

(continua)

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
stacked_x1	0,680	1,081	0,942	0,619	0,523	0,523
stacked_x2	0,760	0,977	0,922	0,428	0,428	0,571
stacked_x3	1,129	0,851	0,694	0,476	0,476	0,571
stacked_x5	0,935	0,727	0,685	0,428	0,380	0,523

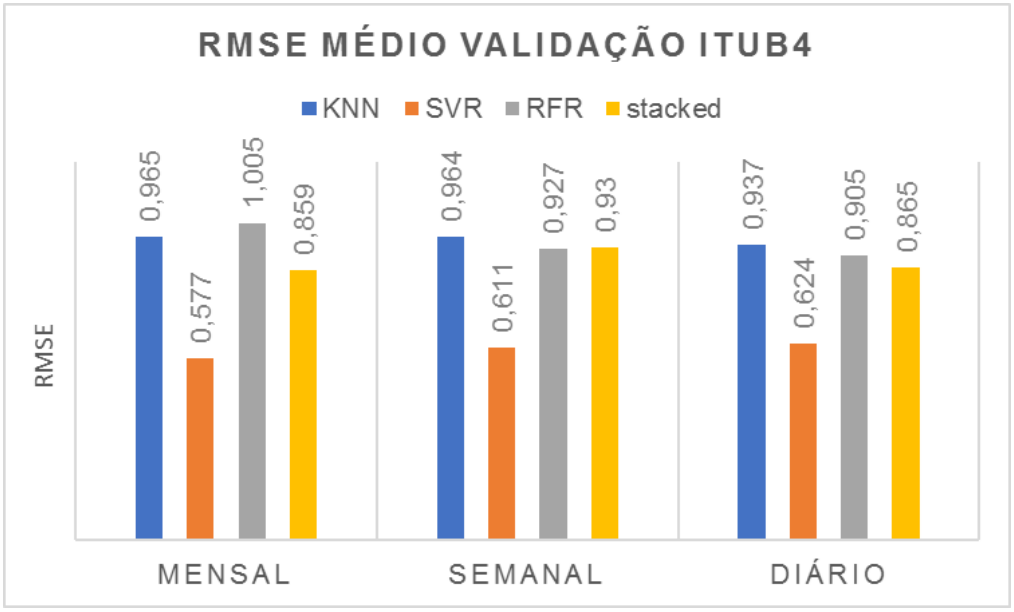
(conclusão)

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
stacked_x10	0,928	0,838	0,891	0,476	0,476	0,476
stacked_x20	0,891	0,937	0,876	0,523	0,428	0,619
stacked_x30	0,784	1,033	0,869	0,523	0,523	0,428
stacked_x50	0,762	0,960	0,888	0,571	0,476	0,619
stacked_x80	0,839	0,968	1,015	0,619	0,476	0,523
média	0,859	0,930	0,865	0,518	0,465	0,539

Fonte: Elaborado pelo autor.

Ao compararmos os resultados com o período de testes verifica-se piora significativa nos valores de RMSE, com os melhores resultados apresentados pelos modelos SVR. A figura 13 ilustra esse comportamento.

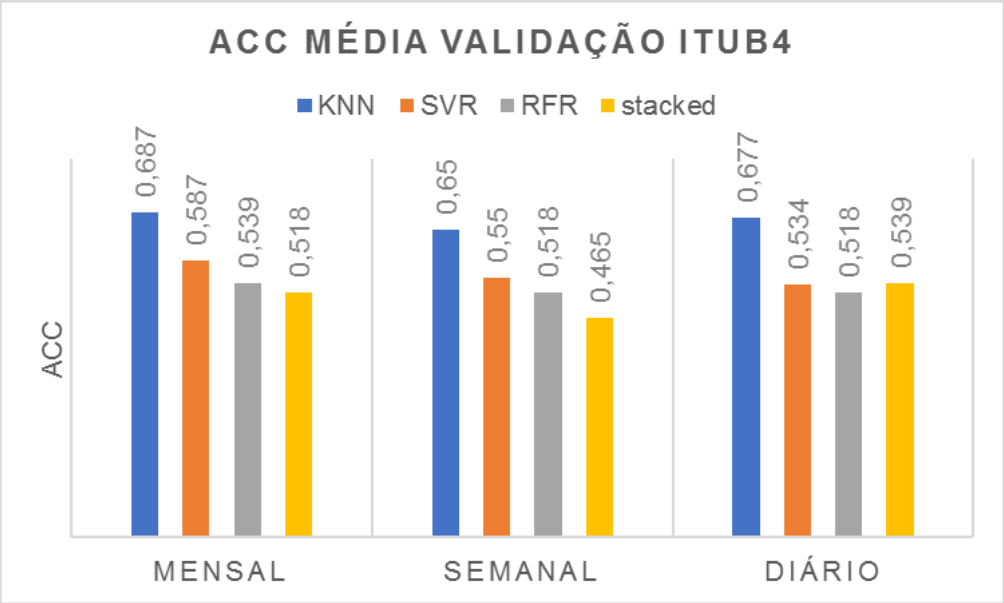
Figura 13 – Gráfico de RMSE médio para ITUB4 no período de validação



Fonte: Elaborado pelo autor.

O resultado difere dos apresentados durante o treinamento, em que o algoritmo KNN apresentou valores menores do que os demais. O melhor resultado obtido de RMSE médio foi 0,577, o que representa 2,45% do valor de cotação do ativo no fechamento do período, que foi R$ 23,57. Quanto ao resultado de acurácia (ACC), a modelagem em KNN manteve os melhores valores, seguido pelos modelos com SVR. Na figura 14 são apresentados os valores médios de ACC para cada período de treinamento.

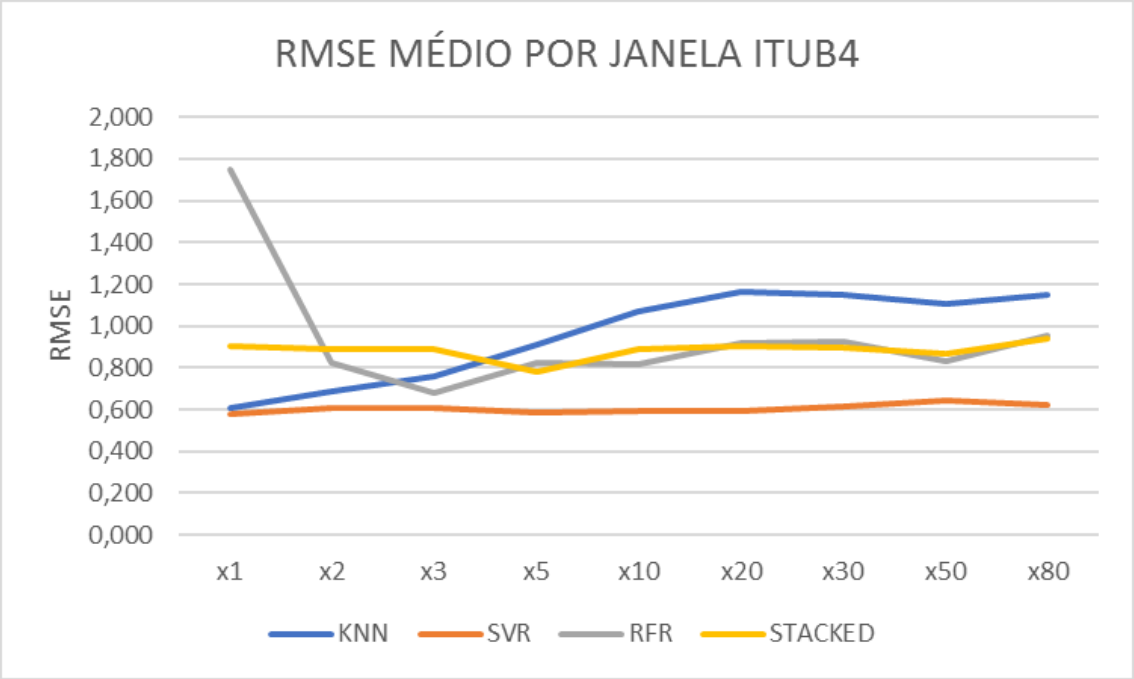
Figura 14 – Gráfico de ACC média para ITUB4 no período de validação



Fonte: Elaborado pelo autor.

