UNIVERSIDADE FEDERAL DE SANTA CATARINA

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

YURI BATISTA

Predição de séries temporais em projetos ágeis

FLORIANÓPOLIS
2022

YURI BATISTA

Predição de séries temporais em projetos ágeis

Trabalho de Conclusão do Curso de Graduação em Sistemas de Informação, do Departamento de Informática e Estatística, do Centro Tecnológico da Universidade Federal de Santa Catarina, requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Elder Rizzon Santos.

FLORIANÓPOLIS
2022

RESUMO

A metodologia ágil está presente em grande parte das empresas - principalmente nos segmentos de tecnologia -, e vem contribuindo para alcançarem o sucesso com a velocidade acelerada que o mercado exige cada vez mais nas últimas décadas. Uma dos principais framework utilizados é o Scrum, em que os objetivos são divididos em tarefas menores e criadas sprints periódicas, para finalizá-las em um curto período de tempo. Para auxiliar, são utilizadas ferramentas para organização das sprints, em que são guardadas no histórico e, posteriormente, os dados podem ser transformados em uma série temporal. Neste trabalho, após inúmeros testes, foi desenvolvido um modelo para predizer o tempo necessário para concluir cada tarefa das sprints em empresas de tecnologia, a partir de métodos preditivos que utilizam Machine Learning, com o objetivo de apoiar negócios a serem mais assertivos em seus planejamentos. Neste sentido, a empresa JExperts apoiou com o desenvolvimento do trabalho a partir dos dados de seus projetos de sua própria plataforma, que possui um módulo Agile, o Channel, para a criação de séries temporais em que foram utilizadas, testadas e avaliadas em diversos métodos de predição de dados. O método Random Forest obteve os melhores resultados nas três séries temporais utilizadas, onde o tempo predito para finalizar as tarefas teve uma média de erro de aproximadamente cinco minutos, em tarefas que em média levaram 85 a 90 minutos para serem finalizadas, e os piores resultados alcançados por esse mesmo modelo, foi um erro médio de aproximadamente 11 minutos em tarefas que em média foram finalizados entre 70 e 80 minutos. Então conclui-se que é possível utilizar essa técnica em projetos ágeis.

Palavras-chave: Predição de séries temporais, Metodologia ágil, Aprendizado de máquina, Random Forest.

ABSTRACT

The agile methodology is present in most companies - mainly in the technology segments - and has contributed to achieving success with the accelerated speed that the market has increasingly demanded in recent decades. One of the main frameworks used is Scrum, in which objectives are divided into smaller tasks and periodic sprints are created to complete them in a short period of time. To help, tools are used to organize the sprints, in which they are stored in the history and, later, the data can be transformed into a time series. In this work, after numerous tests, a model was developed to predict the time needed to complete each sprint task in technology companies, based on predictive methods that use Machine Learning, with the objective of supporting businesses to be more assertive in their planning . In this sense, the company JExperts supported the development of the work based on data from its projects on its own platform, which has an Agile module, the Channel, for creating time series in which they were used, tested and evaluated in various methods. of data prediction. The Random Forest method obtained the best results in the three time series used, where the predicted time to finish the tasks had an average error of approximately five minutes, in tasks that took 85 to 90 minutes on average to be finished, and the worst results achieved by this same model, was an average error of approximately 11 minutes in tasks that on average were completed between 70 and 80 minutes. So it is concluded that it is possible to use this technique in agile projects.

Key-words: Time series prediction, Agile methodology, Machine learning, Random Forest.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ST	Série Temporal
kNN	k-Nearest Neighbors
kNN-TSPI	k-Nearest Neighbors - Time Series Prediction with Invariances
kNN-TSP	k-Nearest Neighbors - Time Series Prediction
MASE	Mean Absolute Scaled Error
MSE	Mean Squared Error
POCID	Prediction of Change in Direction
ARIMA	Autoregressive Integrated Moving Average
SARIMA	Seasonal Autoregressive Integrated Moving Average
SVM	Support vector machines
MLP	Multilayer Perceptron
RMSE	Root Mean-Square Error
MAE	Mean Absolute Error
ML	Machine Learning
SP	Seleção Progressiva
PBS	Poda Baseada em Sensibilidade
MAPE	Mean Absolute Percentage Error
ARV	Average Relative Variance
SES	Suavização Exponencial Simples
MA	Moving Averages
HES	Suavização Exponencial de Holt
AHW	Holt-Winters Aditivo
MHW	Holt-Winters Multiplicativo
TU	U de Theil
IA	Index of Agreement
SVR	Support Vector Regression
RF	Random Forest
TI	Tecnologia da informação

SUMÁRIO

1. INTRODUÇÃO

A 4ª Revolução Industrial aumentou a velocidade das mudanças com que as empresas lutam para conseguir acompanhar o ritmo de crescimento. Mas, o que vimos na última década, é um novo padrão muito mais rápido, em que as organizações buscam formas de acompanhar o mesmo movimento. Um dos esforços para alcançar o objetivo é a implementação de metodologias ágeis.

Métodos ágeis transformaram os projetos de tecnologia e aumentaram as taxas de sucesso dos times de Tecnologia da informação (TI) nas últimas duas décadas. Hoje, as metodologias ágeis estão sendo aplicadas em todas as áreas da empresa, otimizando o trabalho de desenvolvimento, seja em projetos de softwares ou em qualquer outro processo. Segundo o relatório 15th *State of the Agile Report*, 94% dos entrevistados reportaram que a empresa que trabalham utiliza a metodologia ágil.

Existem diversos métodos para utilizar a gestão ágil e determinadas metodologias podem ser mais adequadas conforme a necessidade do projeto. No entanto, segundo o relatório 15th *State of the Agile Report*, 66% dos entrevistados responderam que utilizam Scrum (DIGITAL, 2021). O Scrum é uma metodologia que procura ser eficaz, produtivo e controlar melhor os riscos do processo, consiste em uma série de *sprints*, que possuem tarefas a serem concluídas até o final da duração de um mês ou menos (GONÇALVES, 2018).

Para facilitar a gestão, as empresas utilizam ferramentas ágeis de gerenciamento de projetos, em que as *sprints* e tarefas são organizadas e mantêm histórico à disposição. Assim, após algum tempo utilizando a metodologia ágil como recurso e armazenando os dados gerados, é formada uma série temporal. Uma série temporal, também denominada série histórica, é uma coleção de dados em intervalos regulares feitas sequencialmente ao longo do tempo (LATORRE e CARDOSO, 2001).

Assim, com a análise de dados e métodos preditivos, é possível gerar informações concretas para apoiar a tomada de decisão. Uma das decisões mais favorecidas pelo histórico de dados e mais usada indiretamente, é a estimativa de tempo de conclusão para cada tarefa, em que o time não olha diretamente para os

dados, mas tomam decisões com base na experiência de executar uma tarefa parecida no passado, limitando-se apenas pela memória dos integrantes.

Dessa forma, é possível observar a aplicação da metodologia ágil como ferramenta para contribuir com as análises de dados, que possui duas vertentes: descritiva e preditiva. A análise descritiva busca compreender os acontecimentos a partir da conversão de dados úteis que passaram por processos de categorização, caracterização, consolidação e classificação (EVANS, 2012). Em contrapartida, a análise preditiva examina as relações entre os conjuntos de dados históricos a serem explorados, de forma a compreender e buscar prever valores futuros (EVANS, 2012).

2. OBJETIVOS

2.1 OBJETIVO GERAL

O objetivo geral é desenvolver um modelo para predizer o tempo estimado de execução de tarefas de projetos da empresa JExperts.

2.2 OBJETIVOS ESPECÍFICOS

- Compreender e consolidar conceitos de métodos ágeis, predição, séries temporais e análise de dados;
- Analisar técnicas e modelos de predição de dados e séries temporais;
- Definir um modelo para predição de tempo das tarefas;
- Realizar um experimento para avaliação do modelo.

3. FUNDAMENTAÇÃO TEÓRICA

3.1 METODOLOGIA ÁGIL

As metodologias ágeis para desenvolvimento de software são uma resposta às chamadas metodologias pesadas ou tradicionais (SOARES, 2004). Ainda de acordo com Soares (2004), o termo “Metodologias Ágeis” tornou-se popular em

2001, quando 17 especialistas e processos de desenvolvimento de software representando os métodos Scrum (SCHWABER e BEEDLE, 2001), *Extreme Programming* (XP) (BECK, 1999) e outros, estabeleceram princípios comuns compartilhados por todos esses métodos.

A partir destes estudos, foi criada então a Aliança Ágil, estabelecendo o Manifesto Ágil com quatro conceitos chave: indivíduos e interações ao invés de processos e ferramentas; software executável ao invés de documentação; colaboração do cliente ao invés de negociação de contratos e respostas rápidas às mudanças ao invés de seguir planos.

Em comparação às metodologias tradicionais, a ideia das metodologias ágeis é o enfoque nas pessoas, e não em processos ou algoritmos. Existe a preocupação em gastar menos tempo com a documentação, e mais tempo com a implementação (SOARES, 2004).

Diante deste cenário, tornou-se cada vez mais importante a gestão de projetos e a necessidade de criar um equilíbrio entre as demandas de escopo, tempo, custo, qualidade e bom relacionamento com o cliente (TORREÃO, 2006), contexto que surge a metodologia Scrum.

3.1.1 SCRUM

Com o objetivo de fornecer um processo conveniente para projeto e desenvolvimento orientado a objeto, de acordo com Soares (2004), a Scrum apresenta uma abordagem empírica, que aplica algumas ideias da teoria de controle de processos industriais para o desenvolver softwares, com as premissas de flexibilidade, adaptabilidade e produtividade.

Neste cenário aplicado à metodologia Scrum, os lançamentos de produtos de software são planejados com base nas seguintes variáveis, segundo Schwaber (1995):

• **Requisitos do cliente**: como o sistema atual precisa ser aprimorado;

• **Pressão de tempo:** qual prazo é necessário para obter uma vantagem competitiva;

• **Competição:** o que a concorrência está fazendo e o que é necessário para superá-la;

• **Qualidade:** qual é a qualidade exigida, dadas as variáveis acima;

• **Visão:** quais mudanças são necessárias nesta fase para cumprir a visão do sistema;

• **Recurso:** quantas pessoas e financiamento estão disponíveis.

Tais variáveis formam o planejamento inicial para um projeto de aperfeiçoamento de software, entretanto, podem mudar ao longo do caminho. Um framework de desenvolvimento bem-sucedido precisa levar em conta essas variáveis e sua natureza evolutiva (SCHWABER, 1995).

3.2 SÉRIES TEMPORAIS

Uma série temporal (ST) é uma coleção de observações feitas sequencialmente no tempo. Os exemplos ocorrem em uma variedade de campos, desde a economia até a engenharia, e os métodos de análise de séries temporais constituem uma importante área da estatística (CHATFIELD, 2013).

De acordo com Parmezan (2016), os métodos para predição de séries temporais são baseados essencialmente na ideia de que dados históricos contemplam padrões intrínsecos, geralmente de difícil identificação e nem sempre interpretáveis que, se descobertos, podem auxiliar na descrição futura do fenômeno investigado. Neste sentido, as séries temporais buscam responder em quais circunstâncias, determinados padrões se repetem, bem como as mudanças que podem acontecer ao longo do tempo.

Assim, a série temporal é composta por uma função onde Tt é a tendência da série, St é a sazonalidade e Et o ruído (PARMEZAN, 2016), que serão explicados em detalhes posteriormente. Em uma decomposição clássica, temos:

$$X = f(T_t, S_t, E_t) \tag{3.1}$$

Para Chatfield (2013), a análise de uma série temporal pode ser feita a partir de quatro grupos de objetivos:

1. **Descrição:** este tipo de análise busca descrever os comportamentos da ST, como existência ou não de tendências, variação sazonal, discrepâncias (outliers) e alterações na estrutura da série como pontos de curva, ou seja, mudanças no padrão de uma tendência ou sazonalidade crescente para descrente;

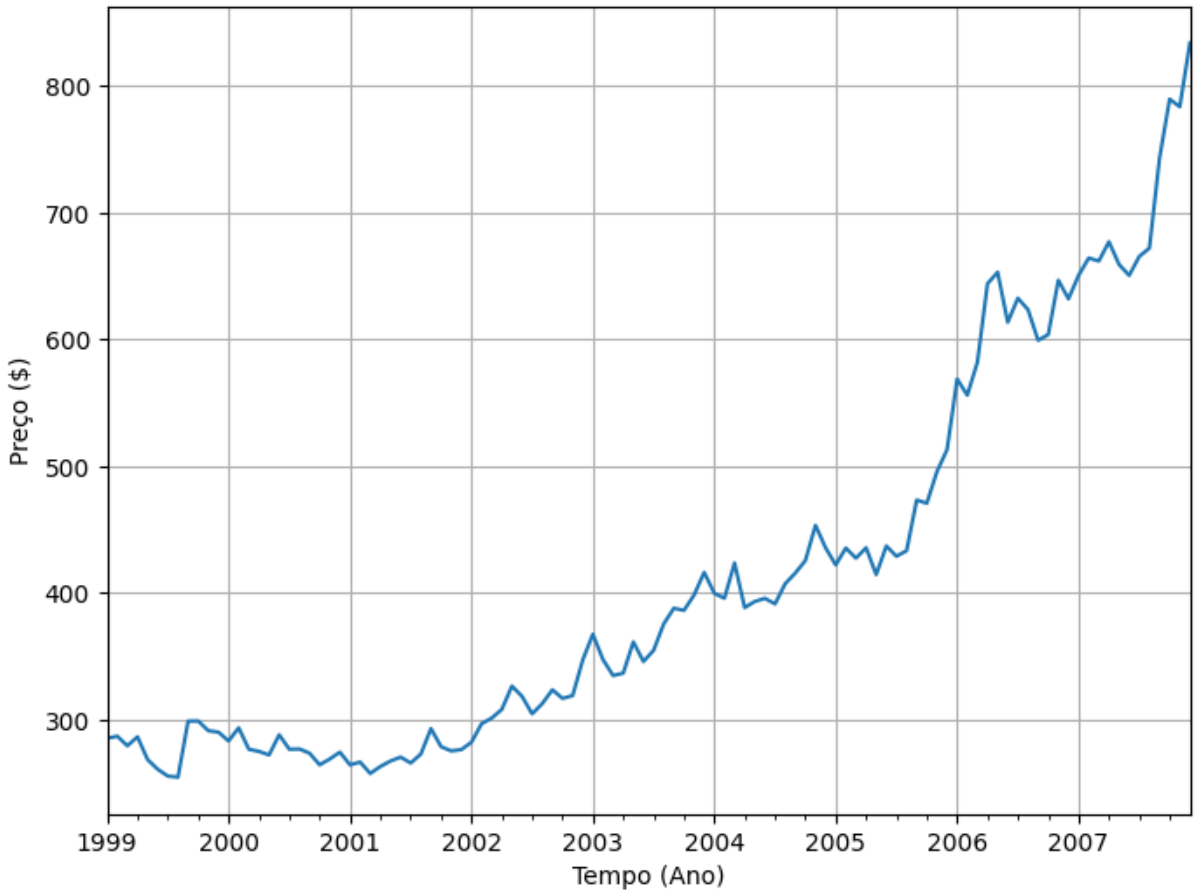
2. **Explicação**: esta análise exige a identificação de duas ou mais variáveis para apoiar na determinação das paridades entre duas séries temporais. Este tipo de análise permite explicar a variação de uma série com base em outra;
3. **Predição:** esta análise busca sintetizar as propriedades presentes na série temporal para caracterizar seu comportamento, identificando e sugerindo a partir de dados históricos, um modelo que permita predizer os valores futuros da série;
4. **Controle:** nesta análise, os valores de uma série temporal expressam dados de controle sobre determinado processo, com o objetivo de mensurar a sua qualidade.

Na **Figura 1,** é esquematizada uma ST real referente ao preço do ouro nos Estados Unidos, no período de janeiro de 1999 a dezembro de 2007. Essas observações foram coletadas pela *World Gold Council*[1]*,* organização de desenvolvimento de mercado para indústria do ouro.

Observa-se que os valores da ST não oscilam em torno de um nível fixo. Ao invés disso, eles apresentam um comportamento crescente cujo período de variação se mantém constante à medida que o nível aumenta. A decomposição utiliza essas e outras propriedades para a identificação de componentes da série e a obtenção de índices que podem ser utilizados para predição de valores futuros (PARMEZAN, 2016).

[1] <https://www.gold.org/goldhub/data/gold-prices>

Figura 1 - Preço do ouro nos Estados Unidos



Fonte: Elaborada pelo autor.