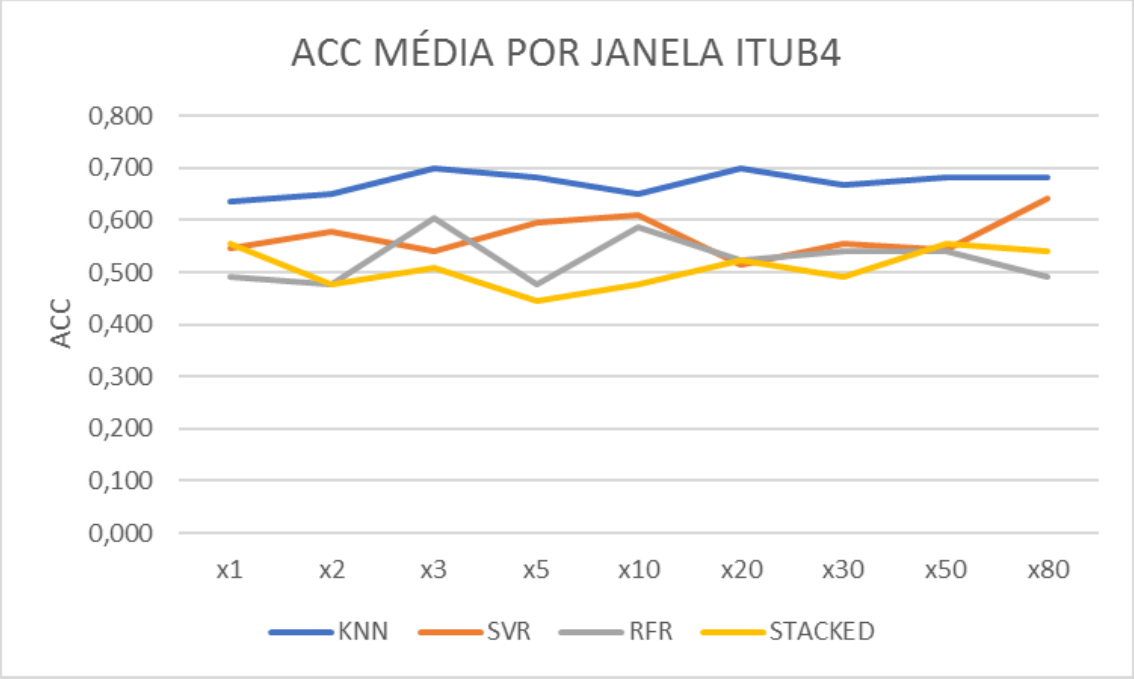
Os valores médios de RMSE e de ACC fornecem uma visão geral do desempenho dos modelos. Nas figuras 13 e 14, nota-se que a periodicidade de treinamento do robô preditor não afetou significativamente os resultados para o ativo ITUB4. Para complementar a análise, as figuras 15 e 16 apresentam os valores vistos sob a ótica do tamanho das janelas utilizadas, em que é possível verificar as que possuem melhor performance. Para isso os valores de cada janela foram agrupados por modelo, através do cálculo da média dos valores encontrados nos períodos diário, semanal e mensal. O objetivo desse agrupamento é fornecer uma visão geral do desempenho dos modelos de acordo com a variação nas janelas utilizadas para análise.

Figura 15 – Gráfico de RMSE médio por janela na validação de ITUB4



Fonte: Elaborado pelo autor.

Figura 16 – Gráfico de ACC média por janela na validação de ITUB4



Fonte: Elaborado pelo autor.

Conforme apresentado na figura 15, existe uma leve piora nos dados de RMSE para o algoritmo KNN conforme aumenta o tamanho da janela de predição. Os demais algoritmos, com exceção de RFR que apresentou um valor elevado de

erro na janela x1, o aumento do tamanho da janela tende a diminuir o valor da métrica. Já no caso da ACC, apresentada na figura 16, verifica-se que a tendência é de uma leve melhora na performance conforme aumenta-se o tamanho da janela.

4.2.2 Resultados do período de validação para PETR4

As tabelas 25, 26, 27 e 28 apresentam os resultados do período de validação para o ativo PETR4.

Tabela 25 – Resultados do modelo KNN para período de validação PETR4

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
knn_x1	0,958	0,705	0,735	0,428	0,380	0,285
knn_x2	0,980	0,733	0,694	0,428	0,523	0,619
knn_x3	1,031	0,826	0,756	0,428	0,523	0,571
knn_x5	1,029	0,964	0,637	0,428	0,666	0,714
knn_x10	2,200	1,252	1,146	0,380	0,571	0,619
knn_x20	2,706	1,763	1,104	0,380	0,285	0,619
knn_x30	2,470	1,016	0,700	0,380	0,571	0,666
knn_x50	1,582	0,914	0,746	0,380	0,666	0,714
knn_x80	1,935	1,074	0,799	0,380	0,666	0,714
média	1,655	1,027	0,813	0,401	0,539	0,613

Fonte: Elaborado pelo autor.

Tabela 26 – Resultados do modelo SVR para período de validação PETR4

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
svr_x1	0,544	0,543	0,545	0,380	0,380	0,476
svr_x2	0,542	0,539	0,536	0,428	0,523	0,714
svr_x3	0,541	0,541	0,543	0,619	0,714	0,666
svr_x5	0,550	0,554	0,548	0,523	0,476	0,619
svr_x10	0,535	0,534	0,531	0,619	0,619	0,666
svr_x20	0,548	0,605	0,576	0,523	0,476	0,571
svr_x30	0,543	0,732	0,507	0,428	0,333	0,666
svr_x50	0,788	0,887	0,753	0,380	0,571	0,761
svr_x80	0,989	0,768	0,862	0,380	0,571	0,619
média	0,620	0,634	0,600	0,476	0,518	0,640

Fonte: Elaborado pelo autor.

Tabela 27 – Resultados do modelo RFR para período de validação PETR4

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
rfr_x1	1,177	1,109	1,058	0,380	0,380	0,380
rfr_x2	1,134	1,078	1,018	0,428	0,380	0,380
rfr_x3	0,958	0,928	0,848	0,380	0,380	0,380
rfr_x5	0,883	0,855	0,803	0,428	0,428	0,476
rfr_x10	0,792	0,750	0,766	0,428	0,476	0,380
rfr_x20	0,816	0,821	0,795	0,428	0,476	0,476
rfr_x30	0,775	0,753	0,743	0,476	0,523	0,476
rfr_x50	0,785	0,779	0,707	0,571	0,523	0,571
rfr_x80	0,625	0,624	0,662	0,714	0,666	0,571
média	0,883	0,855	0,822	0,470	0,470	0,454

Fonte: Elaborado pelo autor.

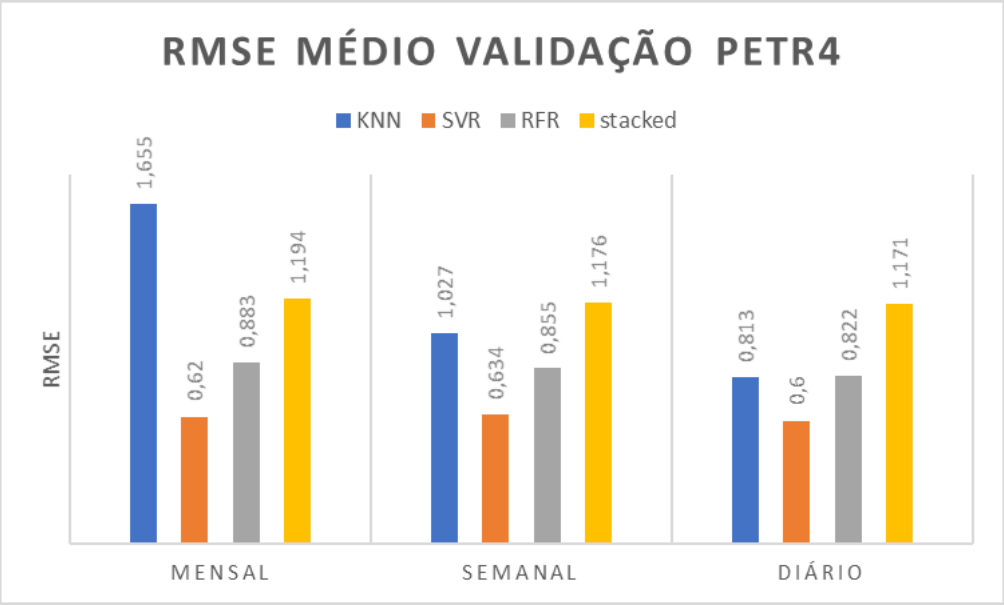
Tabela 28 – Resultados do modelo *stacked* para período de validação PETR4

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
stacked_x1	0,863	0,882	0,944	0,476	0,428	0,333
stacked_x2	0,673	0,911	1,045	0,380	0,666	0,523
stacked_x3	0,691	0,726	0,707	0,666	0,619	0,761
stacked_x5	0,603	0,597	0,631	0,476	0,714	0,476
stacked_x10	1,056	0,855	1,043	0,428	0,285	0,238
stacked_x20	1,181	1,583	1,419	0,380	0,380	0,380
stacked_x30	0,866	1,379	1,559	0,285	0,333	0,380
stacked_x50	2,124	1,862	1,584	0,380	0,380	0,333
stacked_x80	2,686	1,786	1,609	0,380	0,380	0,380
média	1,194	1,176	1,171	0,428	0,465	0,423

Fonte: Elaborado pelo autor.

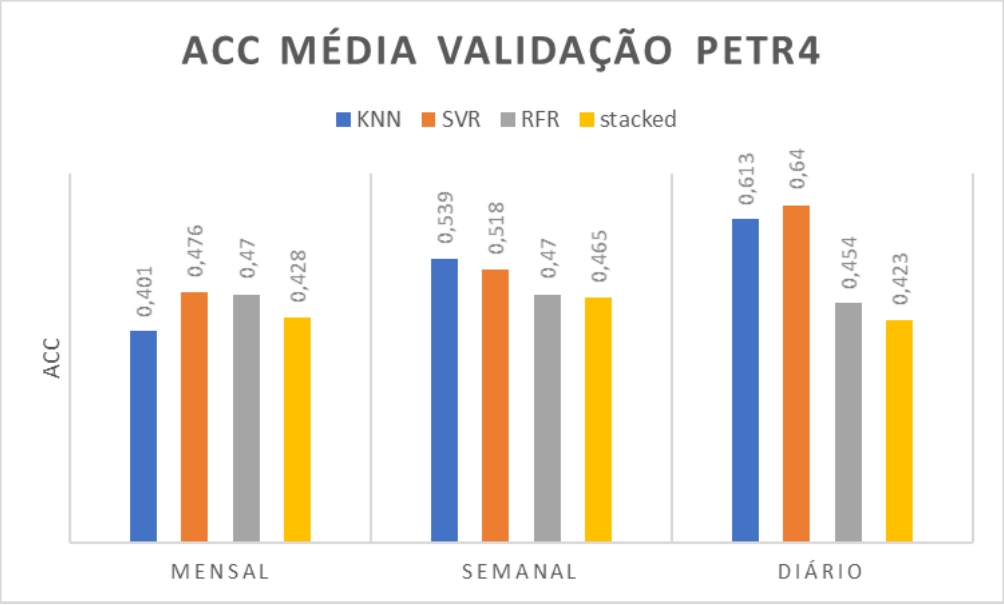
Assim como ocorreu para o ativo ITUB4 as métricas apresentam resultados inferiores na maioria dos modelos. O RMSE apresentou valores elevados de desvio e a ACC valores baixos. Dentre os resultados, porém, nota-se novamente uma boa performance do modelo SVR em comparação com os demais. Esse apresentou valor de RMSE consideravelmente menor do que os concorrentes e a melhor ACC, conforme apresentado nas figuras 17 e 18.

Figura 17 – Gráfico de RMSE médio para PETR4 no período de validação



Fonte: Elaborado pelo autor.

Figura 18 – Gráfico de ACC média para PETR4 no período de validação

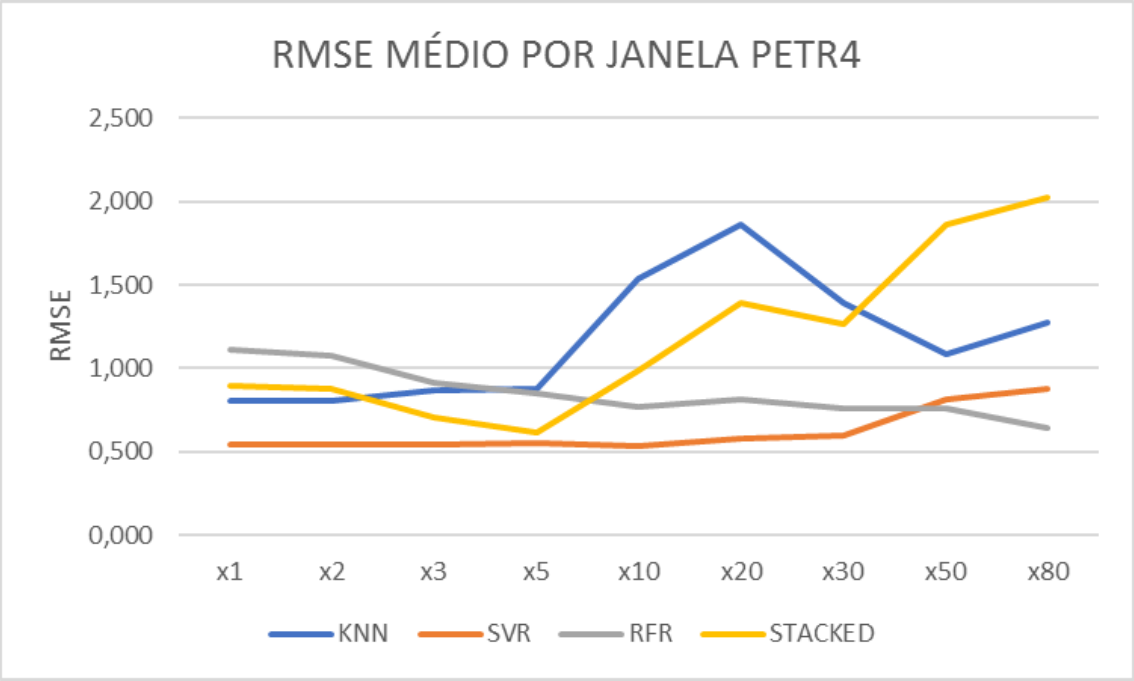


Fonte: Elaborado pelo autor.