Neste sentido, as séries temporais podem ser classificadas em:

- **Série temporal determinística:** a série temporal determinística é descrita pela função matemática (PARMEZAN, 2016): $Y = f(tempo);$
- **Série temporal não-determinística:** a série temporal não-determinística é descrita pela função matemática, onde ε é um tempo aleatório (PARMEZAN, 2016): $Y = f(tempo, \varepsilon);$
- **Série temporal contínua:** a série temporal contínua é quando a observação dos dados é ininterrupta durante um intervalo de tempo específico (BROCKWELL, 2001). Sendo denotada por: $X(t): t \in T, T = t : t_1 < t < t_2;$
- **Série temporal discreta:** a série temporal discreta é quando a observação dos dados é feita em intervalos fixos de tempo, geralmente iguais. Também tem a possibilidade de obter uma ST discreta a partir de uma contínua (BROCKWELL, 2001). Denotada por: $X(t): t \in T, T = t_1, \ldots, t_n.$

3.2.1 Tendência

Não existe uma definição precisa de tendência e diferentes autores usam este termo de diferentes formas. Podemos pensar em tendência como uma mudança de longo prazo no nível médio da série (EHLERS, 2009). A forma mais simples de tendência é:

$$X_t = \alpha + \beta_t + \varrho_t \quad (3.2)$$

Em que α e β são constantes a serem estimadas, enquanto ϱt denota um erro aleatório com média zero. O nível médio da série no tempo t é dado por mt = α + βt, também chamado de tendência (EHLERS, 2009). Essa componente envolve um comportamento de extensa duração, podendo ser tanto crescente, quanto descrente, assumindo uma grande variação de padrões. Ainda de acordo com Ehlers (2009), os crescimentos que se destacam são:

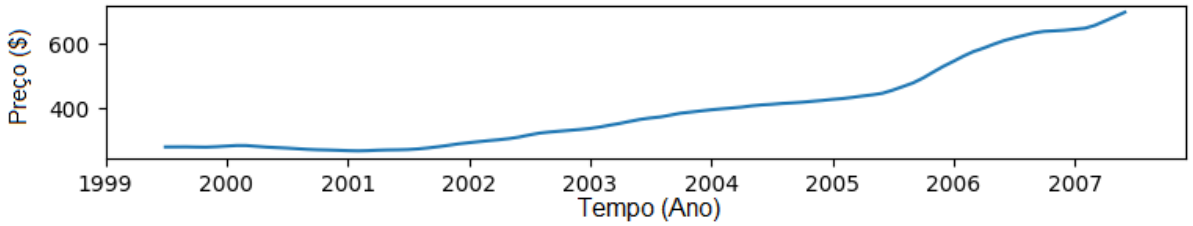
1. **Crescimento linear:** taxa de crescimento constante para os dados, obedecendo uma proporção linear;
2. **Crescimento exponencial:** aumento progressivo do percentual dos dados por um período;
3. **Crescimento amortecido:** ocorre quando a taxa de crescimento de dados futuros é menor que os atuais.

Neste sentido, identificar a componente de tendência aborda 3 objetivos (MORETTIN e TOLOI, 2006):

1. Avaliar o seu comportamento e incluí-lo em um modelo preditivo;
2. Remover o componente da série temporal para os demais componentes terem visibilidade;
3. Estimar o nível da série, ou seja, o valor ou faixa típica de valores que a variável pode assumir caso não tenha uma tendência crescente ou descrente a longo prazo.

Na **Figura 2** é mostrada a tendência da ST referente ao preço do ouro. A tendência permanece estável e com pouca alteração entre os anos de 1999 e 2002. Entre os anos de 2002 e 2005 o preço começa a ascender de forma uniforme. Posteriormente, o movimento ascendente aumenta.

Figura 2 - Linha de tendência da série de preços do ouro nos Estados Unidos



Fonte: Elaborada pelo autor.

3.2.2 Sazonalidade

Um comportamento que tende a se repetir em diferentes períodos nas séries temporais é conhecido como sazonalidade. Com a sazonalidade identificada, podemos removê-la aplicando um modelo aditivo na série temporal original ou, ainda, por um modelo multiplicativo, que divide os dados da série temporal original pela sazonalidade (PARMEZAN, 2016).

De acordo com Brocklebank e Dickey (2003), além de estar frequentemente relacionada às estações do ano, também pode acontecer a partir de causas naturais, econômicas e/ou sociais. Podemos observar essa particularidade em aumento no número de vendas em determinada época do ano (Natal e Páscoa, por exemplo).

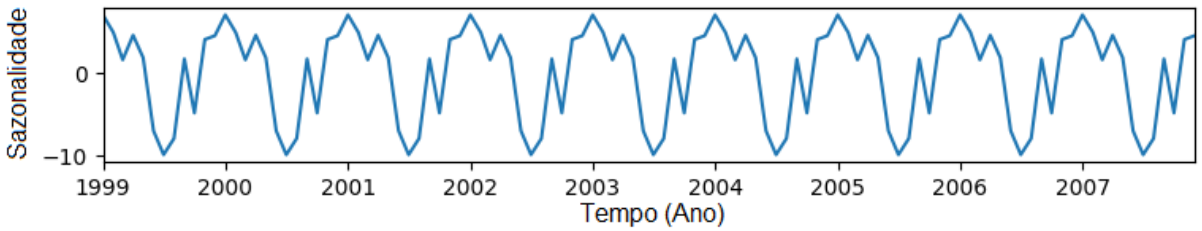
Para Parmezan (2016), é possível compreender aspectos relevantes do fenômeno observado por meio da explicitação do componente de sazonalidade. Entretanto, dependendo do campo de aplicação, a existência de sazonalidade pode dificultar o reconhecimento e a interpretação de movimentos não-sazonais peculiares em uma série.

A sazonalidade pode ser categorizada a partir das suas variações (EHLERS, 2009):

1. **Sazonalidade aditiva:** nesta categoria, a sazonalidade a série apresenta uma flutuação sazonal estável, sem considerar o nível global da série;
2. **Sazonalidade multiplicativa:** nesta categoria, a sazonalidade acontece quando o tamanho da flutuação sazonal varia de acordo com o nível global da série.

Na **Figura 3** é mostrada a sazonalidade da ST referente ao preço do ouro.

Figura 3 - Linha de sazonalidade da série de preços do ouro nos Estados Unidos



Fonte: Elaborada pelo autor.

3.2.3 Resíduo ou ruído

O resíduo ou ruído é representado na série temporal pelos movimentos aleatórios causados por fatos eventuais e inesperados, como é o caso de catástrofes naturais, guerras, greves e decisões governamentais intempestivas. Esses fatos não regulares e que também se repetem em um padrão particular, podem comprometer os resultados dos estudos (PARMEZAN, 2016).

Para Kirchgassner e Wolters (2007), identificar o componente residual é essencial tanto para removê-lo, quanto para verificar as variações cíclicas que podem acontecer no conjunto de dados categorizado como resíduo. A análise de séries temporais parte do pressuposto que a tendência e a sazonalidade não são influenciadas por eventos aleatórios, sendo resumidos por funções determinísticas do tempo, enquanto o resíduo é o que sobra após estimar os componentes de tendência e sazonalidade (PARMEZAN, 2016).

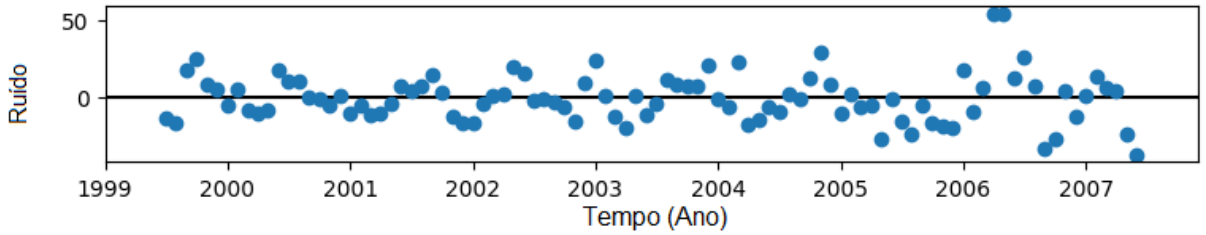
De acordo com Cowpertwait e Metcalfe (2009), o resíduo N de um instante de tempo t de uma série temporal pode ser definido a partir dos modelos clássicos de decomposição, sendo Z a série temporal e T e S como tendência e sazonalidade, respectivamente. Assim, temos as seguintes equações:

$$N_t = Z_t - (T_t + S_t) \tag{3.3}$$

$$N_t = \frac{Z_t}{(T_t \times S_t)} \tag{3.4}$$

Na **Figura 4** é novamente tratada da ST referente ao preço do ouro. O resultado é apenas as variações cíclicas e irregulares, após a remoção dos componentes de tendência e de sazonalidade.

Figura 4 - Linha de sazonalidade da série de preços do ouro nos Estados Unidos



Fonte: Elaborada pelo autor.

3.2.4 Estacionariedade

Estacionariedade define uma série que se desenvolve no tempo aleatoriamente, estando em equilíbrio com a média constante (MORETTIN e TOLOI, 2006).

3.3 MACHINE LEARNING APLICADO ÀS SÉRIES TEMPORAIS

Nas últimas duas décadas, os modelos de *machine learning* (ML) chamaram a atenção e estabeleceram-se como sérios concorrentes aos modelos estatísticos clássicos da comunidade de previsão (BONTEMPI *et al,* 2013). De acordo com Shalev-Shwartz e Ben-David (2014), o termo “aprendizado de máquina” refere-se à detecção automatizada de padrões nos dados e vem sendo comumente usada para tarefas que requerem a extração de informações de grandes conjuntos de dados.

No campo de séries temporais, numerosos estudos foram publicados sobre modelos de *machine learning*, com desempenhos relativamente melhores em comparação às técnicas clássicas de previsão de séries temporais (SEZER *et al*, 2020). Isso porque, de acordo com Parmezan (2016), os modelos de *machine learning* são orientados a dados, sem a necessidade de compreensão do conjunto

de dados no qual o algoritmo precisará passar por um treinamento. Tal característica é uma vantagem do ML em comparação aos modelos estatísticos.

De acordo com Sérgio (2017), os modelos de *machine learnin*g são:

- **Redes Neurais *Feedforward*:** sistemas paralelos distribuídos compostos por unidades de processamento simples, que calculam funções matemáticas (HAYKIN *apud* SÉRGIO, 2017);
- ***Deep Learning*:** aprendizagem não-supervisionada das representações dos dados, para pré-treinar cada uma das camadas das redes neurais, destacando-se entre os modelos as redes neurais conhecidas como *Deep Belief Networks* (DBN) e *Stacked Denoising Autoencoders* (SDAE);
- ***Support Vector Regression*:** modelo de aprendizado baseado na Teoria da Aprendizagem Estatística, proposto por Vapnik e Chervonenkis (VAPNIK *apud* SÉRGIO, 2013).

3.3.1 Random Forest

O Random Forest pode gerar milhares de *Decision Trees* que atuam como funções de regressão, a média dos resultados de todas as árvores de decisões é o resultado final do RF. *Decision Trees* é um modelo estatístico não paramétrico, uma árvore composta por nós de decisão, que avaliam cada amostra alimentada por uma função de teste, passadas para diferentes ramificações, com base nas suas características e nós folhas, os quais são os últimos nós das ramificações (LI *et a*l, 2018).

Durante o treinamento, os dados de entrada são divididos pelo algoritmos, de forma a otimizar os parâmetros das funções de divisão para que ajustem-se ao conjunto S_n, representado conforme abaixo, em que n é o número de observações, X é a amostra de entrada contendo m características, $X = \{x_1, x_2, \ldots x_m\}$ e Y é o escalar de saída:

$$S_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}, X \in R^m, Y \in R \quad (3.5)$$

A árvore faz a divisão ideal entre as variáveis, processo aplicado para cada nó e repetido até que um nó folha seja atingido. Ao final do processo de treinamento, uma função de predição $\hat{h}\ (X, S_n)$ é construída sobre S_n (LI *et al*, 2018).

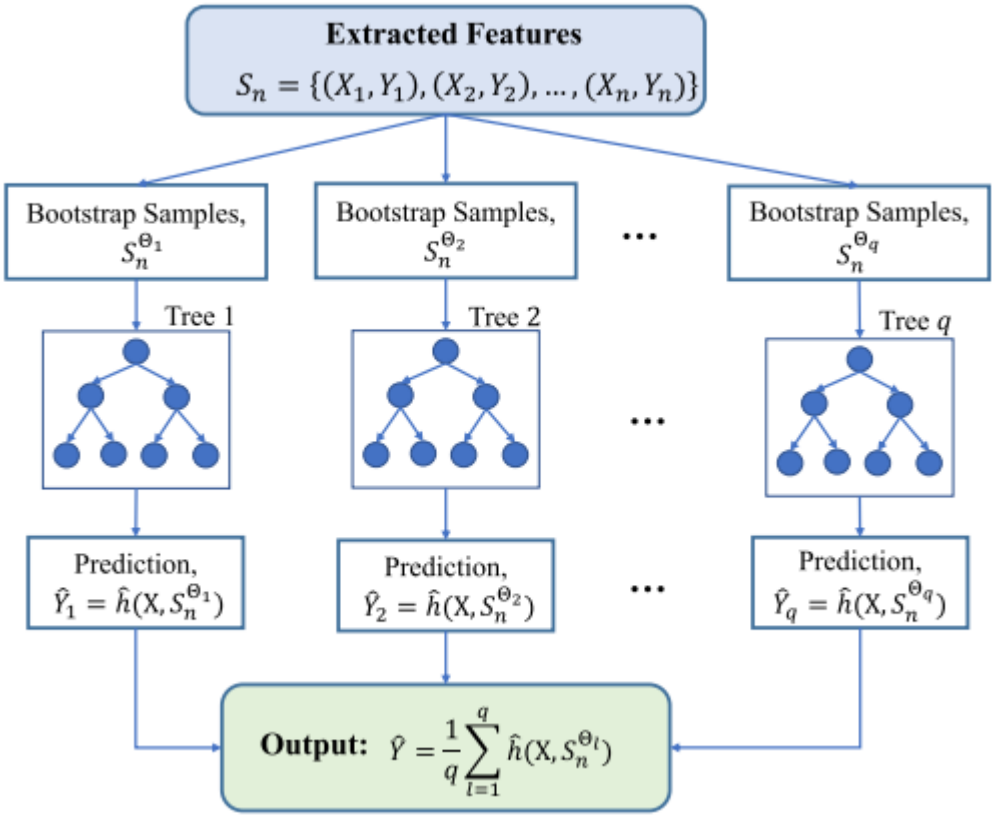
> A random forest é um classificador, que consiste em uma coleção de classificadores estruturados em árvore $\{h(x, \Theta_k), k = 1,...\}$, em que $\{\Theta_k\}$ são vetores aleatórios, distribuídos de maneira idêntica e independente, de forma que cada árvore lança um voto único para a classe mais popular na entrada *x*.(BREIMAN, 2001)

Para combinar todas as árvores geradas e diminuir a variância relacionada à previsão, é utilizado o algoritmo *bagging* (ou *bootstrap aggregation*), uma vez que a RF pode ser construída por amostragem aleatória de um subconjunto de recursos, para cada árvore de decisão ou, ainda, também por amostragem aleatória de um subconjunto de dados de treinamento para cada árvore de decisão (LI *et a*l, 2018).

O processo amostral coletado de forma aleatória é conhecido por '*bootstrap*'. Uma amostra *bootstrap* é obtida a partir da seleção aleatória de n observações com reposição de Sn, em que cada observação tem a probabilidade de 1/n de ser a escolhida. O algoritmo de *bagging* seleciona diversas amostras *bootstrap* ($S_n^{\Theta 1}$, …, $S_n^{\Theta q}$), aplicando o algoritmo de decisão de árvore anterior às amostras, para a construção de uma coleção de q árvores de predição $\hat{h}\ (X , S_n^{\Theta 1})$, …, $\hat{h}\ (X , S_n^{\Theta q})$. O conjunto produz *q* saídas correspondentes a cada árvore, $\hat{Y}_1 = \hat{h}\ (X, S_n^{\Theta 1})$, $\hat{Y}_2 = \hat{h}\ (X , S_n^{\Theta 2})$, …, $\hat{Y}_q = \hat{h}\ (X , S_n^{\Theta q})$ (LI *et al*, 2018).

Na sequência, é realizada a agregação a partir da média das saídas de todas as árvores. Em consequência, a estimativa $\hat{Y}$ da saída pode ser obtida por onde $\hat{Y}_l$ na saída da *l* a arvore, e $l = 1, 2, ...,q$. A estrutura do uso de regressão de RF para previsão é ilustrada na **Figura 5** (LI *et al*, 2018).

Figura 5 - Ilustração da construção do Random Forest



Fonte: LI *et al,* 2018.

Um benefício de utilizar o algoritmo de uso de *bagging* é de evitar a correlação de diferentes árvores, uma vez que existe a possibilidade de ampliar a diversidade de árvores, com crescimento a partir de subconjuntos de dados de treinamentos, criados em RF. Neste sentido, alguns dados podem ser utilizados mais de uma vez no treinamento, enquanto outros podem nunca chegar a serem utilizados. Dessa forma, maior estabilidade de RF é alcançada graças ao uso do algoritmo de *bagging*, tornando a RF mais robusta diante de poucas alterações nos dados de entrada (LI *et al*, 2018).

Mais uma vantagem do *bagging* é o que diz respeito à sua imunidade aos ruídos, uma vez que gera árvores não correlacionadas a partir de diferentes amostras de treinamento. Um preditor fraco pode ser sensível ao ruído, mas a média de várias árvores de decisão não correlacionadas, podem diminuir de maneira significativa, a sensibilidade ao ruído. Por isso, uma característica essencial da RF é que as árvores dentro dela crescem sem poda, de maneira a torná-las ainda mais leves, do entendimento computacional (LI *et al*, 2018).