O menor valor médio de RMSE obtido foi 0,6 e este representa 3,16% do valor de cotação do ativo ao fim do período, que foi R$ 19,01. Outro dado importante que ficou evidente no ativo ITUB4 foi a melhora significativa nos dados de acordo com a periodicidade de treino dos modelos. Com treinamento diário, foram encontrados os melhores resultados das métricas de avaliação por uma grande

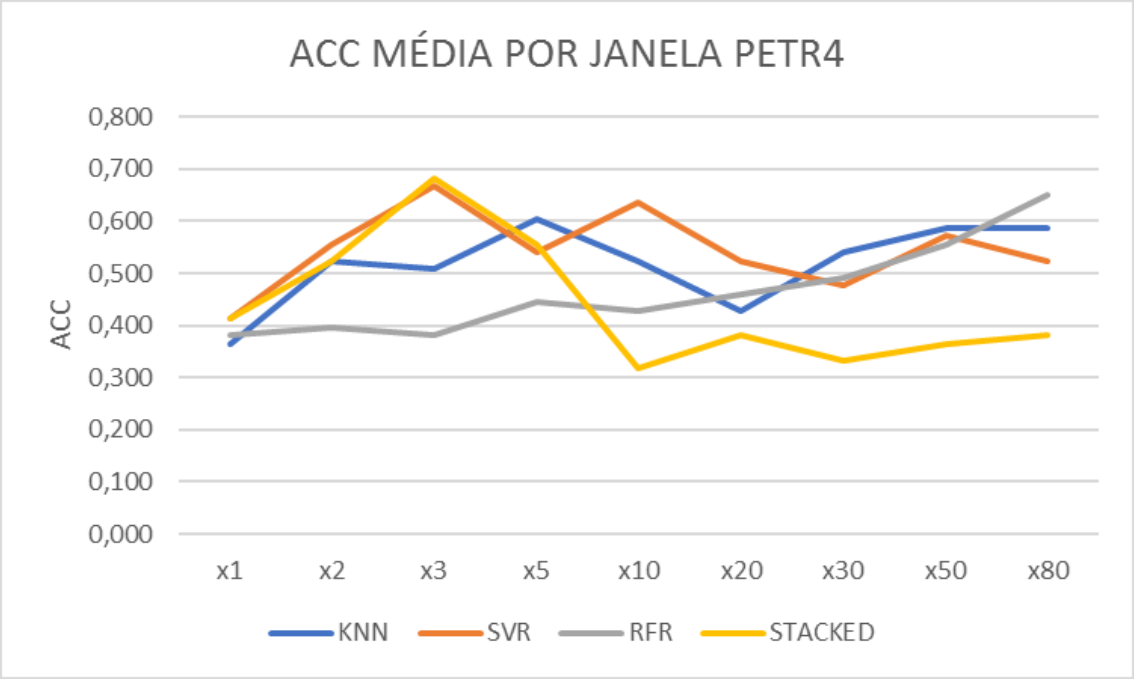
margem. De forma complementar, as figuras 19 e 20 apresentam a variação das métricas de acordo com o tamanho da janela.

Figura 19 – Gráfico de RMSE médio por janela na validação de PETR4



Fonte: Elaborado pelo autor.

Figura 20 – Gráfico de ACC média por janela na validação de PETR4



Fonte: Elaborado pelo autor.

Os resultados obtidos para as diferentes janelas de observação indicam grande variação da performance. Enquanto os valores de RMSE tendem a piorar com o aumento do tamanho da janela, a ACC apresenta leve melhora, mas contrastada por pontos de bom desempenho nas janelas x3 e x10 do algoritmo SVR. O modelo *stacked* também apresentou um ótimo valor de ACC no período x3, porém os demais resultados continuaram abaixo dos outros modelos conforme ocorreu também para o ativo ITUB4.

4.2.3 Resultados do período de validação para WEGE3

As tabelas 29, 30, 31 e 32 apresentam os resultados do período de validação para o ativo WEGE3.

Tabela 29 – Resultados do modelo KNN para período de validação WEGE3

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
knn_x1	11,073	5,545	4,109	0,285	0,333	0,333
knn_x2	11,123	5,967	4,731	0,285	0,380	0,333
knn_x3	11,616	6,525	4,890	0,285	0,333	0,285
knn_x5	12,392	7,669	6,214	0,285	0,333	0,380
knn_x10	12,052	9,380	6,672	0,285	0,333	0,380
knn_x20	13,675	10,238	7,585	0,285	0,285	0,285
knn_x30	15,493	10,027	8,693	0,285	0,285	0,285
knn_x50	15,769	11,370	9,440	0,285	0,285	0,285
knn_x80	16,198	11,465	10,212	0,285	0,285	0,285
média	13,266	8,687	6,950	0,285	0,317	0,317

Fonte: Elaborado pelo autor.

Tabela 30 – Resultados do modelo SVR para período de validação WEGE3

(continua)

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
svr_x1	2,351	4,914	5,850	0,761	0,761	0,761
svr_x2	2,297	3,696	5,169	0,761	0,761	0,761
svr_x3	2,218	3,706	4,021	0,809	0,666	0,714
svr_x5	2,232	2,667	3,662	0,666	0,714	0,714
svr_x10	2,222	2,667	2,791	0,761	0,809	0,809

(conclusão)

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
svr_x20	2,256	2,455	2,640	0,571	0,666	0,571
svr_x30	2,331	2,245	2,438	0,428	0,619	0,714
svr_x50	2,305	2,134	2,277	0,571	0,761	0,761
svr_x80	2,392	2,359	2,208	0,523	0,666	0,761
média	2,289	2,983	3,451	0,650	0,714	0,730

Fonte: Elaborado pelo autor.

Tabela 31 – Resultados do modelo RFR para período de validação WEGE3

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
rfr_x1	17,297	16,337	16,057	0,285	0,285	0,285
rfr_x2	14,163	13,870	13,725	0,285	0,285	0,285
rfr_x3	13,341	13,019	13,142	0,285	0,285	0,285
rfr_x5	14,922	14,729	14,583	0,285	0,285	0,285
rfr_x10	14,524	15,089	14,642	0,285	0,285	0,285
rfr_x20	13,973	14,133	14,248	0,285	0,285	0,285
rfr_x30	13,291	13,733	13,870	0,285	0,285	0,285
rfr_x50	14,368	15,401	15,552	0,285	0,285	0,285
rfr_x80	25,430	24,817	24,170	0,285	0,285	0,285
média	15,701	15,681	15,554	0,285	0,285	0,285

Fonte: Elaborado pelo autor.

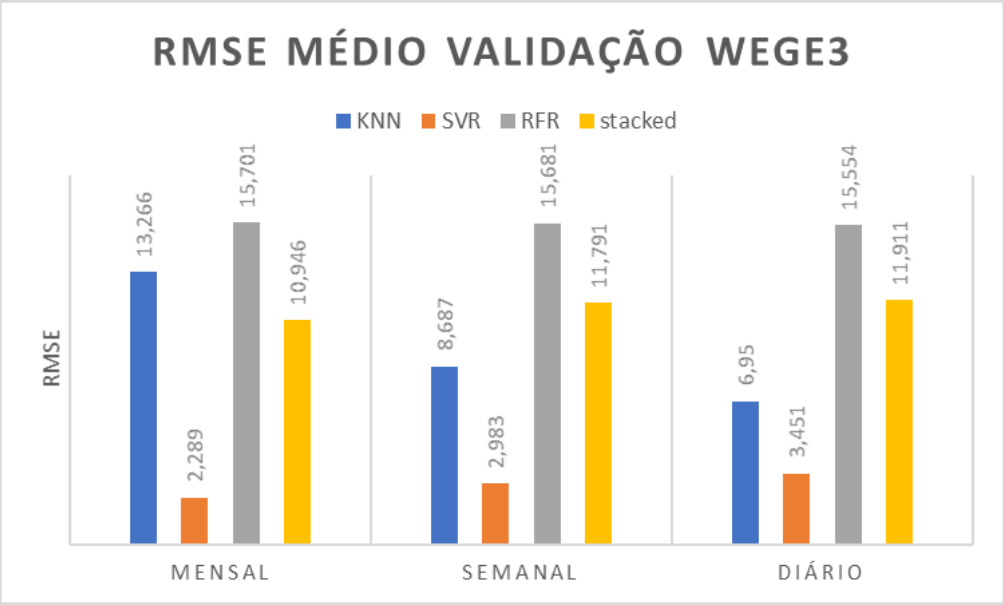
Tabela 32 – Resultados do modelo *stacked* para período de validação WEGE3

Modelo	Mensal	Semanal	Diário	Mensal	Semanal	Diário
	RMSE			ACC		
stacked_x1	11,086	11,225	11,285	0,380	0,380	0,380
stacked_x2	11,082	11,488	11,576	0,380	0,380	0,380
stacked_x3	10,970	11,610	11,252	0,380	0,380	0,380
stacked_x5	11,893	12,048	11,844	0,380	0,380	0,333
stacked_x10	10,864	11,328	11,264	0,380	0,380	0,380
stacked_x20	10,873	11,405	11,606	0,380	0,380	0,380
stacked_x30	11,754	12,019	11,818	0,333	0,333	0,380
stacked_x50	10,441	11,813	12,095	0,380	0,380	0,380
stacked_x80	9,550	13,185	14,461	0,380	0,380	0,380
média	10,946	11,791	11,911	0,375	0,375	0,375

Fonte: Elaborado pelo autor.

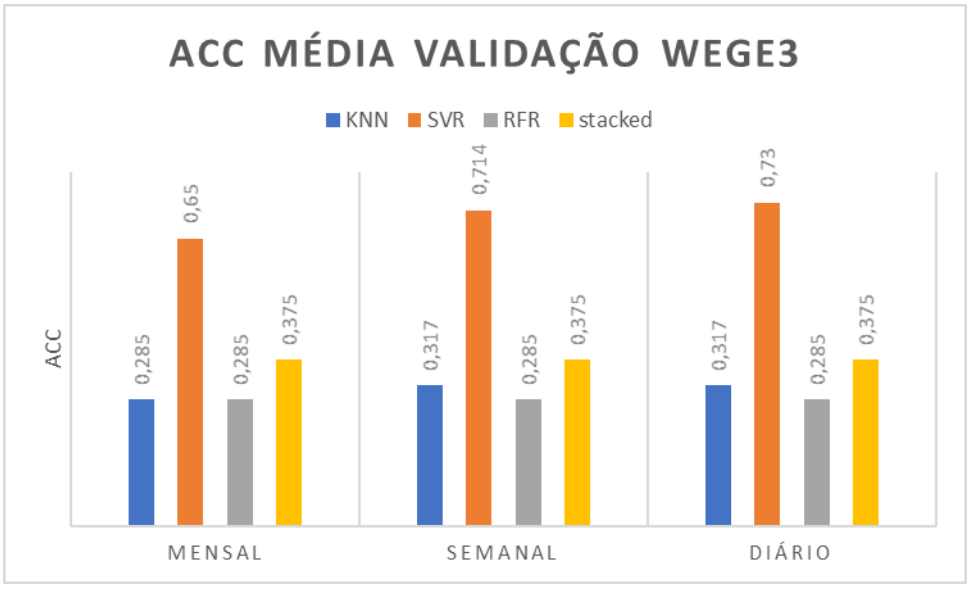
O algoritmo SVR, cujos valores de RMSE foram os menores dos modelos gerados, apresentou valor mais de quatro vezes maior ao encontrado no período de testes do modelo. Dentre todos os modelos, porém, o SVR foi o que se saiu melhor para o ativo com larga margem de diferença, conforme apresentado nas figuras 21 e 22.

Figura 21 – Gráfico de RMSE médio para WEGE3 no período de validação



Fonte: Elaborado pelo autor.

Figura 22 – Gráfico de ACC média para WEGE3 no período de validação

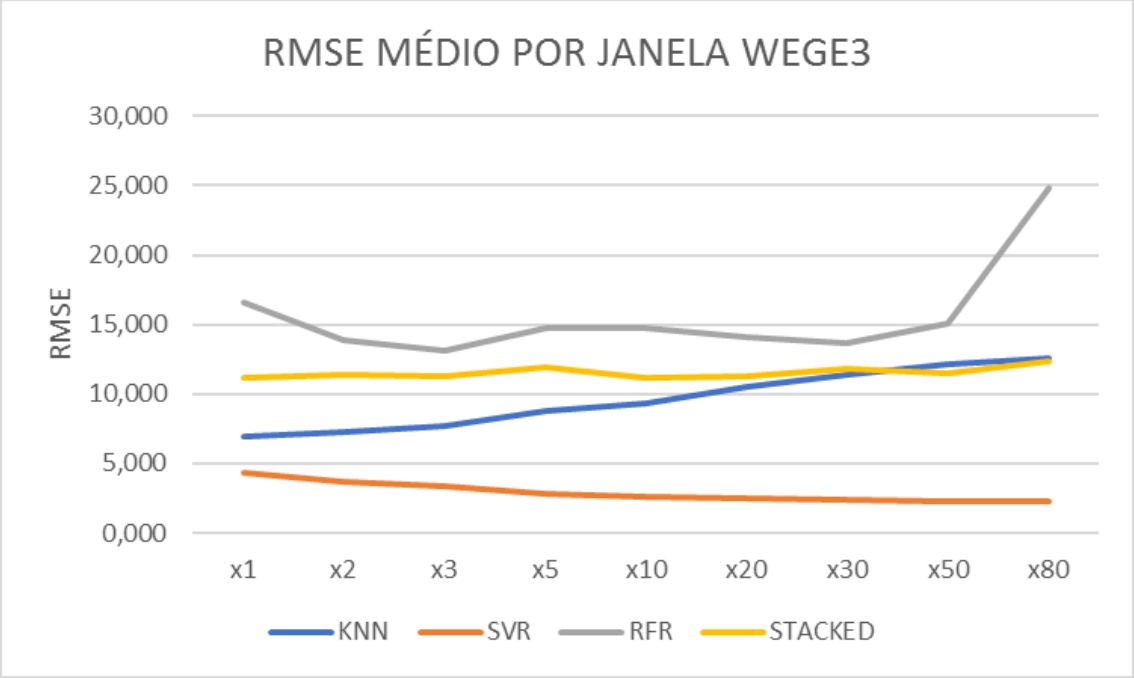


Fonte: Elaborado pelo autor.

O melhor resultado de RMSE obtido foi 2,289, o que corresponde a 2,98% da cotação do ativo no fim do período, que foi R$ 76,70. Bons resultados foram obtidos na métrica ACC para o ativo com o modelo SVR, que atingiu 80,9% de acerto em dois modelos e constituiu ACC médio de 73% com treinamento diário. Nota-se que ao contrário do que ocorreu para PETR4, treinar o modelo semanalmente ou diariamente resultou em piora dos resultados de RMSE. A ACC, por sua vez, apresentou significativa melhora nos modelos com treinamento diário.

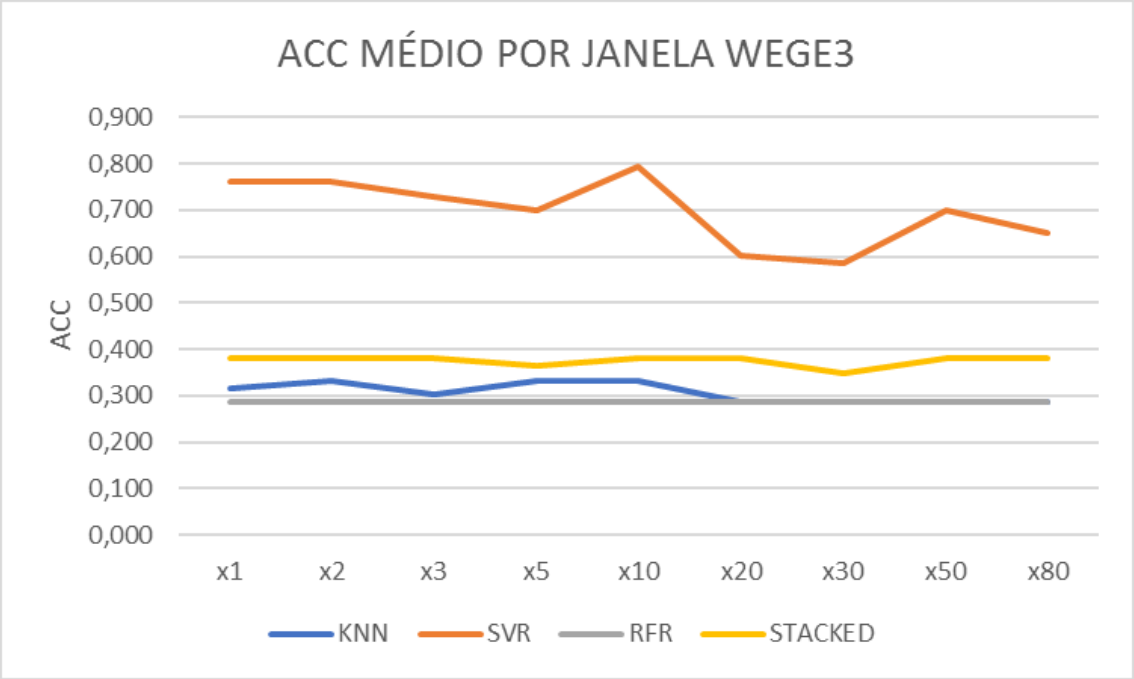
Nas figuras 23 e 24 são apresentados os resultados de performance por janela de predição.