3.4 MÉTRICAS DE AVALIAÇÃO

Em predição de ST, é necessário medidas de erro que qualificam a diferença entre os valores reais e os valores preditos pelo modelo escolhido. Entre as métricas mais utilizadas estão MASE, MAE, RMSE e MAPE que serão utilizadas neste trabalho e estão descritas a seguir.

- **MASE:** *Mean Absolute Scaled Erro*r é calculado dividindo o erro médio por um fator de escala. Esse fator de escala depende do valor da sazonalidade,*M*, que é selecionado com base na frequência de previsão. Um valor mais baixo indica um modelo mais preciso (AMAZON, 2022).

$$\text{MASE} = \text{mean}\left(\frac{|e_j|}{\frac{1}{T-m}\sum_{t=m+1}^{T}|Y_t - Y_{t-m}|}\right) = \frac{\frac{1}{J}\sum_j |e_j|}{\frac{1}{T-m}\sum_{t=m+1}^{T}|Y_t - Y_{t-m}|} \quad (3.6)$$

- **MAE:** O *Mean Absolute Error* calcula a média das diferenças absolutas entre os reais e os previstos. Um valor mais baixo indica um modelo mais preciso (LI *et al*, 2018).

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (3.7)$$

- **RMSE:** O *Root Mean-Square Error* calcula a raiz quadrada da média de erros ao quadrado. Um valor mais baixo indica um modelo mais preciso (LI *et al*, 2018).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad (3.8)$$

- **MAPE:** *Mean Absolute Percentage Error* calcula o valor absoluto do erro percentual entre os valores observados e previstos para cada unidade de tempo e, em seguida, calcula estes valores. Um valor mais baixo indica um modelo mais preciso (AMAZON, 2022).

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \quad (3.9)$$

4. TRABALHOS RELACIONADOS

Com base no tema deste trabalho, foram buscadas pesquisas que utilizassem séries temporais e modelos preditivos, com o propósito de estudar os modelos de predição e seus desempenhos para ajudar na escolha de um modelo

para este presente trabalho. Essa seção apresenta três trabalhos abordando estudos sobre séries temporais, e três trabalhos que comparam diferentes modelos preditivos.

4.1 PREDIÇÃO DE SÉRIES TEMPORAIS POR SIMILARIDADES

Parmezan (2016) buscou, a partir de 95 conjuntos de dados reais, realizar experimentos computacionais para avaliar as questões que impactam a qualidade dos resultados de Séries Temporais por similaridade. A partir de uma revisão sistemática de conceitos e avanços científicos já realizados na área de estudos de ST, a hipótese central do trabalho consiste em confirmar que os métodos para predição, baseados em similaridade, podem prover resultados competitivos em relação aos obtidos com a aplicação de métodos estatísticos estado-da-arte (PARMEZAN, 2016).

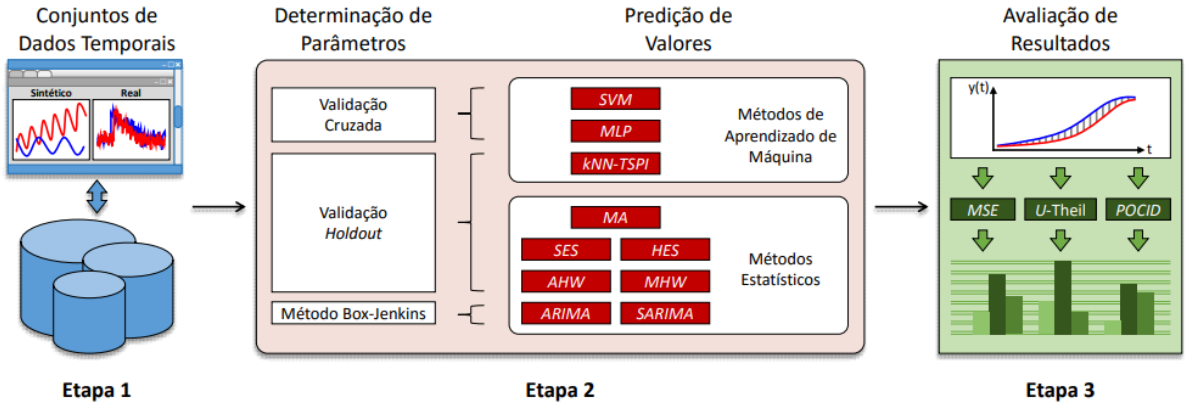
Assim, a partir de uma análise de métodos tradicionais que consideram, também, o *machine learning* como suporte à criação de modelos não-paramétricos para a predição de ST, o autor buscou investigar, a partir de adaptações já feitas de métodos da área de ML, a superação de questões temporais, consideradas restrições dentro da esfera dos métodos tradicionais de ML.

Com a investigação de métodos que foram originados a partir *machine learning*, o autor utilizou métodos baseados em similaridade a partir de uma subsequência de referência e com auxílio de uma medida de distância, as k subsequências mais similares dentro de uma determinada série, usando os valores seguintes das subsequências como entrada para uma função de predição e, assim, calcular o valor futuro (PARMEZAN, 2016).

Neste sentido, a pressuposição central do trabalho do autor consiste em demonstrar que, com a combinação adequada de invariância à amplitude, ao deslocamento e, da recentemente proposta invariância de complexidade (BATISTA *et al*, 2014), associada a uma política para evitar casamentos triviais proposta pelo autor, é possível alcançar predições mais assertivas e significativas.

Deste modo, o autor propôs uma nova extensão do algoritmo k-N*earest Neighbors - Time Series Prediction with Invariances* (kNN-TSPI) (PARMEZAN; BATISTA, 2016), em que incorpora 3 técnicas para obter a invariância à amplitude e deslocamento, invariância à complexidade e tratamento de casamentos triviais.

Figura 6 – Configuração experimental



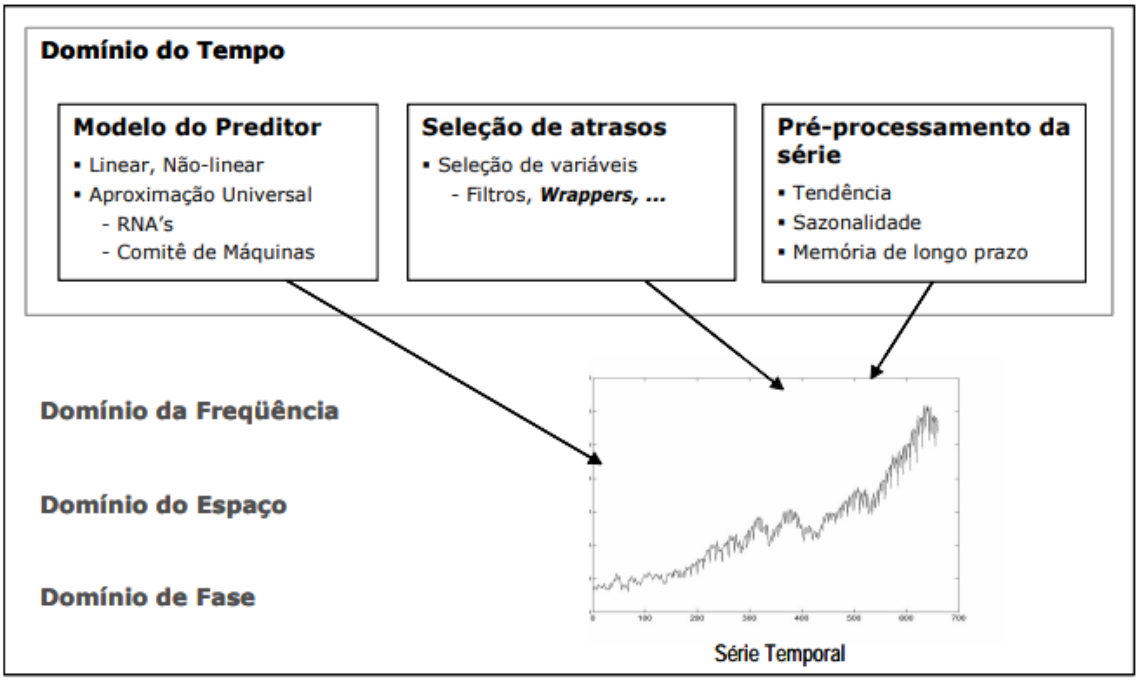
Fonte: PARMEZAN, 2016.

Seguindo o protocolo experimental ilustrado na **Figura 6** e a partir de uma análise comparativa de 95 conjuntos de dados, foram confrontados 10 algoritmos para a predição de valores, sendo 7 deles usando a abordagem paramétrica, e 3 de acordo com a abordagem não-paramétrica. Para verificar a acurácia das predições, foram utilizadas três métricas: *Mean Squared Error* (MSE), *Prediction Of Change In Direction* (POCID) e *Estatística de U-Theil.* Assim, os resultados da avaliação experimental utilizando séries reais e artificiais concluiu que o kNN-TPSI é um método mais competitivo em relação aos métodos SARIMA e SVM (Máquina de Suporte Virtual), uma vez que o algoritmo com invariâncias é consideravelmente mais simples de compreender, codificar e ajustar (PARMEZAN, 2016).

4.2 COMITÊ DE MÁQUINAS EM PREDIÇÃO DE SÉRIES TEMPORAIS

Villanueva (2006) buscou utilizar o comitê de máquinas (HAYKIN, 1999) para resolver problemas de regressão não-linear de dados, a partir de redes neurais artificiais com aprendizado supervisionado *off-line*. Com uma proposta para superar o desempenho de uma máquina de aprendizado operando de forma isolada, a motivação da sua pesquisa pode ser sustentada pela **Figura 7** e consiste em uma forma de tratar questões problemáticas relacionadas à predição de séries, a partir de abordagens já utilizadas.

Figura 7 - Formas de abordar o problema de predição de séries temporais



Fonte: VILLANUEVA, 2006.

Deste modo, seu trabalho foi dividido em dois objetivos, utilizando metodologias distintas para sua validação. Para o estudo de formas computacionais que permitam agrupar mais de uma máquina de aprendizado e, assim, melhorar seu desempenho, o autor propôs o Comitê de Máquinas: *ensemble* e mistura de especialistas. Um *ensemble* permite combinar as respostas de várias propostas de máquinas de aprendizado com o intuito de obter uma resposta global. Cada proposta atenderia o total do problema e o resultado seria uma espécie de consenso dos participantes (VILLANUEVA, 2006).

Para o caso de mistura de especialistas, além da combinação das propostas de máquinas de aprendizado, denominadas especialistas, resolvendo diretamente o problema, existe uma outra máquina que aprende a discriminar aspectos do problema e alocar para cada um deles um único ou um subconjunto de especialistas (VILLANUEVA, 2006).

Para o seu segundo objetivo, o autor abordou um dos tipos de seleção de variáveis chamado envoltório (GUYON e ELISSEEFF, 2003), tendo como preditor uma rede neural MLP (*Multilayer Perceptron*) (VILLANUEVA, 2006) e utilizando técnicas de seleção de variáveis aplicadas ao problema de predição de séries

temporais: Seleção Progressiva - SP (do inglês *Forward Selection*) e Poda Baseada em Sensibilidade - PBS (do inglês *Sensitivity Based Pruning*).

Em relação ao Comitê de Máquinas, foram utilizadas duas métricas para verificar a acurácia das predições: *Root Mean Squared Error* (RMSE) e *Mean Absolute Error* (MAE). Assim, o autor observou um ganho significativo na predição ao combinar modelos preditores, seja em forma de ensemble ou mistura de especialistas, em comparação ao resultado obtido com apenas um único modelo de predição.

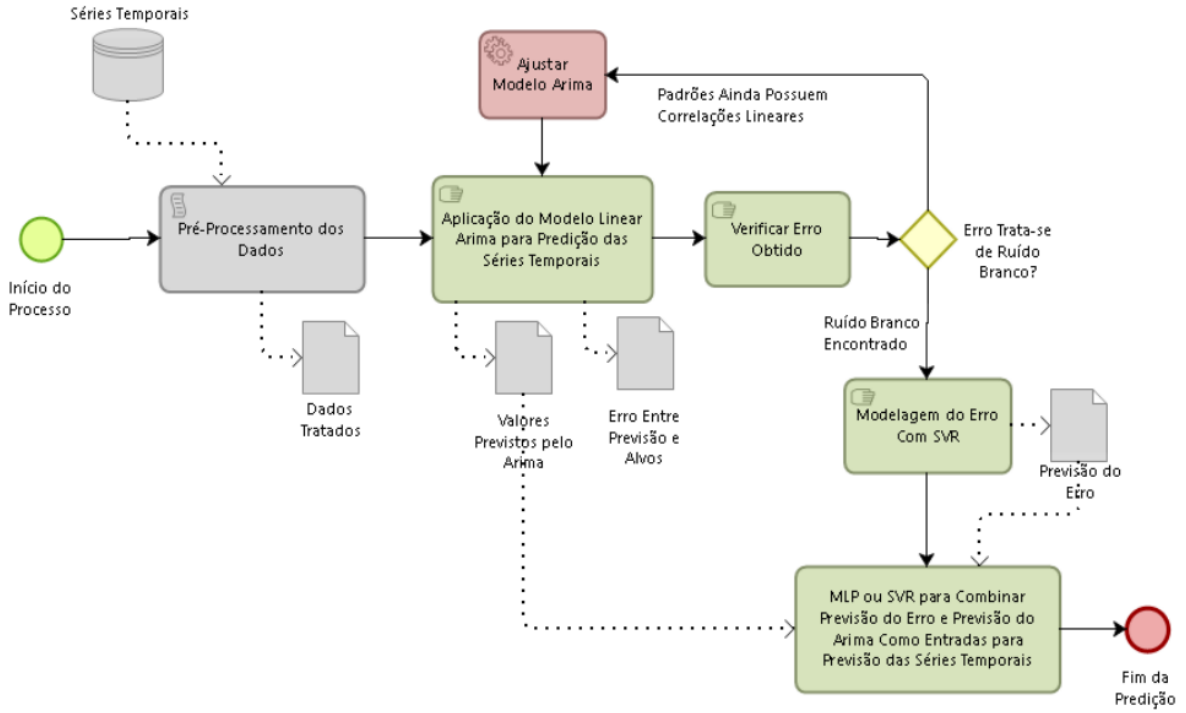
Já o resultado da abordagem envoltório, utilizando os métodos de Seleção Progressiva (SP) e Poda Baseada em Sensibilidade (PBS), trouxe análises comparativas em relação aos dois modelos. Enquanto a Seleção Progressiva (SP) trouxe resultados melhores em comparação à Poda Baseada em Sensibilidade (PBS), o número de variáveis selecionadas via SP foi menor em todos os casos, quando comparado ao número de variáveis selecionadas via PBS (VILLANUEVA, 2006).

4.3 UMA ABORDAGEM HÍBRIDA PARA A PREDIÇÃO DE SÉRIES TEMPORAIS

Lucena (2016) buscou utilizar uma abordagem híbrida para previsão de séries temporais, buscando atingir maior acurácia em comparação às previsões individuais de modelos combinados (LUCENA, 2016).

Sua abordagem consiste em combinar previsões pela modelagem do erro da previsão a partir de uma determinada ST, em que o erro é a diferença entre a previsão de um modelo utilizado, e outro valor alvo a ser alcançado. Com uma literatura extensa sobre o assunto, o autor apoiou-se na combinação de modelos distintos como alternativa para previsão de séries temporais com origem em observações do mundo real. A abordagem híbrida é representada na **Figura 8**.

Figura 8 - Estratégia Híbrida Implementada



Fonte: LUCENA, 2016.

Assim, pelas observações possuírem diferentes características que impactam no resultado, a combinação de modelos de naturezas diferentes tende a capturar, de maneira mais ampla, tais características (LUCENA, 2016), uma vez que a utilização de métodos de maneira individual, não é capaz de abordar essas diferentes características de um mesmo objeto de observação.

Dessa forma, a partir da separação dos padrões apresentados entre as correlações não lineares, das correlações lineares, utilizando a modelagem do erro para buscar a melhor utilização de preditores (ZHANG, 2003). Entretanto, o autor buscou uma proposta alternativa à literatura da área, que propõe a combinação das previsões por meio da soma ou média. Lucena utilizou a técnica de aprendizagem de máquina SVR e rede neural artificial MLP para combinar as previsões. Para verificar a acurácia das predições, foram utilizadas seis métricas: *Mean Squared Error* (MSE), *Mean Absolute Percentage Error* (MAPE), *Index of Agreement* (IA), *Prediction Of Change In Direction* (POCID), *Average Relative Variance* (ARV) e Estatística de Theil.

Neste sentido, a análise dos experimentos demonstrou que os resultados empíricos obtidos pelo uso da abordagem híbrida, por meio da sua modelagem pela rede neural MLP e a técnica SVM, aumentaram a acurácia das predições (LUCENA, 2016). Entretanto, o autor observou que as métricas utilizadas para medir a acurácia das predições, poderiam gerar dificuldades na análise e, assim, dificultar a evolução das métricas abordadas nos diferentes modelos aplicados. Sendo assim, em alguns casos, as abordagens combinadas podem acabar perdendo em alguma das métricas utilizadas em modelos individuais (LUCENA, 2016).