Figura 23 – Gráfico de RMSE médio por janela na validação de WEGE3



Fonte: Elaborado pelo autor.

Figura 24 – Gráfico de ACC média por janela na validação de WEGE3



Fonte: Elaborado pelo autor.

Para a métrica RMSE verifica-se que aumentar o tamanho da janela melhora a performance no caso do SVR. Para os demais, que apresentaram valores altíssimos de erro, houve piora nos resultados. No caso da ACC, é registrada ligeira piora conforma aumenta o tamanho exceto pela janela x10 que apresentou um valor alto de acerto. Para os demais algoritmos os resultados se mantiveram estáveis e não se nota melhora ou piora significativa nos valores.

4.3 Análise dos resultados por ativo

O período de vinte e um dias de validação dos modelos construídos forneceu dados que contribuem para uma análise mais completa da performance do sistema. Foram obtidos resultados não esperados em vários cenários quando comparados com a performance dos modelos no período de treinamento. Os valores de RMSE encontrados são consideravelmente piores do que os da primeira etapa. Os resultados da ACC, por outro lado, indicam que alguns modelos conseguiram manter a sua acurácia embora os valores de RMSE subissem consideravelmente.

Nos itens seguintes serão apresentados dados relevantes sobre os ativos analisados no mês de outubro de 2020 com o objetivo de serem identificadas

possíveis relações desses dados com os resultados obtidos pelos modelos no período de validação.

4.3.1 Análise do ativo ITUB4

Considerado um ativo de alta estabilidade no mercado financeiro, o ITUB4 manteve esse comportamento praticamente durante todo mês de outubro de 2020, com exceção de quatro dias marcados por grande volatilidade, conforme pode ser visto na figura 25. Apesar dos dias citados, no restante do mês as cotações se mantiveram dentro de um limite considerado estável.

Figura 25 – Volatilidade de ITUB4 em outubro de 2020

Data	Fechamento	Variação	Variação (%)	Abertura	Máxima	Mínima	Volume
30 Out 2020	23,57	-0,04	-0,17%	23,65	23,93	23,23	39.190.000
29 Out 2020	23,61	-0,27	-1,13%	23,58	24,04	23,04	42.962.500
28 Out 2020	23,88	-0,73	-2,97%	24,25	24,42	23,73	38.652.600
27 Out 2020	24,61	-1,03	-4,02%	26,10	26,17	24,54	47.280.100
26 Out 2020	25,64	0,19	0,75%	25,30	25,86	25,16	32.133.100
23 Out 2020	25,45	-0,35	-1,36%	25,90	26,30	25,41	50.548.200
22 Out 2020	25,80	1,32	5,39%	24,57	25,92	24,55	66.456.800
21 Out 2020	24,48	0,14	0,58%	24,26	24,94	24,07	38.138.400
20 Out 2020	24,34	0,95	4,06%	23,52	24,40	23,50	45.724.900
19 Out 2020	23,39	0,24	1,04%	23,25	23,95	22,94	38.280.600
16 Out 2020	23,15	-0,36	-1,53%	23,41	23,45	23,09	21.261.200
15 Out 2020	23,51	-0,14	-0,59%	23,46	23,73	23,26	21.805.300
14 Out 2020	23,65	-0,05	-0,21%	23,89	23,97	23,59	22.051.900
13 Out 2020	23,70	-0,31	-1,29%	23,98	24,13	23,62	31.012.000
12 Out 2020	24,01	0,00	+0,00%	23,90	24,61	23,59	0
09 Out 2020	24,01	-0,07	-0,29%	23,90	24,61	23,59	48.948.600
08 Out 2020	24,08	1,40	6,17%	22,72	24,28	22,60	63.209.400
07 Out 2020	22,68	-0,14	-0,61%	22,93	23,00	22,57	22.796.100
06 Out 2020	22,82	0,01	0,04%	23,09	23,38	22,71	21.135.900
05 Out 2020	22,81	0,24	1,06%	22,72	23,01	22,34	22.837.500
02 Out 2020	22,57	0,05	0,22%	22,36	23,32	22,34	35.482.000
01 Out 2020	22,52	0,03	0,13%	22,55	22,60	22,18	20.474.100

Fonte: Adaptado de ADVFN, 2020.

No período de testes dos modelos construídos para o ITUB4 o algoritmo com KNN apresentou os melhores resultados, comportamento que não ficou evidente no

período de validação. Na métrica RMSE, o KNN foi superado pelo SVR em grande margem, enquanto na ACC os dois apresentaram resultados próximos na média.

4.3.2 Análise do ativo PETR4

O ativo PETR4 apresentou uma variação diária em suas cotações maior do que o ITUB4, conforme pode ser visto na figura 26. Poucos dias de operação do ativo apresentaram variação diária menor do que 1%, o que caracteriza um período de maior volatilidade para o ativo em comparação ao que ocorreu para o ITUB4.

Figura 26 – Volatilidade de PETR4 em outubro de 2020

Data	Fechamento	Variação	Variação (%)	Abertura	Máxima	Mínima	Volume
30 Out 2020	19,01	-0,21	-1,09%	19,14	19,54	18,87	64.185.100
29 Out 2020	19,22	0,51	2,73%	18,43	19,37	17,74	86.794.800
28 Out 2020	18,71	-0,97	-4,93%	19,35	19,44	18,61	79.247.400
27 Out 2020	19,68	-0,56	-2,77%	20,26	20,37	19,68	48.978.400
26 Out 2020	20,24	-0,42	-2,03%	20,33	20,53	20,03	58.229.100
23 Out 2020	20,66	-0,21	-1,01%	20,94	21,14	20,54	61.711.200
22 Out 2020	20,87	0,72	3,57%	20,10	20,87	20,05	98.926.900
21 Out 2020	20,15	-0,08	-0,4%	20,16	20,33	19,83	60.594.900
20 Out 2020	20,23	0,67	3,43%	19,67	20,27	19,59	68.407.700
19 Out 2020	19,56	0,19	0,98%	19,41	19,94	19,25	107.419.300
16 Out 2020	19,37	-0,43	-2,17%	19,67	19,71	19,30	49.929.000
15 Out 2020	19,80	-0,20	-1,0%	19,62	19,80	19,44	56.952.100
14 Out 2020	20,00	-0,05	-0,25%	20,09	20,37	19,95	44.584.300
13 Out 2020	20,05	0,23	1,16%	19,91	20,13	19,73	55.330.600
12 Out 2020	19,82	0,00	+0,00%	20,34	20,39	19,80	0
09 Out 2020	19,82	-0,58	-2,84%	20,34	20,39	19,80	67.962.300
08 Out 2020	20,40	0,50	2,51%	20,04	20,59	19,86	78.070.400
07 Out 2020	19,90	0,00	0,0%	20,00	20,05	19,56	53.521.000
06 Out 2020	19,90	-0,17	-0,85%	20,47	20,70	19,87	77.064.600
05 Out 2020	20,07	1,07	5,63%	19,36	20,16	19,24	73.288.500
02 Out 2020	19,00	-0,88	-4,43%	19,73	19,74	19,00	85.306.000
01 Out 2020	19,88	0,25	1,27%	19,64	20,06	19,13	77.537.300

Fonte: Adaptado de ADVFN, 2020.

No período de testes, novamente o algoritmo com melhor desempenho foi o KNN, o que também não se repetiu para o período de validação. O algoritmo SVR

apresentou resultados melhores tanto para RMSE quanto ACC, desta vez obtidos com treino diário dos modelos.

4.3.3 Análise do ativo WEGE3

Ao analisarmos o comportamento do ativo no mês utilizado para validação nota-se que este apresentou a maior volatilidade no valor de suas cotações, conforme é apresentado na figura 27.

Figura 27 – Volatilidade de WEGE3 em outubro de 2020

Data	Fechamento	Variação	Variação (%)	Abertura	Máxima	Mínima	Volume
30 Out 2020	76,70	-3,28	-4,1%	80,33	80,74	75,83	9.597.700
29 Out 2020	79,98	0,85	1,07%	79,40	81,60	79,32	8.871.600
28 Out 2020	79,13	-4,11	-4,94%	82,00	82,69	78,75	8.129.800
27 Out 2020	83,24	0,59	0,71%	82,97	84,34	82,54	6.747.900
26 Out 2020	82,65	0,92	1,13%	81,49	83,99	81,25	6.316.900
23 Out 2020	81,73	-0,60	-0,73%	82,25	83,26	80,64	8.096.400
22 Out 2020	82,33	2,98	3,76%	79,65	82,68	78,45	13.136.300
21 Out 2020	79,35	-4,53	-5,4%	87,00	87,21	78,31	20.608.300
20 Out 2020	83,88	1,82	2,22%	82,90	83,95	81,47	6.467.500
19 Out 2020	82,06	-0,19	-0,23%	83,01	83,81	81,11	5.843.800
16 Out 2020	82,25	1,20	1,48%	81,06	83,36	80,91	6.411.700
15 Out 2020	81,05	0,76	0,95%	79,64	81,28	78,10	6.125.200
14 Out 2020	80,29	1,04	1,31%	79,26	80,98	79,26	6.618.700
13 Out 2020	79,25	2,09	2,71%	77,88	79,50	77,35	6.076.300
12 Out 2020	77,16	0,00	+0,00%	74,94	77,20	74,55	0
09 Out 2020	77,16	2,20	2,93%	74,94	77,20	74,55	5.789.300
08 Out 2020	74,96	1,51	2,06%	73,36	75,56	73,32	7.218.300
07 Out 2020	73,45	1,67	2,33%	71,78	74,19	71,68	7.759.000
06 Out 2020	71,78	1,63	2,32%	70,41	72,29	70,30	7.761.600
05 Out 2020	70,15	4,15	6,29%	67,25	70,18	67,12	7.236.300
02 Out 2020	66,00	-0,90	-1,35%	66,48	67,38	65,80	5.795.200
01 Out 2020	66,90	1,30	1,98%	65,84	66,90	64,94	5.192.300

Fonte: Adaptado de ADVFN, 2020.

A modelagem com algoritmos SVR foi a que forneceu resultados mais próximos do aceitável para o período de validação, conforme visto na sessão anterior. Apesar dos valores de RMSE obtidos serem inferiores aos obtidos no período de testes, os valores de ACC apresentaram ótimos resultados.

4.4 Síntese dos resultados obtidos

Considerando os dois períodos de coleta de métricas de resultado, teste e validação, considera-se que os resultados obtidos atendem parcialmente a proposta. Conforme evidenciado na etapa de validação dos modelos, a métrica RMSE apresentou resultados consideravelmente ruins em comparação com o período de treinamento. Os melhores resultados, obtidos pelos modelos SVR, chegaram a representar mais de 3% do valor do ativo. A métrica ACC, por outro lado, apresentou resultados considerados bons tanto na etapa de modelagem dos algoritmos quanto na validação, atingindo valores superiores a 70%.

Analisando individualmente os modelos, verificou-se que os melhores resultados foram obtidos pelos algoritmos KNN, no período de testes, e SVR, no período de validação. Os modelos *stacked* não apresentaram bons resultados em comparação aos algoritmos KNN, SVR e RFR utilizados de forma individual.

A tabela 33 apresenta um resumo com os resultados obtidos pelos melhores modelos em cada ativo analisado nos períodos de teste e validação. No caso da coluna Valor, os valores de RMSE são acrescidos do percentual que representam da cotação do ativo ao fim do período de modelagem, entre parêntesis. No caso da métrica ACC, essa mesma coluna Valor exibe entre parêntesis o valor da ACC convertido para base percentual.

Tabela 33 – Síntese dos melhores resultados obtidos por ativo

Etapa	Métrica	Ativo	Modelo	Treinamento	Valor	Resultado
Testes	RMSE	ITUB4	knn_x80	Modelagem	0,305 (1,36%)	Satisfatório
		PETR4	knn_x30	Modelagem	0,473 (2,41%)	Insatisfatório
		WEGE3	svr_x50	Modelagem	0,378 (0,58%)	Satisfatório
	ACC	ITUB4	knn_x80	Modelagem	0,735 (73,5%)	Satisfatório
		PETR4	knn_x80	Modelagem	0,733 (73,3%)	Satisfatório
		WEGE3	knn_x80	Modelagem	0,753 (75,3%)	Satisfatório
Validação	RMSE	ITUB4	svr_x30	Mensal	0,555 (2,35%)	Insatisfatório
		PETR4	svr_x30	Diário	0,507 (2,67%)	Insatisfatório
		WEGE3	svr_x80	Diário	2,208 (2,88%)	Insatisfatório
	ACC	ITUB4	knn_x3	Mensal	0,761 (76,1%)	Satisfatório
		PETR4	svr_x50	Diário	0,761 (76,1%)	Satisfatório
		WEGE3	svr_x10	Diário	0,809 (80,9%)	Satisfatório

Fonte: Elaborado pelo autor.

Conforme apresentado na tabela 33, é possível verificar que no período de testes o algoritmo knn_80 apresentou a melhor performance dentre todas as análises realizadas. Os resultados obtidos foram satisfatórios, à exceção da métrica RMSE para PETR4, 0,41% acima do limite pré-determinado.

Para o período de validação se constatou que os valores de RMSE obtidos não foram satisfatórios para todos os ativos analisados. Quanto a métrica ACC, por outro lado, os resultados obtidos foram excelentes considerando o limite estipulado de 60%. Os modelos baseados em SVR apresentaram os melhores resultados nessa etapa, com valores bem acima dos outros modelos utilizados. De modo geral, também pode se avaliar que o treino diário trouxe ganhos de performance, melhor evidenciado nos casos em que o ativo apresenta alta volatidade, como foi o caso de WEGE3 e PETR4. Para ITUB4, em que a volatilidade da ação não foi grande para o período de validação, nota-se que os modelos com treinamento mensal tiveram os melhores resultados.

5 CONCLUSÃO E TRABALHOS FUTUROS

O objetivo proposto por esse projeto foi a construção de um robô preditor para análise de ativos da B3 utilizando conceitos de *machine learning*, com a finalidade de realizar predições dos valores de cotações futuras e então aferir sua performance através das métricas RMSE e acurácia (ACC). Essa última, foi adaptada com o objetivo de traduzir os resultados para uma representação mais próxima do mercado e das operações de *trading*, que buscam prever diretamente o comportamento da cotação de um ativo em curto período. Escolhidos os ativos ITUB4, PETR4 e WEGE3, o *software* desenvolvido na linguagem de programação *Python* com a biblioteca *sklearn* construiu modelos de predição utilizando os algoritmos KNN, SVR, RFR e *XGBoost*, sendo esse último construído através da técnica *stacking* com a junção dos outros três citados.

O robô preditor foi capaz de interpretar dados históricos dos ativos escolhidos com registros que compreendem cerca de vinte anos de operação na bolsa de valores. Os dados foram processados e analisados sob nove óticas temporais diferentes, através do conceito de janelas. A aplicação então construiu os modelos de *machine learning*, cujas predições foram avaliadas em dois períodos distintos: teste e validação.

Durante o período de teste, constatou-se que a inclusão das cotações do índice IBOV e do barril de petróleo cru, no caso de PETR4, melhoraram significativamente os resultados de RMSE e ACC para todos os ativos em comparação com a modelagem sem a sua incorporação. Adicionalmente, verificou-se que o aumento no tamanho da janela de cotações utilizada na modelagem provocou melhora significativa na performance. Os resultados das métricas para o período de testes foi considerada satisfatória, com a exceção da métrica RMSE para o ativo PETR4 que não atingiu o resultado esperado.