O estudo observou também que, ao separar as correlações lineares das não lineares, mesmo que o erro encontrado a partir da diferença entre a previsão e o valor real por um modelo linear venha a possuir características de ruído branco, existe uma grande possibilidade de diferentes fatores (como falta de atributos, inconsistências ou até a limitação de um modelo linear) possam fazer com que, mesmo se tendo passado pela verificação que detecta se o erro tem características de ruído branco, estes resíduos não sejam apropriados para modelagem (LUCENA, 2016).

4.4 OUTROS TRABALHOS

4.4.1 Análise de séries temporais em epidemiologia: uma introdução sobre os aspectos metodológicos

Latorre e Cardoso (2001) buscaram, a partir de diferentes modelos estatísticos, analisar séries temporais dentro da área de epidemiologia, para contribuir com os estudos sobre séries temporais aplicadas à área da saúde. Neste artigo, as autoras mostram as possibilidades de obter uma série temporal com diferentes modelos estatísticos, a partir de uma série de estudos com: modelo de regressão polinomial, modelos autorregressivos e modelos lineares generalizados.

Os modelos foram utilizados para analisar tendências de mortalidade infantil e seus componentes no Município de Guarulhos entre 1979 e 1998, evolução da mortalidade por desnutrição em idosos nas regiões metropolitanas dos estados do Rio de Janeiro (RMRJ) e São Paulo (RMSP), para verificar suas tendências entre 1980 e 1996 e análise da associação entre mortalidade por idosos e associação atmosférica na cidade de São Paulo entre 1994 a 1997, respectivamente.

Em todos os estudos foi possível observar e analisar tendências, que contribuíram para identificar períodos de maior ou menor incidência, a partir do contexto vivido naquele período. Os autores esperavam mostrar as inúmeras aplicações da análise de séries temporais, estimulando-se a busca de técnicas estatísticas adequadas (LATORRE e CARDOSO, 2001).

4.4.2 Leishmaniose visceral no Piauí, 2007-2019: Análise ecológica de séries temporais e distribuição espacial de indicadores epidemiológicos e operacionais

Chaves *et al* (2022) analisaram o contexto epidemiológico da leishmaniose visceral no Piauí em 13 anos selecionados. O estado apresenta alta ocorrência de casos de LV, e fatores como desigualdades socioeconômicas e condições ambientais têm contribuído para o aumento dos casos da doença (CHAVES *et al,* 2022). Neste artigo, os autores utilizaram as análises de séries temporais para observar a tendência de crescimento dos casos da doença em diferentes regiões.

Foi observado no estudo aplicando a análise ecológica de séries temporais, que os índices elevados de LV em diferentes regiões do estado do Piauí, também podem ser explicados pela ausência das medidas preventivas e de controle de parte dos municípios, preconizadas pelo Ministério da Saúde, além da carência de recursos humanos capacitados no nível local (*apud* FARIAS *et al,* 2019).

Diante deste cenário, os estudos utilizando séries temporais concluem que a LV se mantém como uma doença negligenciada no Piauí, com preocupante tendência crescente da incidência e elevado percentual de casos em tratamento com evolução ignorada (CHAVES *et al*, 2022), observado a partir da aplicação da análise ecológica de séries temporais.

4.4.3 Modelos de séries temporais e gráficos de controle estatístico aplicados a indicadores de vigilância epidemiológica do ministério da saúde

Veloso (2018) buscou demonstrar a aplicação de modelos de séries temporais e gráficos de controle estatístico, a partir da análise exploratória de exames para detecção de doenças do vírus Influenza, do vírus da Febre Amarela,

vírus da Zika e Chikungunya. Seu estudo focou nos impactos no vírus Influenza e foi analisado o Indicador referente ao total de exames com resultado positivo para Influenza ao longo de uma Semana Epidemiológica no município de Curitiba-PR.

Foi possível perceber certo padrão sazonal no comportamento da série histórica deste Indicador. O modelo de Séries Temporais que se mostrou mais adequado para a série do Indicador foi o SARIMA (1,0,0) × (0, 1, 1), conforme os critérios de seleção AICC e BIC e análise de resíduo (VELOSO, 2018).

Dentro de um contexto epidêmico, o estudo contribui com o propósito central da Vigilância Epidemiológica, que é o fornecimento de orientação técnica permanente e atual para profissionais de saúde terem a possibilidade de diagnosticar uma situação epidêmica inicial e adoção imediata de medidas de controle (VELOSO, 2018).

4.5 CONSIDERAÇÕES

Considerando-se o objetivo deste trabalho, é necessário escolher um processo de predição de valores, além de escolher um modelo de predição e métricas de avaliação. Os trabalhos relacionados citados possuem grande carga teórica sobre o assunto, além de apresentarem discussões e comparações entre modelos de predição conforme o **Quadro 1**.

Quadro 1 - Comparativo de trabalhos relacionados.

Trabalhos	Demonstra predição de séries temporais?	Faz comparação de algoritmos de predição?	Modelos paramétricos	Modelos não-paramétricos	Modelo que obteve melhor resultado	Métricas utilizadas
PARMEZAN, 2016	Sim	Sim	MA, SES, HES, AHW, MHW, ARIMA e SARIMA	SVM, MLP e kNN-TSPI	kNN-TSPI	MSE, POCID e TU
VILLANUVEA, 2006	Sim	Sim	ARIMA	MLP	MLP	RMSE e MAE
LUCENA, 2016	Sim	Sim	ARIMA	SVR, MLP (ARIMA, SVR) e SVR (ARIMA, SVR)	MLP	MSE, MAPE, IA, POCID, ARV e TU
LATORRE e CARDOSO, 2001	Não	Não	-	-	-	-
CHAVES *et al*, 2022	Não	Não	-	-	-	-
VELOSO, 2018	Sim	Não	SARIMA	-	-	-

Fonte: Elaborada pelo autor.

O trabalho do Parmezan demonstrou-se mais completo e agregador pelo processo de predição proposto, pelos diversos modelos testados e avaliados e a grande carga teórica sobre o assunto.

Os trabalhos relacionados apresentaram diversos modelos e métricas, que por fim me levou a escolher estudar modelos não-paramétricos, em busca de um

que se melhor encaixe na proposta deste trabalho, e utilizar métricas para avaliar os resultados.

Este trabalho se diferencia pelo tipo do *dataset* utilizado, a empresa segue a metodologia ágil para gerenciar seus projetos, ao analisar e predizer, esses dados poderão ser usados para tomar decisões utilizando a variável mais importante para esse tipo de metodologia: o tempo.

5. DESENVOLVIMENTO

Para cumprir o objetivo deste trabalho, o desenvolvimento será dividido em 3 etapas:

Primeira etapa: Preparação das séries temporais.

Segunda etapa: Testes e análise dos resultados de todos modelos preditivos disponibilizados pelo PyCaret.

Terceira etapa: Análise dos resultados do modelo preditivo com melhor desempenho com novos dados.

5.1 PREPARAÇÃO DAS SÉRIES TEMPORAIS

A empresa JExperts[2], atuante no mercado de desenvolvimento de software no Brasil há 20 anos, disponibilizou um conjunto de 10 mil dados, do período de 2018 a 2022 para experimentos, coletados do módulo Agile da plataforma para escritórios corporativos de gestão de portfólios, programas e projetos, Channel. Os dados pertencem a projetos internos da empresa, referentes à manutenção e migração das tecnologias usadas na plataforma. Os dados são correspondentes às tarefas da equipe de desenvolvimento e descritas a partir de atributos. Para este estudo, serão utilizados os atributos de “nome”, “descrição”, “tag” (palavra chave para facilitar na busca de tarefas do mesmo conjunto, por exemplo: BACK-END, FRONT-END), “data de criação” e “tempo” (tempo utilizado para concluir a tarefa) da tarefa. As informações sobre “nome”, "descrição" e “tag” serão usados para agrupar os dados, enquanto dados de “data de criação” e “tempo” serão utilizados para realizar a predição.

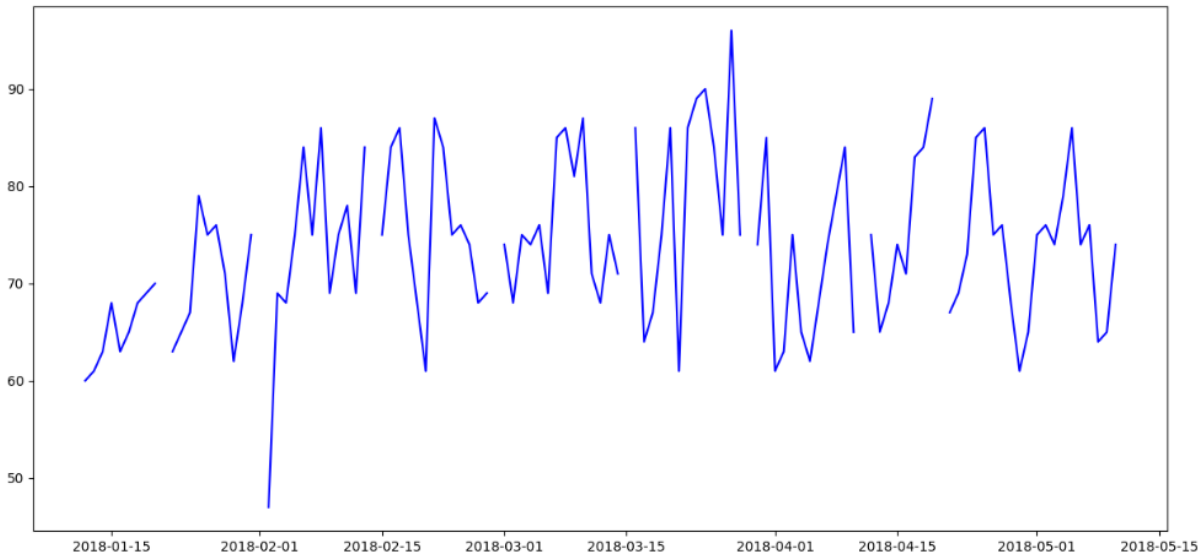
[2] <https://www.jexperts.com.br>

Foi necessário extrair *datasets* deste conjunto de dados e analisar cada tarefa para, assim, desenvolver uma série temporal de dados com características iguais ou semelhantes, conforme demonstrado na **Figura 1**. No exemplo, a série temporal do ouro é sempre o seu preço, portanto, não haveria coerência na construção de uma série temporal com todos os 10 mil dados disponibilizados, em que a maioria não possui correlação entre si.

A primeira etapa do processo consistiu em separar os dados pelas tags “BACK-END” e “FRONT-END”. Em seguida, os dados foram agrupados por semelhança entre nomes e descrições, utilizando postgreSQL e análise manual para reorganização de dados, a partir do detalhamento dos atributos de “nome” e “descrição”. Após a formação dos *datasets*, apenas três tiveram um conjunto considerável, com mais de 100 dados cada, correspondentes às tarefas de desenvolvimento de “CRUD”, “Validadores” e “Modal”. CRUD são as operações básicas de criar, consultar, atualizar e excluir dados em bases de dados relacionais, Validadores é uma camada lógica para validar os dados enviados para a execução das operações do CRUD e Modal é uma janela que abre sobre o conteúdo da página.

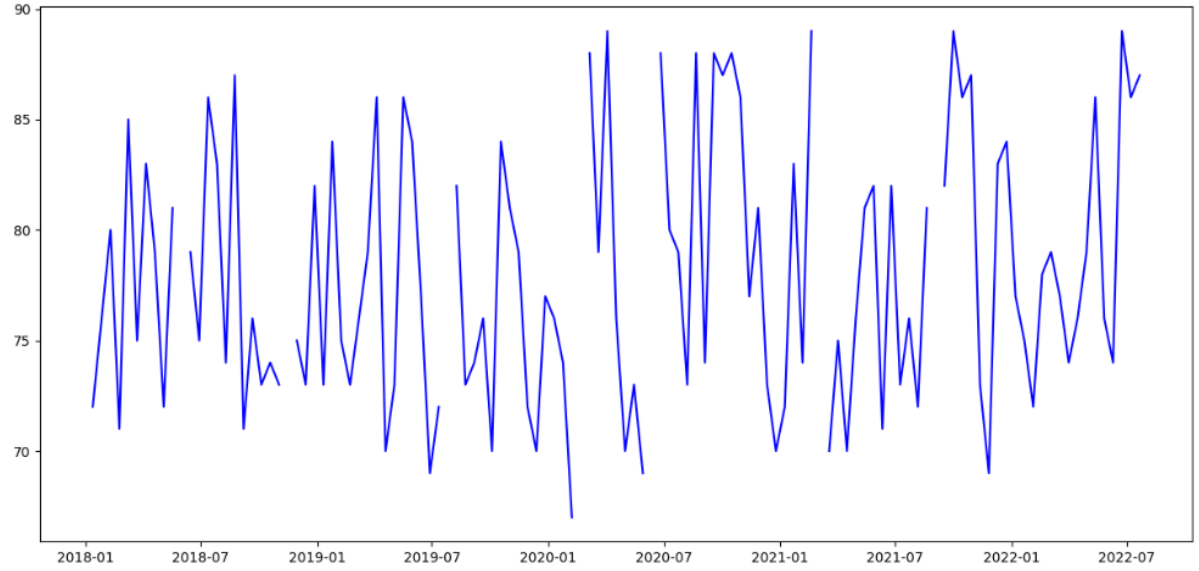
Em um contexto de trabalho utilizando metodologia ágil e organizando as tarefas a partir de *sprints* com duração de duas semanas, a periodicidade das séries temporais serão correspondentes. As tarefas são criadas no último dia da sprint anterior, geralmente às sextas-feiras. Entretanto, nem todas as tarefas serão criadas no mesmo dia da semana. Por isso, as tarefas criadas durante as *sprints* terão sua data de criação alterada para comportar a periodicidade da ST, conforme demonstrado nas **Figuras 9**, **10** e **11**.

Figura 9 - Série temporal Modal incompleta



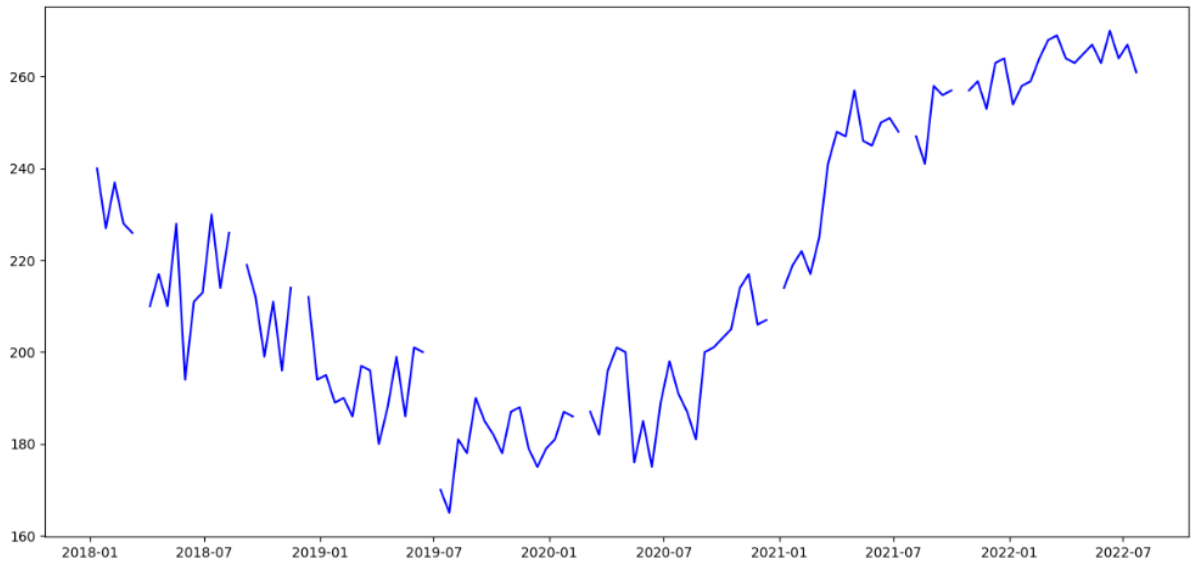
Fonte: Elaborada pelo autor.

Figura 10 - Série temporal Validadores incompleta



Fonte: Elaborada pelo autor.

Figura 11 - Série temporal CRUD incompleta



Fonte: Elaborada pelo autor.

5.1.1 Dados faltantes

Como observado nas **Figuras 9**, **10** e **11**, as séries temporais estão incompletas, uma vez que nem toda sprint possui uma tarefa que seja compatível com o período das séries apresentadas, dificultando o processo de construção e análise das ST. Para sanar tal obstáculo, foram testados 6 métodos utilizados para completar os valores faltantes: *Mean imputation, Median imputation, Last Observation Carried Forward, Next Observation Carried Backward,* Linear *interpolation e Spline interpolation*. De acordo com Koech (2022):

- ***Mean imputation*:** completa os valores faltantes com a média de todos os dados;
- ***Median imputation*:** completa os valores faltantes com a mediana de todos os dados;
- ***Last Observation Carried Forward*:** completa o valor faltante com o valor anterior;
- ***Next Observation Carried Backward*:** completa o valor faltante com o próximo valor.