Para o período de validação do robô preditor foi escolhido o mês de outubro de 2020. Os resultados de RMSE apresentados para os ativos foram inferiores aos obtidos no período de testes por grande margem, apesar de serem utilizadas nas modelagens técnicas para evitar sobreajuste, como dividir o conjunto de dados em subconjuntos e a utilização de *randomsearch* com *cross-validation* para efetuar *tunning* de parâmetros dos algoritmos. A métrica ACC, por outro lado, continuou a

apresentar resultados satisfatórios com taxas de 76,1% de acerto para ITUB4 e PETR4, e 80,9% para WEGE3.

Quanto aos resultados insatisfatórios obtidos na métrica RMSE para os ativos, acredita-se que para melhorar os resultados da métrica na modelagem devem ser exploradas outras técnicas de *machine learning* que envolvam conversão e análise de correlação das variáveis, com o objetivo de analisar dados em potencial que poderiam ser incorporados aos dados históricos analisados.

A análise de janelas maiores poderia também trazer melhora nos resultados. Para o escopo desse projeto o limite estipulado foi de oitenta períodos, carecendo a exploração em janelas maiores de tempo. Adicionalmente, pode se citar a simulação dos valores intermediários das janelas definidas. O principal fator limitante para a realização dessa quantidade de simulações é a complexidade computacional para a geração dos modelos de predição para cada dia, o que torna necessária a alocação de recursos de hardware mais potentes para ser viável a sua utilização.

Quanto aos resultados insatisfatórios do modelo construído com a técnica *stacking* utilizando o algoritmo *XGBoost*, atribui-se ao fato do mesmo não ter sido explorado em sua totalidade, principalmente na etapa de *tunning* de parâmetros, que pelo tempo necessário para execução teve de ser limitado a rodar com valores *default*. No âmbito desse projeto pode oferecer contribuição aos resultados a sua utilização de forma separada como um quarto modelo de predição, e o uso da técnica de *stacking* com um algoritmo mais simples, como a regressão logística.

Além dos pontos citados, para trabalhos futuros sugere-se a portabilidade dos algoritmos criados para a linguagem de programação C++ e a utilização da biblioteca *tensorflow* para modelagem. Essa mudança traria um ganho significativo de performance ao sistema e possibilitaria maior número de simulações e análises, incluindo *tunning* de parâmetros mais exaustivos para otimização.

Por fim, acredita-se que os resultados foram bons no contexto da solução proposta. A análise de ativos financeiros possui o claro objetivo de se estimar a direção do ativo, então credita-se aos bons resultados obtidos pela métrica ACC a relevância do projeto para contribuir junto aos demais trabalhos na área e estudados na bibliografia.

REFERÊNCIAS

XAVIER, A. Estratégias estatísticas em investimentos. São Paulo: Novatec, 2009.

DEBASTIANI, C. A. Análise Técnica de Ações - Identificando oportunidades de compra e venda. São Paulo: Novatec, 2008.

ALDRIDGE, Irene. High-Frequence Trading. Hoboken, New Jersey, USA: John Wiley & Sons, Inc., 2010.

RUSSEL, Stuart; NORVIG, Peter. Artificial Intelligence: A Modern Approach Third Edition. Upper Saddle River, New Jersey, USA: Pearson Education, Inc., 2010.

GÉRON, Aurélien. Hands-On Machine Learning with Scikit-Learn and TensorFlow. Sebastopol, California, USA: O'Reilly Media, Inc., 2017.

SILVA, Diego Maicon. THALER - Um protótipo de robô investidor utilizando análise técnica e máquinas de vetores de suporte. Instituto Federal Minas Gerais. Dissertação para obtenção do grau de Bacharel em Ciência da Computação, julho de 2018.

NAMETALA, Ciniro Aparecido Leite. Construção de um Robô Investidor baseado em Redes Neurais Artificiais e Preditores Econométricos. Universidade Federal de Minas Gerais. Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, fevereiro de 2017.

LU, Ning. A Machine Learning Approach to Automated Trading. Boston College. Tese apresentada ao Programa Senior de Ciência da Computação, maio de 2016.

NAIR, Binoy B.; DHARINI, N.Mohana; MOHANDAS, V.P. A Stock market trend prediction system using a hybrid decision tree-neuro-fuzzy system. International Conference on Advances in Recent Technologies in Communication and Computing, 2010.

BAO, Wei; YUE, Jun; RAO, Yulei. A deep learning framework for financial time series using stacked autoencoders and longshort term memory. PLoS ONE, julho de 2017.

CNBC. Site CNBC. Just 10% of trading is regular stock picking, JPMorgan estimates. Disponível em: https://www.cnbc.com/2017/06/13/death-of-the-human-investor-just-10-percent-of-trading-is-regular-stock-picking-jpmorgan-estimates.html. Acesso em: 31/05/2020.

NOONAN, Laura. JPMorgan develops robot to execute trades. Financial Times, 2017. Disponível em: https://www.ft.com/content/16b8ffb6-7161-11e7-aca6-c6bd07df1a3c. Acesso em: 31/05/2020.

Cointimes. Site Cointimes. Investidores usam inteligência artificial para ganhar vantagem nos mercados. Disponível em: https://cointimes.com.br/investidores-usam-inteligencia-artificial/. Acesso em 31/05/2020.

ROCHA, André. As estratégias de Warren Buffett. Site Valor, 2011. Disponível em: https://www.valor.com.br/valor-investe/o-estrategista/999486/estrategias-de-warren-buffett. Acesso em: 31/05/2020.

TRADINGVIEW. Site TradingView. Índice Ibovespa: gráfico completo. Disponível em: https://br.tradingview.com/chart/?symbol=BMFBOVESPA%3AIBOV. Acesso em: 27/06/2020.

____. Site TradingView. Cotação Bitcoin: gráfico complete. Disponível em: https://br.tradingview.com/chart/?symbol=BITSTAMP%3ABTCUSD. Acesso em: 27/06/2020.

ADVFN. Site ADVFN. ITUB4 dados históricos. Disponível em: https://br.advfn.com/bolsa-de-valores/bovespa/itau-unibanco-pn-ITUB4/historico/mais-dados-historicos. Acesso em: 27/06/2020.

WAWRZENIAK, Diego. O Que É Análise Técnica? Site Bússola do Investidor, 2014. Disponível em: https://www.bussoladoinvestidor.com.br/o-que-e-analise-tecnica/. Acesso em: 27/06/2020.

ORDONES, Arthur. Como fazer para investir no fundo que rende 80% ao ano desde 1988? Site InfoMoney: 2014. Disponível em: https://www.infomoney.com.br/onde-investir/como-fazer-para-investir-no-fundo-que-rende-80-ao-ano-desde-1988/. Acesso em: 27/06/2020.

CHIAVEGATTO, Alexandre. Inteligência artificial e machine learning com foco em predição. 2018. (11min32s). Disponível em: https://youtu.be/eiZoEw-_GA0?list=PLAudUnJeNg4tvUFZ8tXQDoAkFAASQzOHm. Acesso em: 21/06/2020.

HARRISON, Onel. Machine Learning Basics with the K-Nearest Neighbors Algorithm. Site Towards Data Science: 2018. Disponível em: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761. Acesso em: 28/06/2020.

DWIVEDI, Rohit. How Does Linear And Logistic Regression Work In Machine Learning? Site Analytic Steps: 2020. Disponível em:

https://www.analyticssteps.com/blogs/how-does-linear-and-logistic-regression-work-machine-learning. Acesso em: 28/06/2020.

Yahoo Finanças. Site Yahoo Finanças. Disponível em: https://br.financas.yahoo.com/. Acesso em: 31/10/2020.

MORETTIN, P. A.; TOLOI, C. M. C. Análise de Séries Temporais. 2o . ed. São Paulo: Blucher, 2006.