- ***Linear interpolation*****:** completa com valores desconhecidos, assumindo relação linear dentro de um intervalo de pontos de dados;
- ***Spline interpolation*****:** completa com valores que minimizam a curvatura geral, obtendo uma superfície lisa passando pelos pontos de entrada.

Tabela 1 - Valores sequentes da série temporal incompleta Modal

Data	**Minutos**
05/10/2018	75
19/10/2018	
02/11/2018	47

Fonte: Elaborada pelo autor.

Tabela 2 - Valores obtidos pelos métodos para completar valor faltante da **Tabela 1**

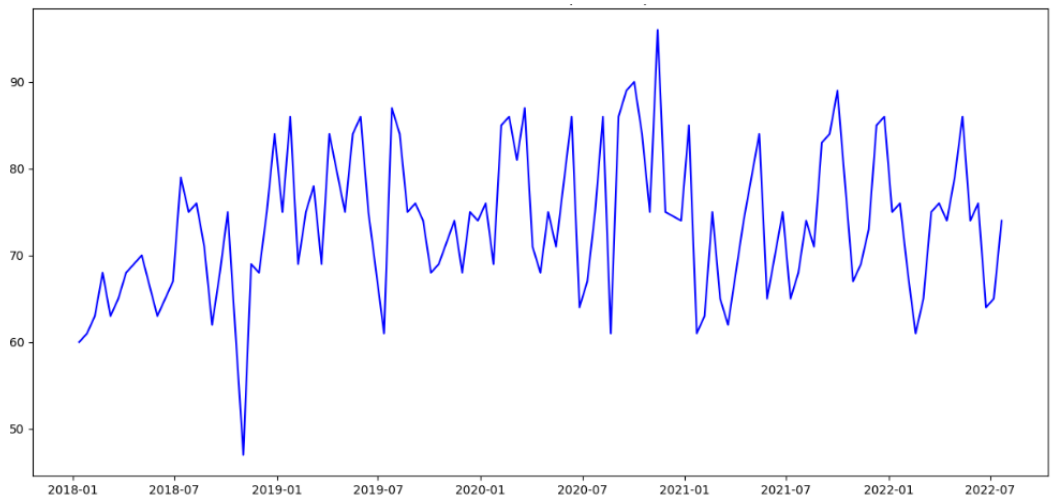
Métodos	**Minutos**
Mean imputation	73.6
Median imputation	74
Last Observation Carried Forward	75
Next Observation Carried Backward	45
Linear interpolation	61
Spline interpolation	61

Fonte: Elaborada pelo autor.

A **Tabela 1** apresenta 3 valores da série temporal Modal para ilustrar os resultados obtidos na **Tabela 2**. Os métodos Linear *interpolation* e *Spline interpolation* tiveram resultados idênticos e positivos, destacando-se nas três séries temporais. Por fim, o método *Spline interpolation* foi escolhido para completar os dados faltantes, uma vez que a interpolação linear e *spline* tende a fornecer valores de imputação, assim gerando as séries temporais completas ilustradas nas **Figuras**

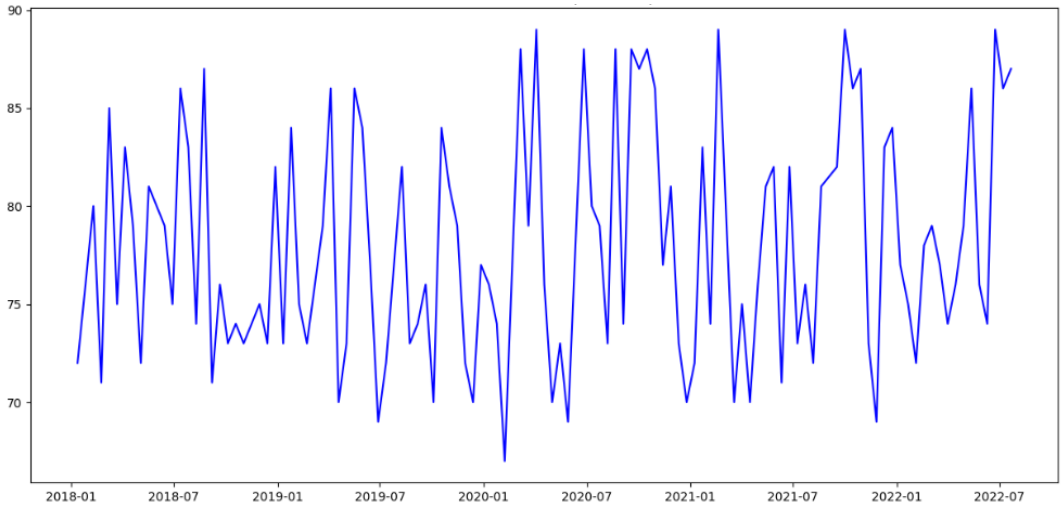
12, **13** e **14**. Os dados imputados possuem o Erro Quadrado Médio, sendo consideradas as melhores técnicas nesse nível (KOECH, 2022).

Figura 12 - Série temporal Modal



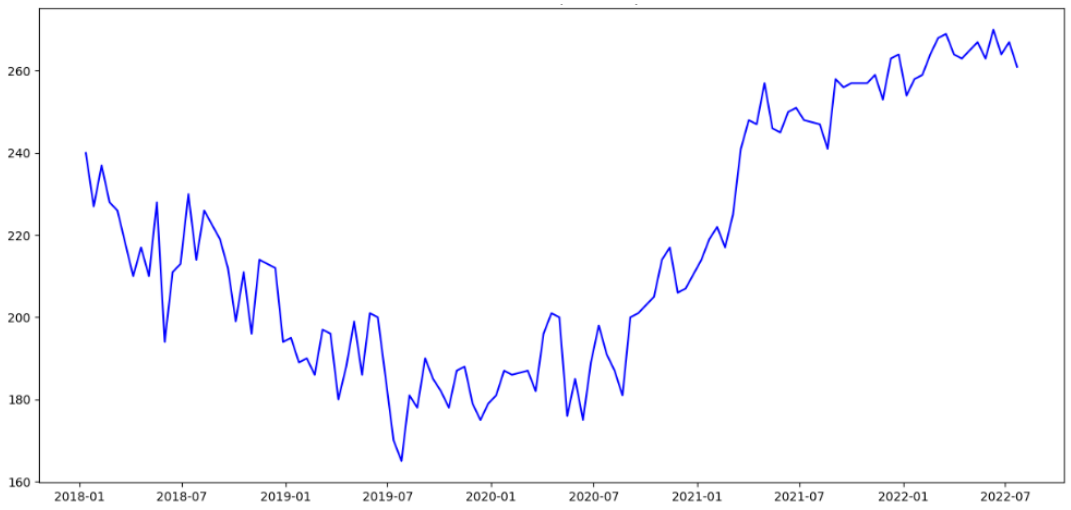
Fonte: Elaborada pelo autor.

Figura 13 - Série temporal Validadores



Fonte: Elaborada pelo autor.

Figura 14 - Série temporal CRUD



Fonte: Elaborada pelo autor.

5.2 TESTES

Após a montagem de três séries temporais, é necessária a escolha do modelo ideal, que melhor se adequa às características das séries atuais e futuras. A primeira característica corresponde ao tamanho das ST, uma vez que o modelo escolhido precisa apresentar resultados positivos em séries com poucos dados, já que o volume utilizado neste estudo são cerca de 110 dados. Além disso, o modelo também precisa saber lidar com a redução de erros, por serem STs pequenas, os outliers podem atrapalhar com facilidade e comprometer os resultados.

O ambiente utilizado para desenvolvimento foi o Jupyter notebook, utilizando a linguagem Python. Para os testes e predição dos dados, foi utilizado o módulo de Séries Temporais da biblioteca PyCaret[3], que já conta com alguns modelos de predição disponíveis.

PyCaret é uma biblioteca Python de machine learning open source que automatiza fluxos de aprendizado de máquina com o intuito de acelerar e aumentar a produtividade do ciclo de experimentos. A biblioteca conta com o módulo de série temporal, sendo possível treinar, testar e analisar diversos modelos.

O primeiro passo consiste em importar a biblioteca Pandas para manipular a série temporal escolhida. O resultado na listagem de minutos onde o index de cada tarefa é representado pela data.

[3] <https://pycaret.org/>

```
# Importar biblioteca
import pandas as pd

# Ler dados da série temporal
dt = pd.read_csv('CRUD.csv')

# Converter data no formato string para o formato datetime
dt['Data'] = pd.to_datetime(dt['Data'], infer_datetime_format=True)

# Definir a data como index da listagem
dt = dt.set_index('Data')
```

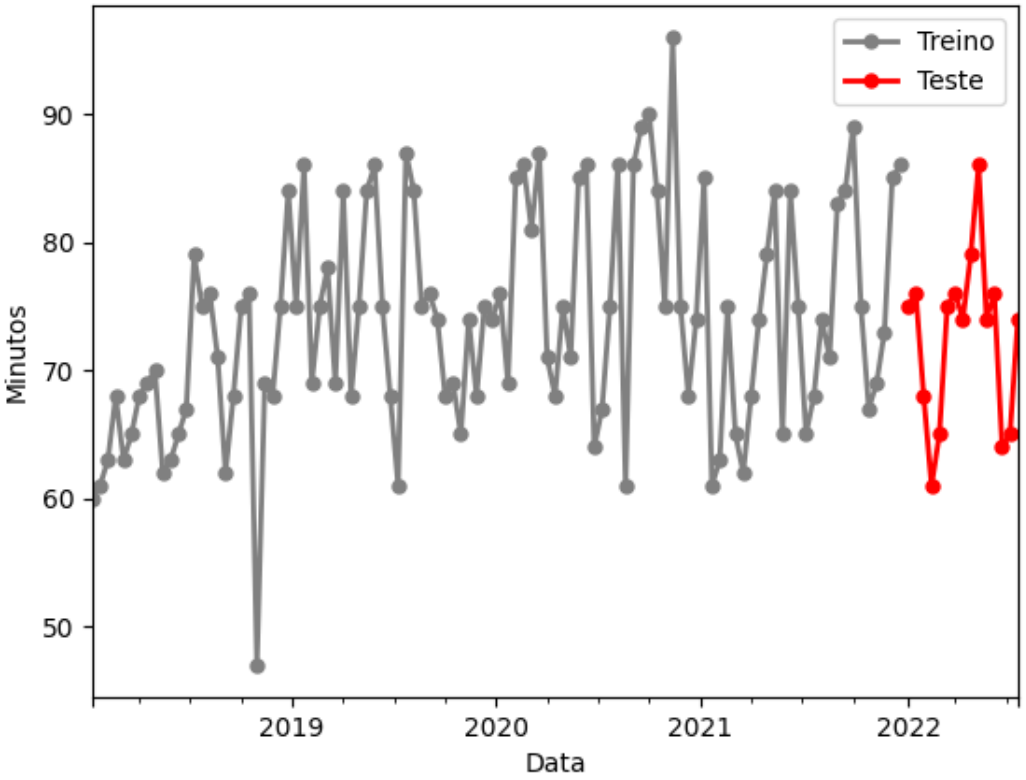
Em seguida, é necessário criar o ambiente de treinamento utilizando a função *setup,* utilizando 3 parâmetros. O primeiro parâmetro é a série temporal (dt) resultante do primeiro passo, seguido pelo horizonte de previsão (fh), correspondente à quantidade de dias escolhida para realização dos testes. Logo, o modelo vai predizer os 15 últimos valores, que serão comparados com os reais valores. Por fim, no período sazonal (seasonal period), como já citado anteriormente, os dados foram criados quinzenalmente, em inglês *biweekly,* assim sendo representado pela letra B. Por fim, para alcançar os resultados, foi utilizada a função *compare_models,* que treina e avalia o desempenho de todos os modelos disponíveis na biblioteca.

```
# Importar módulo de séries temporais da biblioteca pycaret
from pycaret.time_series import *

# Inicializar o ambiente de treinamento
setup(dt, fh=15, seasonal_period='B')

# Treinar e avaliar o desempenho de todos os modelos disponíveis
compare_models()
```

As **Figuras 15**, **16** e **17** representam as séries temporais Modal, Validadores e CRUD, respectivamente. As linhas em cinza representam o conjunto de dados que foram utilizados para os treinos, enquanto as linhas em vermelho representam o

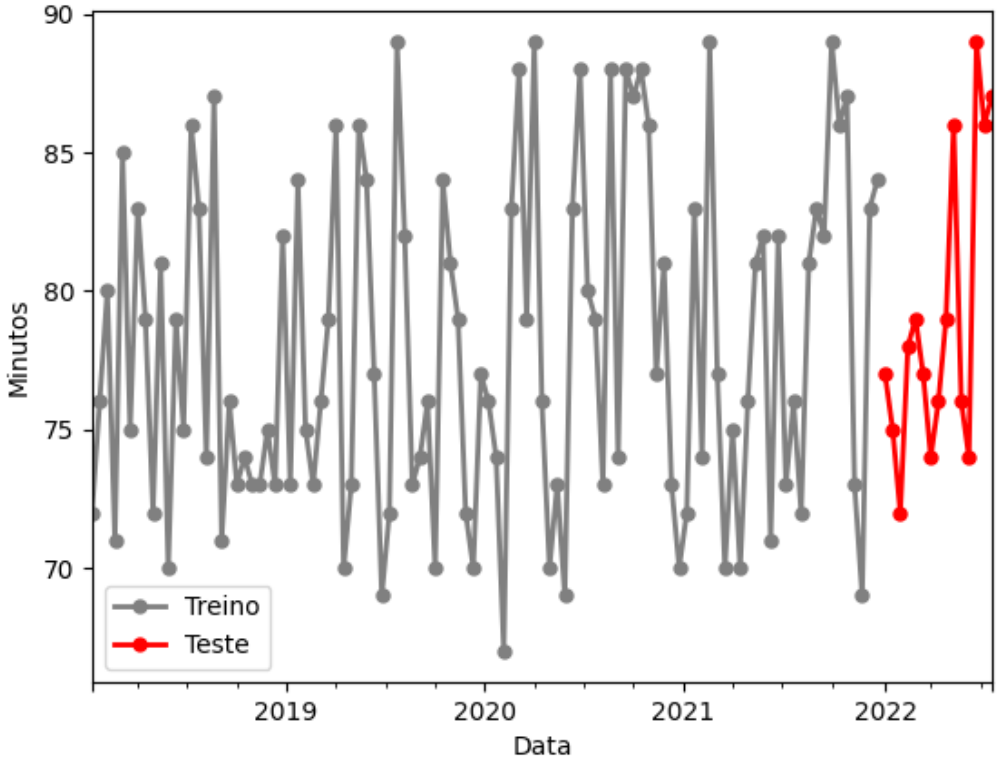
conjunto de dados utilizados para testes. As **Tabelas 3**, **4** e **5** mostram os valores das métricas para cada modelo utilizado nas 3 séries temporais.

Figura 15 - Série temporal Modal, destacando os dados de treino e teste



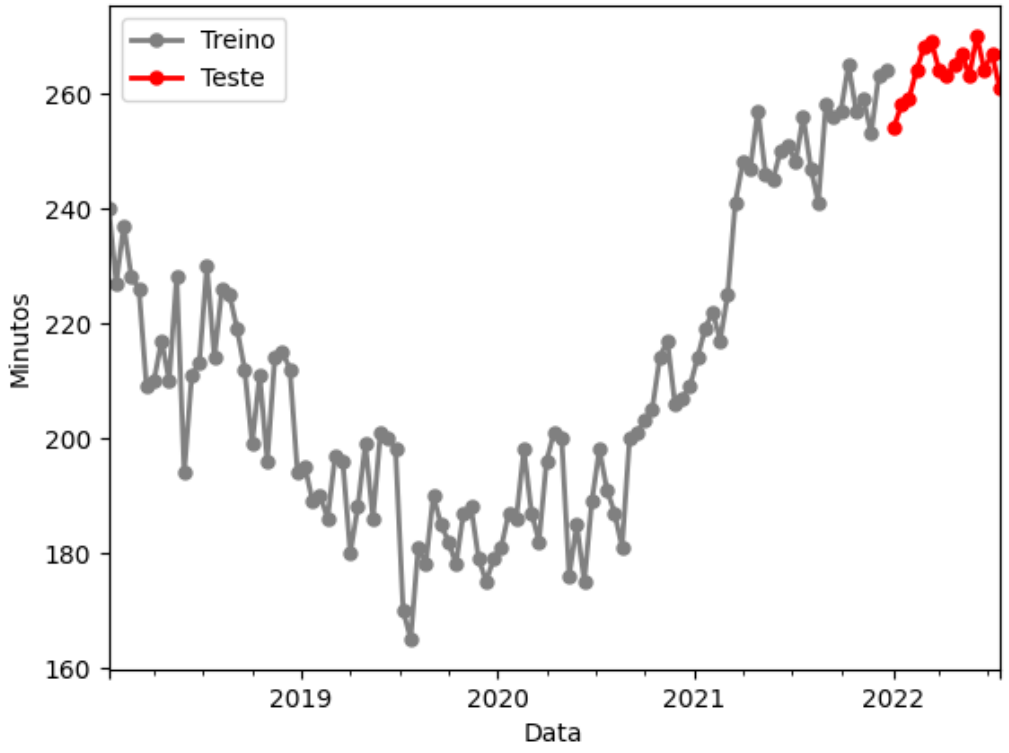
Fonte: Elaborada pelo autor.

Figura 16 - Série temporal Validadores, destacando os dados de treino e teste



Fonte: Elaborada pelo autor.

Figura 17 - Série temporal CRUD, destacando os dados de treino e teste



Fonte: Elaborada pelo autor.

Tabela 3 - Valores das métricas resultantes da predição da série temporal Modal, utilizando o modelo Random Forest

Modelo	MASE	MAE	RMSE	MAPE
Random Forest	2,778	9,8520	6,1358	0,217
Decision Tree	3,285	10,1254	6,9296	0,266
Extra Trees	4,665	10,9271	10,4868	0,351
Gradient Boosting	5,080	11,6600	11,2017	0,387
K Neighbors	5,527	11,6746	11,5837	0,420
Light Gradient Boosting	5,653	11,7441	13,1867	0,429
Huber	5,743	11,8520	12,5842	0,453
AdaBoost	6,168	12,5250	12,5971	0,468
Ridge	6,301	12,7835	137,784	0,484
Least Angular Regressor	6,301	12,7834	137,782	0,484
Linear	6,301	12,7834	137,782	0,484
Elastic Net	6,307	12,7923	138,143	0,484
Lasso	6,328	12,8263	138,745	0,485
Bayesian Ridge	6,380	12,9105	140,442	0,487
Naive Forecaster	7,119	13,2222	149,994	0,533
Theta Forecaster	7,558	13,8865	156,467	0,564
ETS	8,601	147,172	175,528	0,640
Exponential Smoothing	8,604	147,215	175,575	0,640
Auto ARIMA	8,762	150,606	183,391	0,645
Orthogonal Matching Pursuit	9,252	159,488	199,867	0,686
ARIMA	9,303	159,071	188,724	0,686
Croston	14,068	242,169	264,636	1,025
Seasonal Naive Forecaster	14,359	246,667	280,466	1,055
Grand Means Forecaster	15,433	268,833	298,148	1,153
Lasso Least Angular Regressor	15,515	269,707	300,899	1,155
Polynomial Trend Forecaster	15,951	277,414	307,453	1,190

Fonte: Elaborada pelo autor.

Tabela 4 - Valores das métricas resultantes da predição da série temporal Validadores, utilizando o modelo Random Forest

Modelo	MASE	MAE	RMSE	MAPE
Random Forest	8,445	5,3795	6,4137	0,702
Gradient Boosting	8,719	5,7749	6,4340	0,730
Extra Trees	8,719	5,7749	6,4340	0,730
Decision Tree	8,776	5,8125	6,4521	0,740
Polynomial Trend Forecaster	8,777	5,8134	6,4532	0,740
Exponential Smoothing	8,777	5,8134	6,4532	0,740
ETS	8,777	5,8134	6,4532	0,740
Theta Forecaster	8,779	5,8148	6,4543	0,740
ARIMA	8,780	5,8150	6,4517	0,736
Bayesian Ridge	8,788	5,8210	6,4574	0,741
Lasso	8,810	5,8356	6,4641	0,743
Grand Means Forecaster	8,817	5,8392	6,7219	0,741
Light Gradient Boosting	8,818	5,8412	7,2032	0,725
Elastic Net	8,820	5,8421	6,4675	0,744
Ridge	8,831	5,8494	6,4717	0,745
Least Angular Regressor	8,831	5,8494	6,4717	0,745
Orthogonal Matching Pursuit	8,831	5,8494	6,4717	0,745
Linear	8,831	5,8494	6,4717	0,745
Huber	8,855	5,8647	6,5003	0,743
K Neighbors	8,974	5,9432	6,6119	0,753
AdaBoost	9,099	6,0256	6,7725	0,753
Croston	9,253	6,1296	6,8935	0,788
Auto ARIMA	9,436	6,2550	7,9169	0,783
Lasso Least Angular Regressor	9,812	6,4989	8,6020	0,806
Naive Forecaster	12,013	7,9556	9,5993	1,070
Seasonal Naive Forecaster	13,321	8,4555	10,8745	1,154

Fonte: Elaborada pelo autor.

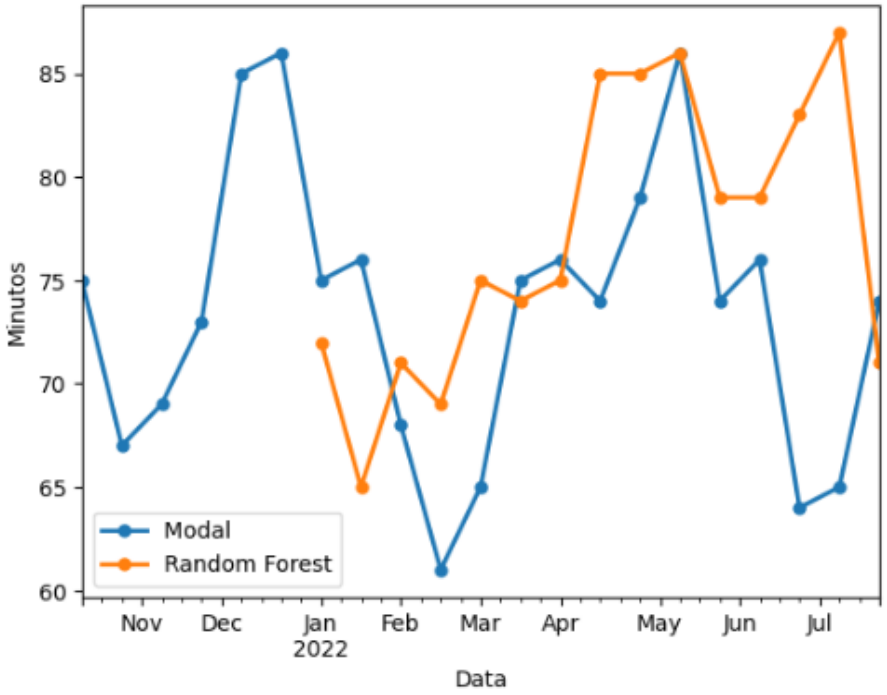
Tabela 5 - Valores das métricas resultantes da predição da série temporal CRUD, utilizando o modelo Random Forest

Modelo	MASE	MAE	RMSE	MAPE
Random Forest	2,403	9,8191	5,9134	0,222
K Neighbors	4,987	10,1975	7,5237	0,282
Extra Trees	5,134	10,2553	9,1460	0,317
Decision Tree	5,234	14,3226	10,2435	0,347
Gradient Boosting	5,281	15,5206	12,7983	0,372
Huber	6,365	16,2763	12,8776	0,411
AdaBoost	6,372	17,3121	14,6664	0,429
Light Gradient Boosting	7,199	17,5921	18,4783	0,464
Least Angular Regressor	8,251	17,6683	18,3553	0,517
Linear	9,610	18,8250	29,6770	0,572
Ridge	9,640	30,6087	38,1699	0,738
Elastic Net	9,712	38,8518	102,590	0,753
Bayesian Ridge	9,755	57,5398	108,626	0,757
Lasso	10,134	58,9839	108,717	0,798
Naive Forecaster	13,182	59,3397	119,101	0,838
ETS	13,481	62,8978	120,918	0,890
Theta Forecaster	14,735	78,8917	138,189	0,943
Auto ARIMA	16,682	127,970	189,102	0,959
Exponential Smoothing	16,821	128,687	192,666	0,997
Orthogonal Matching Pursuit	16,902	157,558	193,783	0,999
ARIMA	17,953	158,423	203,812	1,097
Croston	18,887	204,608	286,236	1,120
Seasonal Naive Forecaster	18,893	218,609	295,247	1,149
Lasso Least Angular Regressor	18,942	223,365	299,309	1,267
Grand Means Forecaster	18,954	233,374	300,666	1,289
Polynomial Trend Forecaster	18,991	233,428	301,667	1,398

Fonte: Elaborada pelo autor.

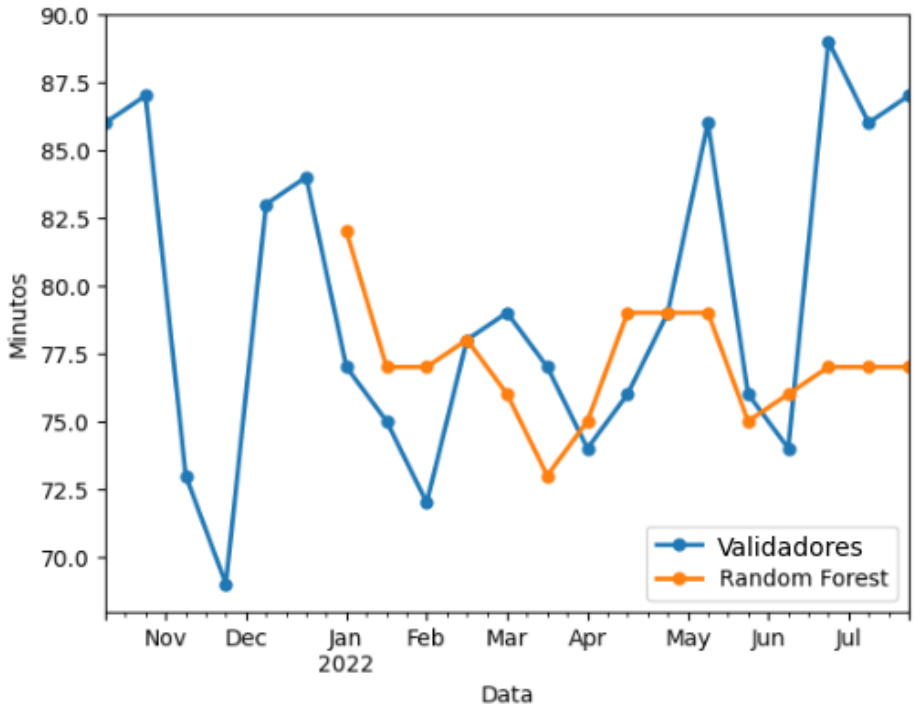
O modelo Random Forest obteve os melhores resultados nas 3 séries temporais junto a outros modelos que também utilizam *decision trees*. Vale ressaltar o modelo K Neighbors (Knn), que também obteve bons resultados. Nas **Figuras 18**, **19** e **20**, os dados preditos pelo modelo Random Forest são representados pela cor laranja, junto aos dados reais em azul, é possível notar que, apesar de poucos dados para treino e teste, em vários pontos o modelo conseguiu seguir chegar a resultados precisos ou um comportamento muito parecido mesmo não acertando nenhum valor, que pode ser observado na **Figura 20** entre os meses de Fevereiro e Abril de 2022.

Figura 18 - Comparação entre os valores previstos pelo modelo Random Forest e os valores reais da série temporal Modal



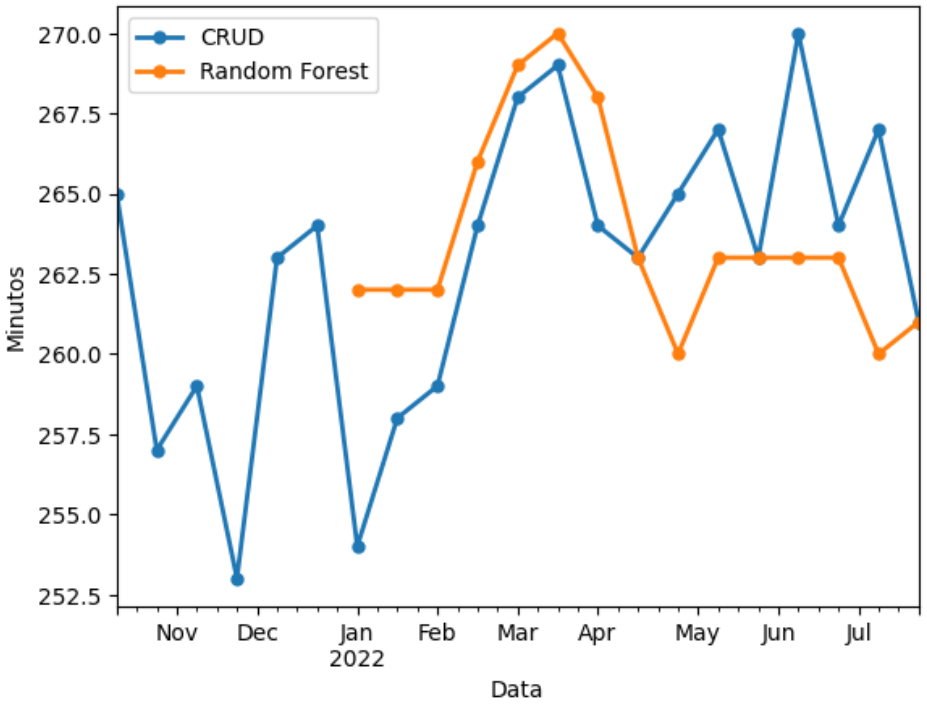
Fonte: Elaborada pelo autor.

Figura 19 - Comparação entre os valores previstos pelo modelo Random Forest e os valores reais da série temporal Validadores



Fonte: Elaborada pelo autor.

Figura 20 - Comparação entre os valores previstos pelo modelo Random Forest e os valores reais da série temporal CRUD



Fonte: Elaborada pelo autor.

Ao analisar as métricas, a ST Validadores foi a que obteve melhor resultado com o modelo, alcançando 5,3 na métrica MAE. A métrica é a média da diferença absoluta entre o realizado e o previsto, e, com a visualização da **Tabela 4**, é possível interpretar que o modelo tem uma média de erro de aproximadamente 5 minutos.

As STs Modal e CRUD, ambas tiveram 9,8 na métrica MAE, mas abertas a diferentes interpretações. Se utilizarmos o mesmo raciocínio da ST Validadores e interpretar que o modelo tem uma média de erro de aproximadamente 10 minutos, o modelo teve um desempenho um pouco melhor na ST CRUD. Se a análise for feita a partir dos dados das duas STs, a ST CRUD está na faixa dos 260 minutos, e a ST Modal está na faixa dos 80 minutos. Caso o modelo tenha esses 10 minutos de erro, é mais impactante em tarefas que levam menos tempo para serem realizadas, refletindo de maneira indireta em outras métricas em que a ST CRUD obteve melhores resultados.

5.3 RESULTADOS: RANDOM FOREST

Após a análise dos testes, o modelo Random Forest será criado e treinado pelas funções *create_model* e *finalize_model*, e os próximos valores da série temporal serão previstos pela função *predict_model*.

```
# Treina e avalia o desempenho do modelo Random Forest
model = create_model('rf_cds_dt')

# Treina o modelo em todo o conjunto de dados
final = finalize_model(model)

# Prevê próximos valores utilizando o modelo treinado
predict = predict_model(final, fh = 1)
```

Durante o desenvolvimento deste trabalho 8 *sprints* se passaram, e seus valores não foram utilizados nos treinamentos e testes do modelo, esse valores foram comparados com os valores preditos pela função *predict_model*, assim obtendo os valores das métricas na **Tabela 6**. Para cada execução da função, a série temporal é atualizada com os novos valores, nesse caso, ao prever os valores

da sprint de 19/08/2022, o valor real da sprint de 05/08/2022 foi adicionada na série temporal. As 8 execuções da função levaram em média 0,1 segundo para retornar o resultado.

Tabela 6 - Valores da métrica MAE para cada valor previsto utilizando o modelo Random Forest em cada série temporal.

Data	Modal (MAE)	Validadores (MAE)	CRUD (MAE)
05/08/2022	12,6428	5,7954	7,4853
19/08/2022	16,3738	5,1948	7,3456
02/09/2022	8,4563	7,5438	8,1352
16/09/2022	10,456	5,1462	15,1346
30/09/2022	8,6389	6,1865	11,5163
14/10/2022	11,4868	5,2345	9,4234
28/10/2022	9,3565	5,8765	10,3684
11/11/2022	15,7867	6,3967	9,3526

Fonte: Elaborada pelo autor.

Analisando a **Tabela 6**, a série temporal de Validadores continuou tendo o melhor resultado com o Random Forest, atingindo 5,9 na métrica MAE, um pouco a mais que nos testes. CRUD continuou com 9,8 e Modal, que teve o pior desempenho entre as 3 séries, obteve 11,65 na métrica MAE.

Por fim, é possível concluir que, apesar da quantidade de dados e da falta de padrão deles, foi possível obter bons resultados, mesmo não tendo uma precisão de acerto, é possível utilizar esse modelo em projetos ágeis.

Todas as etapas do desenvolvimento podem ser encontradas no *link* https://github.com/BatistaYuri/tcc_time_series_prediction, onde estão todos os códigos utilizados. No entanto, os dados utilizados não foram disponibilizados para divulgação.

6. CONCLUSÃO

O presente trabalho apresenta o desenvolvimento de um modelo para predizer o tempo estimado de execução de tarefas de projetos ágeis, utilizando dados da empresa JExperts. Durante o desenvolvimento do TCC, foi analisada a fundamentação teórica em relação ao conceito de séries temporais e *machine learning* aplicado às séries temporais, levantando trabalhos relacionados à predição de séries temporais para contribuir com os estudos.

O modelo tem o objetivo de avaliar a viabilidade de utilizar predição de séries temporais para determinar o tempo gasto para executar tarefas em projetos ágeis. A partir dos dados disponibilizados pela empresa JExperts, foram criadas séries temporais, as quais foram utilizadas em modelos preditivos da biblioteca PyCaret, dentro do Jupyter Notebook, e seus resultados, avaliados.

Para aplicá-la na prática, a primeira dificuldade foi encontrar uma base de dados. Não há vantagens para as empresas deixarem dados dessa natureza públicos, portanto, os dados testados permanecem privados. O segundo obstáculo, o qual demandou mais tempo de desenvolvimento, foi a montagem das séries temporais. Entretanto, devido à impossibilidade de montar uma consulta SQL que resultasse em uma série temporal pronta para os testes (devido aos tipos de informações), houve a necessidade de separar os dados manualmente. Após a separação prévia por SQL, o resultado foi a transformação do grande volume de 10 mil dados, em três séries temporais de aproximadamente 110 dados cada.

Felizmente, o número baixo de dados em cada ST não impediu que os testes obtivessem bons resultados. A desvantagem de possuir poucos dados mostrou que o modelo *Random Forest*, além de retornar resultados rapidamente, soube trabalhar com menor volume de dados, chegando em uma média de erro de 5 minutos em uma das STs, não sendo necessário ter uma grande quantidade, mas sim dados consistentes que sigam uma periodicidade (nesse caso, de 2 semanas). Para tanto, a equipe de desenvolvimento precisa estar comprometida em especificar e atualizar as tarefas, de forma que seja possível contribuir com a evolução dos planejamentos de sprint em equipes de desenvolvimento.

Para trabalhos futuros, pode-se citar a criação de um modelo preditivo específico para séries temporais de projetos ágeis, a criação de uma ferramenta que

monte séries temporais com dados de projetos ágeis, capaz de realizar a análise estatística da série antes de aplicação dos métodos e a automação do processo como um todo.

7. REFERENCIAL TEÓRICO

BATISTA, G. E. A. P. A; KEOGH, E.; TATAW, O. M.; SOUZA, V. M. A. CID: **An efficient complexity-invariant distance for time series**. Data Mining and Knowledge Discovery, Springer, United States of America, vol. 28, 2014.

BECK, K. **Programação Extrema Explicada**. Bookman, 1999.

BONTEMPI, G.; TAIEB, S. B.; BORGNE YA. L. **Machine Learning Strategies for Time Series Forecasting**. eBISS 2012, Business Intelligence, 2013.

BREIMAN, Leo.**Random Forests.** Machine Learning, vol. 45, 2001.

BROCKWELL, P. J. D. R. A. **In: Introduction to time series forecasting**. vol 2, 2001.

CHATFIELD, C. **The analysis of time series: An introduction**. Boca Raton, United States of America: Taylor & Francis, 2013.

CHAVES et al. **Leishmaniose visceral no Piauí, 2007-2019: análise ecológica de séries temporais e distribuição espacial de indicadores epidemiológicos e operacionais**. Revista Brasileira de Epidemiologia, 2022

COWPERTWAIT, P. S. P.; METCALFE, A. V. **Introductory Time Series with R**. Springer Dordrecht Heidelberg London New York, 2009.

DIGITAL.IA. **15th State of the Agile Report,** 2021. Disponível em: <https://itnove.com/wp-content/uploads/2021/07/15th-state-of-agile-report.pdf>. Acesso em: Maio de 2022.

EHLERS, R. S. **Análise de séries temporais**, 5ª edição. Departamento de Estatística, Universidade Federal do Paraná, 2009.

EVANS, J. R. **Business Analytics: The Next Frontier for Decision Sciences**, 2012. Disponível em: <http://faculty.cbpp.uaa.alaska.edu/afef/business_analytics.htm> Acesso em: Maio 2022.

FORECAST, AMAZON. **Guia do Desenvolvedor,** 2022. Disponível em: <https://docs.aws.amazon.com/pt_br/forecast/latest/dg/metrics.html#metrics-RMSE> . Acesso em: Novembro de 2022.

GONÇALVES, L. **Scrum: The methodology to become more agile**. Universidade de Coimbra, 2018.

GUYON, I.; ELISSEEFF, A. **An introduction to variable and feature selection**. Journal of Machine Learning Research, vol. 3, 2003.

KIRCHGASSNER, G.; WOLTERS, J. **Introduction to Modern Time Series Analysis**. Springer-Verlag Berlin Heidelberg New York, 2007.

KOECH, D. KIMUTAI. **A Complete Guide on How to Impute Missing Values in Time Series in Python**. Section, 2022. Disponível em: <https://www.section.io/engineering-education/missing-values-in-time-series/> Acesso em: Novembro 2022.

LATORRE, M. R. D. O. ;CARDOSO, M. R. A. **Análise de séries temporais em epidemiologia: uma introdução sobre os aspectos metodológicos.** Revista Brasileira de Epidemiologia, 2001.

LI et al. **Random forest regression for online capacity estimation of lithium-ion batteries**. Applied Energy, vol. 232, 2018.

LUCENA, A. H. B. F. **Uma Abordagem Híbrida para a Predição de Séries Temporais**. Universidade Federal de Pernambuco, Centro de Informática, Bacharelado em Sistemas de Informação, 2016.

MORETTIN, P. A.; TOLOI, C. M. C. **Análise de séries temporais**. vol 2. ed. São Paulo, Brasil: Blucher, 2006.

PARMEZAN, A. R. S. **Predição de séries temporais por similaridade**. Instituto de Ciências Matemáticas e de Computação - ICMC-USP, 2016.

SÉRGIO, A. T. **Seleção Dinâmica de Combinadores de Previsão de Séries Temporais**. Programa de Pós-Graduação do Centro de Informática, Universidade Federal de Pernambuco, 2017.

SCHWABER, K.; BEEDLE, M. **Agile Software Development with Scrum.** Prentice Hall PTRUpper Saddle River, NJ, United States, 2001.

SCHWABER, K. **SCRUM Development Process**, 1995.

SEZER. O. B.; GUDELEK, M. U.; OZBAYOGLU, A. M. **Financial time series forecasting with deep learning : A systematic literature review: 2005–2019.** Applied Soft Computing, vol. 90, 2020.

SHALEV-SHWARTZ, S.; BEN-DAVID S. **Understanding Machine Learning: From Theory to Algorithms**. Cambridge University Press, 2014.

SOARES, M. S. **Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software**. Universidade Presidente Antônio Carlos, 2004

SOARES, M. S. **Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software**. Universidade Presidente Antônio Carlos, 2004.

TORREÃO, Paula G. B. C, **Gerenciamento de Projetos**, 2006. Disponível em: <https://www.cin.ufpe.br/~if717/leituras/artigo-gerenciamento-de-projetos-paula-coelho.pdf> Acesso em: Julho 2022.

VELOSO, L. T. **Modelos de Séries Temporais e Gráficos de Controle Estatístico aplicados a Indicadores de Vigilância Epidemiológica do Ministério da Saúde**.

Universidade de Brasília - UnB, Instituto de Ciências Exatas - IE, Departamento de Estatística - EST, 2018.

VILLANUEVA, W. J. P. **Comitê de Máquinas em Predição de Séries Temporais**. Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 2006.

ZHANG, G. PETER. **Time series Forecasting using a hybrid ARIMA and neural network model**. Elsevier, Neurocomputing 50, 2003.

Predição de séries temporais em projetos ágeis

Yuri Batista

Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

batista.yuri@grad.ufsc.br

Resumo: A metodologia ágil está presente em grande parte das empresas - principalmente nos segmentos de tecnologia -, e vem contribuindo para alcançarem o sucesso com a velocidade acelerada que o mercado exige cada vez mais nas últimas décadas. Uma dos principais framework utilizados é o Scrum, em que os objetivos são divididos em tarefas menores e criadas sprints periódicas, para finalizá-las em um curto período de tempo. Para auxiliar, são utilizadas ferramentas para organização das sprints, em que são guardadas no histórico e, posteriormente, os dados podem ser transformados em uma série temporal. Neste trabalho, após inúmeros testes, foi desenvolvido um modelo para predizer o tempo necessário para concluir cada tarefa das sprints em empresas de tecnologia, a partir de métodos preditivos que utilizam Machine Learning, com o objetivo de apoiar negócios a serem mais assertivos em seus planejamentos. Neste sentido, a empresa JExperts apoiou com o desenvolvimento do trabalho a partir dos dados de seus projetos de sua própria plataforma, que possui um módulo Agile, o Channel, para a criação de séries temporais em que foram utilizadas, testadas e avaliadas em diversos métodos de predição de dados. O método Random Forest obteve os melhores resultados nas três séries temporais utilizadas, onde o tempo predito para finalizar as tarefas teve uma média de erro de aproximadamente cinco minutos, em tarefas que em média levaram 85 a 90 minutos para serem finalizadas, e os piores resultados alcançados por esse mesmo modelo, foi um erro médio de aproximadamente 11 minutos em tarefas que em média foram finalizados entre 70 e 80 minutos. Então conclui-se que é possível utilizar essa técnica em projetos ágeis.

1.Introdução

A 4ª Revolução Industrial aumentou a velocidade das mudanças com que as empresas lutam para conseguir acompanhar o ritmo de crescimento. Mas, o que vimos na última década, é um novo padrão muito mais rápido, em que as organizações buscam formas de

acompanhar o mesmo movimento. Um dos esforços para alcançar o objetivo é a implementação de metodologias ágeis.

Métodos ágeis transformaram os projetos de tecnologia e aumentaram as taxas de sucesso dos times de Tecnologia da informação (TI) nas últimas duas décadas. Hoje, as metodologias ágeis estão sendo aplicadas em todas as áreas da empresa, otimizando o trabalho de desenvolvimento, seja em projetos de softwares ou em qualquer outro processo. Segundo o relatório 15th *State of the Agile Report*, 94% dos entrevistados reportaram que a empresa que trabalham utiliza a metodologia ágil.

Existem diversos métodos para utilizar a gestão ágil e determinadas metodologias podem ser mais adequadas conforme a necessidade do projeto. No entanto, segundo o relatório 15th *State of the Agile Report*, 66% dos entrevistados responderam que utilizam Scrum (DIGITAL, 2021). O Scrum é uma metodologia que procura ser eficaz, produtivo e controlar melhor os riscos do processo, consiste em uma série de *sprints*, que possuem tarefas a serem concluídas até o final da duração de um mês ou menos (GONÇALVES, 2018).

Para facilitar a gestão, as empresas utilizam ferramentas ágeis de gerenciamento de projetos, em que as *sprints* e tarefas são organizadas e mantêm histórico à disposição. Assim, após algum tempo utilizando a metodologia ágil como recurso e armazenando os dados gerados, é formada uma série temporal. Uma série temporal, também denominada série histórica, é uma coleção de dados em intervalos regulares feitas sequencialmente ao longo do tempo (LATORRE e CARDOSO, 2001).

Assim, com a análise de dados e métodos preditivos, é possível gerar informações concretas para apoiar a tomada de decisão. Uma das decisões mais favorecidas pelo histórico de dados e mais usada indiretamente, é a estimativa de tempo de conclusão para cada tarefa, em que o time não olha diretamente para os dados, mas tomam decisões com base na experiência de executar uma tarefa parecida no passado, limitando-se apenas pela memória dos integrantes.

Dessa forma, foi possível observar a aplicação da metodologia ágil como ferramenta para contribuir com as análises de dados, que possui duas vertentes: descritiva e preditiva. A análise descritiva busca compreender os acontecimentos a partir da conversão de dados úteis que passaram por processos de categorização, caracterização, consolidação e classificação (EVANS, 2012). Em contrapartida, a análise preditiva examina as relações entre os conjuntos de dados históricos a serem explorados, de forma a compreender e buscar prever valores futuros (EVANS, 2012).

De acordo com Parmezan (2016), os métodos para predição de séries temporais são baseados essencialmente na ideia de que dados históricos contemplam padrões intrínsecos, geralmente de difícil identificação e nem sempre interpretáveis que, se descobertos, podem auxiliar na descrição futura do fenômeno investigado. Neste sentido, as séries temporais buscam responder em quais circunstâncias, determinados padrões se repetem, bem como as mudanças que podem acontecer ao longo do tempo.

Diante da dificuldade de estimar tempo para as atividades, o presente trabalho tem como objetivo desenvolver um modelo para predizer o tempo estimado de execução de tarefas de projetos da empresa JExperts.

2. Desenvolvimento

Para cumprir o objetivo deste trabalho, o desenvolvimento foi dividido em 3 etapas:

Primeira etapa: Preparação das séries temporais.

Segunda etapa: Testes e análise dos resultados de todos modelos preditivos disponibilizados pelo PyCaret.

Terceira etapa: Análise dos resultados do modelo preditivo com melhor desempenho com novos dados.

2.1 Preparação de Séries Temporais

A empresa JExperts[4], atuante no mercado de desenvolvimento de software no Brasil há 20 anos, disponibilizou um conjunto de 10 mil dados, do período de 2018 a 2022 para experimentos, coletados do módulo Agile da plataforma para escritórios corporativos de gestão de portfólios, programas e projetos, Channel. Os dados pertencem a projetos internos da empresa, referentes à manutenção e migração das tecnologias usadas na plataforma. Os dados são correspondentes às tarefas da equipe de desenvolvimento e descritas a partir de atributos. Para este estudo, foram utilizados os atributos de “nome”, “descrição”, “tag” (palavra chave para facilitar na busca de tarefas do mesmo conjunto, por exemplo: BACK-END, FRONT-END), “data de criação” e “tempo” (tempo utilizado para concluir a tarefa) da tarefa. As informações sobre “nome”, "descrição" e “tag” foram usadas para agrupar os dados, enquanto dados de “data de criação” e “tempo” foram utilizados para realizar a predição.

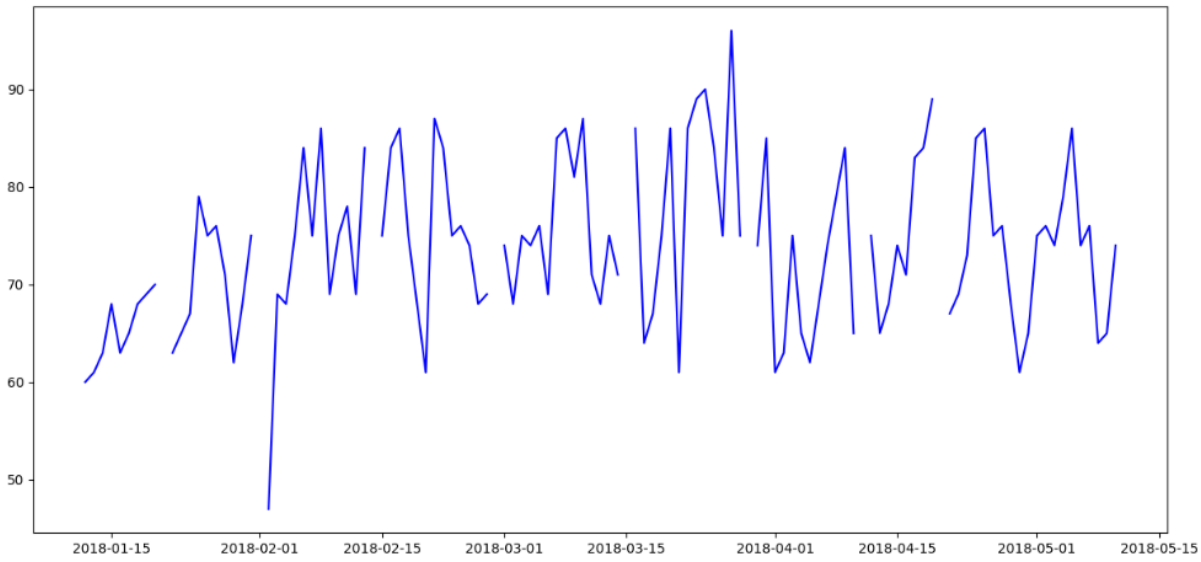
[4] <https://www.jexperts.com.br>

Foram extraídos *datasets* deste conjunto de dados e analisados cada tarefa para, assim, desenvolver uma série temporal de dados com características iguais ou semelhantes, conforme demonstrado na **Figura 1**. No exemplo, a série temporal do ouro é sempre o seu preço, portanto, não haveria coerência na construção de uma série temporal com todos os 10 mil dados disponibilizados, em que a maioria não possui correlação entre si.

A primeira etapa do processo consistiu em separar os dados pelas tags “BACK-END” e “FRONT-END”. Em seguida, os dados foram agrupados por semelhança entre nomes e descrições, utilizando postgreSQL e análise manual para reorganização de dados, a partir do detalhamento dos atributos de “nome” e “descrição”. Após a formação dos *datasets*, apenas três tiveram um conjunto considerável, com mais de 100 dados cada, correspondentes às tarefas de desenvolvimento de “CRUD”, “Validadores” e “Modal”. CRUD são as operações básicas de criar, consultar, atualizar e excluir dados em bases de dados relacionais, Validadores é uma camada lógica para validar os dados enviados para a execução das operações do CRUD e Modal é uma janela que abre sobre o conteúdo da página.

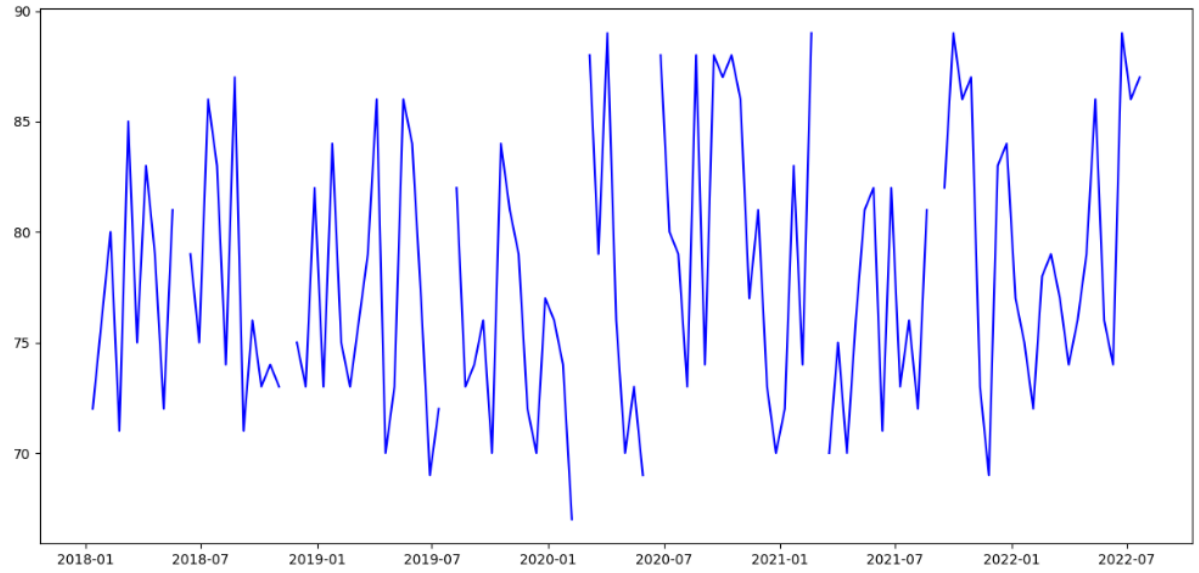
Em um contexto de trabalho utilizando metodologia ágil e organizando as tarefas a partir de *sprints* com duração de duas semanas, a periodicidade das séries temporais foram correspondentes. As tarefas são criadas no último dia da sprint anterior, geralmente às sextas-feiras. Entretanto, nem todas as tarefas serão criadas no mesmo dia da semana. Por isso, as tarefas criadas durante as *sprints* terão sua data de criação alterada para comportar a periodicidade da ST, conforme demonstrado nas **Figuras 1**, **2** e **3**.

Figura 1 - Série temporal Modal incompleta



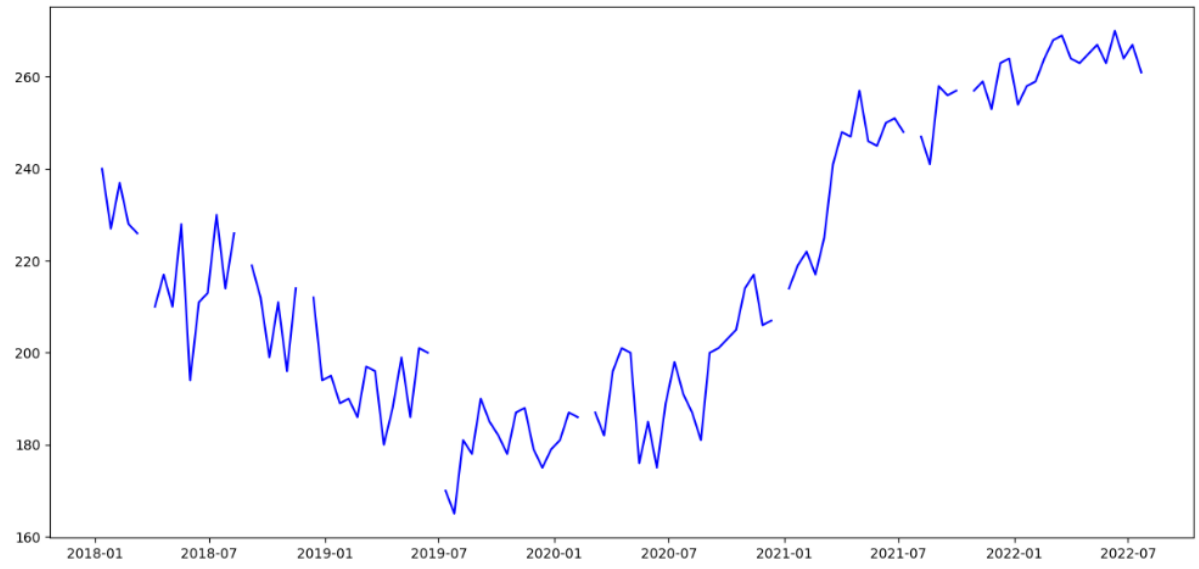
Fonte: Elaborada pelo autor.

Figura 2 - Série temporal Validadores incompleta



Fonte: Elaborada pelo autor.

Figura 3 - Série temporal CRUD incompleta



Fonte: Elaborada pelo autor.

2.1.1 Dados faltantes

Como observado nas **Figuras 1**, **2** e **3**, as séries temporais estão incompletas, uma vez que nem toda sprint possui uma tarefa que seja compatível com o período das séries apresentadas, dificultando o processo de construção e análise das ST. Para sanar tal obstáculo,

foram testados 6 métodos utilizados para completar os valores faltantes: *Mean imputation, Median imputation, Last Observation Carried Forward, Next Observation Carried Backward,* Linear *interpolation e Spline interpolation*.

Tabela 1 - Valores sequentes da série temporal incompleta Modal

Data	**Minutos**
05/10/2018	75
19/10/2018	
02/11/2018	47

Fonte: Elaborada pelo autor.

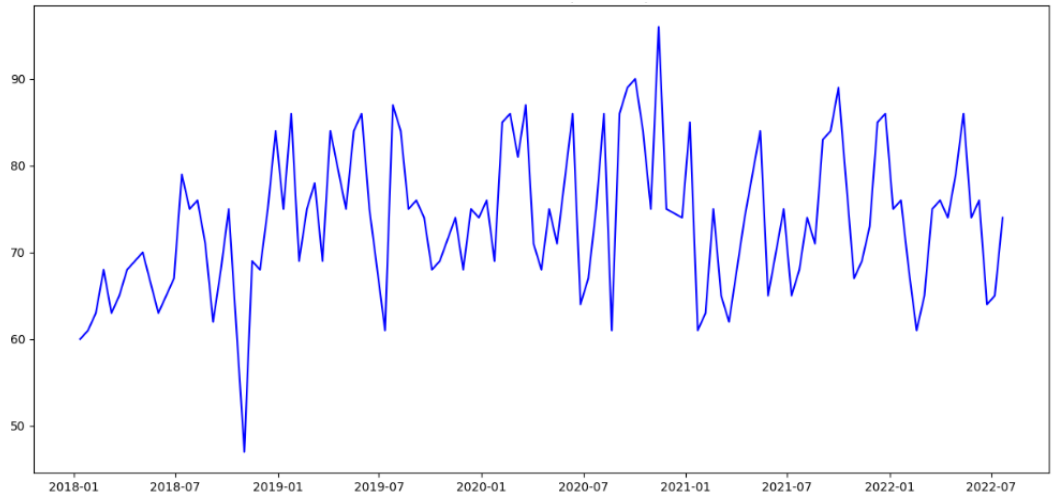
Tabela 2 - Valores obtidos pelos métodos para completar valor faltante da **Tabela 1**

Métodos	**Minutos**
Mean imputation	73.6
Median imputation	74
Last Observation Carried Forward	75
Next Observation Carried Backward	45
Linear interpolation	61
Spline interpolation	61

Fonte: Elaborada pelo autor.

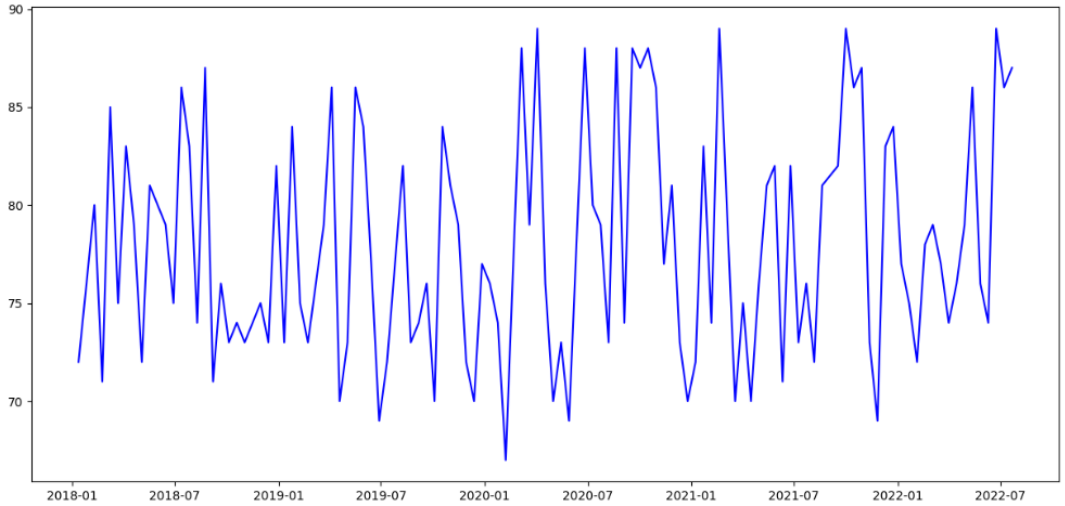
A **Tabela 1** apresenta 3 valores da série temporal Modal para ilustrar os resultados obtidos na **Tabela 2**. Os métodos Linear *interpolation* e *Spline interpolation* tiveram resultados idênticos e positivos, destacando-se nas três séries temporais. Por fim, o método *Spline interpolation* foi escolhido para completar os dados faltantes, uma vez que a interpolação linear e *spline* tende a fornecer valores de imputação, assim gerando as séries temporais completas ilustradas nas **Figuras 4**, **5** e **6**. Os dados imputados possuem o Erro Quadrado Médio, sendo consideradas as melhores técnicas nesse nível (KOECH, 2022).

Figura 4 - Série temporal Modal



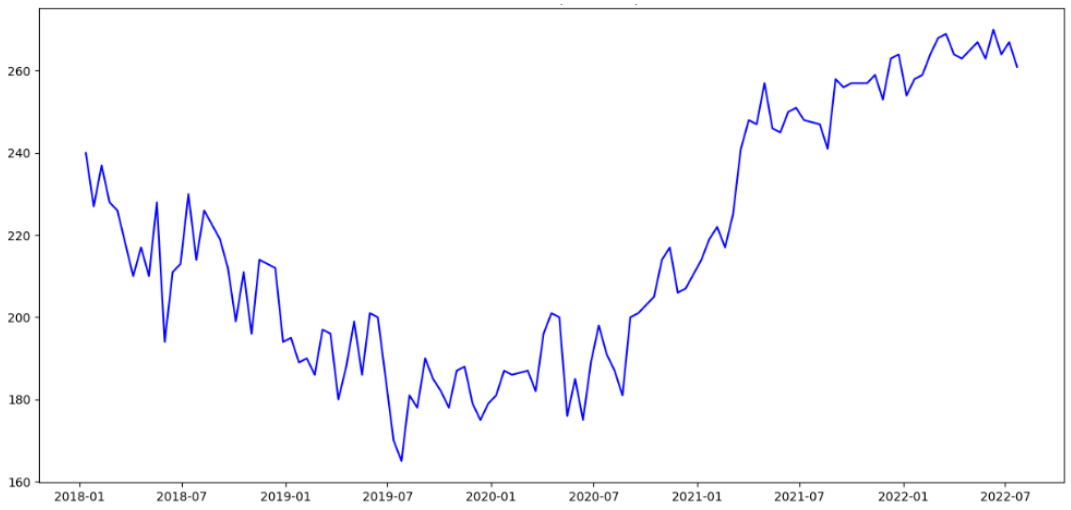
Fonte: Elaborada pelo autor.

Figura 5 - Série temporal Validadores



Fonte: Elaborada pelo autor.

Figura 6 - Série temporal CRUD



Fonte: Elaborada pelo autor.

2.3 TESTES

Após a montagem de três séries temporais, foi escolhido o modelo ideal, que melhor se adequa às características das séries atuais e futuras. A primeira característica corresponde ao tamanho das ST, uma vez que o modelo escolhido precisa apresentar resultados positivos em séries com poucos dados, já que o volume utilizado neste estudo são cerca de 110 dados. Além disso, o modelo também sabe lidar com a redução de erros, por serem STs pequenas, os outliers podem atrapalhar com facilidade e comprometer os resultados.

O ambiente utilizado para desenvolvimento foi o Jupyter notebook, utilizando a linguagem Python. Para os testes e predição dos dados, foi utilizado o módulo de Séries Temporais da biblioteca PyCaret[5], que já conta com alguns modelos de predição disponíveis.

O primeiro passo consistiu na importação da biblioteca Pandas para manipular a série temporal escolhida. O resultado na listagem de minutos onde o index de cada tarefa é representado pela data.

[5] <https://pycaret.org/>

```
# Importar biblioteca
import pandas as pd

# Ler dados da série temporal
dt = pd.read_csv('CRUD.csv')

# Converter data no formato string para o formato datetime
dt['Data'] = pd.to_datetime(dt['Data'], infer_datetime_format=True)

# Definir a data como index da listagem
dt = dt.set_index('Data')
```

Em seguida, o ambiente de treinamento foi criado utilizando a função *setup,* usando 3 parâmetros. O primeiro parâmetro é a série temporal (dt) resultante do primeiro passo, seguido pelo horizonte de previsão (fh), correspondente à quantidade de dias escolhida para realização dos testes. Logo, o modelo prediz os 15 últimos valores, comparados com os reais valores. Por fim, no período sazonal (seasonal period), como já citado anteriormente, os dados foram criados quinzenalmente, em inglês *biweekly,* assim sendo representado pela letra B. Por fim, para alcançar os resultados, foi utilizada a função *compare_models,* que treinou e avaliou o desempenho de todos os modelos disponíveis na biblioteca.

```
# Importar módulo de séries temporais da biblioteca pycaret
from pycaret.time_series import *

# Inicializar o ambiente de treinamento
setup(dt, fh=15, seasonal_period='B')

# Treinar e avaliar o desempenho de todos os modelos disponíveis
compare_models()
```

As **Tabelas 3**, **4** e **5** mostram os valores das métricas para cada modelo utilizado nas 3 séries temporais.

Tabela 3 - Valores das métricas resultantes da predição da série temporal Modal, utilizando o modelo Random Forest

Modelo	MASE	MAE	RMSE	MAPE
Random Forest	2,778	9,8520	6,1358	0,217
Decision Tree	3,285	10,1254	6,9296	0,266
Extra Trees	4,665	10,9271	10,4868	0,351
Gradient Boosting	5,080	11,6600	11,2017	0,387
K Neighbors	5,527	11,6746	11,5837	0,420
Light Gradient Boosting	5,653	11,7441	13,1867	0,429
Huber	5,743	11,8520	12,5842	0,453
AdaBoost	6,168	12,5250	12,5971	0,468
Ridge	6,301	12,7835	137,784	0,484
Least Angular Regressor	6,301	12,7834	137,782	0,484
Linear	6,301	12,7834	137,782	0,484
Elastic Net	6,307	12,7923	138,143	0,484
Lasso	6,328	12,8263	138,745	0,485
Bayesian Ridge	6,380	12,9105	140,442	0,487
Naive Forecaster	7,119	13,2222	149,994	0,533
Theta Forecaster	7,558	13,8865	156,467	0,564
ETS	8,601	147,172	175,528	0,640
Exponential Smoothing	8,604	147,215	175,575	0,640
Auto ARIMA	8,762	150,606	183,391	0,645
Orthogonal Matching Pursuit	9,252	159,488	199,867	0,686
ARIMA	9,303	159,071	188,724	0,686
Croston	14,068	242,169	264,636	1,025
Seasonal Naive Forecaster	14,359	246,667	280,466	1,055
Grand Means Forecaster	15,433	268,833	298,148	1,153
Lasso Least Angular Regressor	15,515	269,707	300,899	1,155
Polynomial Trend Forecaster	15,951	277,414	307,453	1,190

Fonte: Elaborada pelo autor.

Tabela 4 - Valores das métricas resultantes da predição da série temporal Validadores, utilizando o modelo Random Forest

Modelo	MASE	MAE	RMSE	MAPE
Random Forest	8,445	5,3795	6,4137	0,702
Gradient Boosting	8,719	5,7749	6,4340	0,730
Extra Trees	8,719	5,7749	6,4340	0,730
Decision Tree	8,776	5,8125	6,4521	0,740
Polynomial Trend Forecaster	8,777	5,8134	6,4532	0,740
Exponential Smoothing	8,777	5,8134	6,4532	0,740
ETS	8,777	5,8134	6,4532	0,740
Theta Forecaster	8,779	5,8148	6,4543	0,740
ARIMA	8,780	5,8150	6,4517	0,736
Bayesian Ridge	8,788	5,8210	6,4574	0,741
Lasso	8,810	5,8356	6,4641	0,743
Grand Means Forecaster	8,817	5,8392	6,7219	0,741
Light Gradient Boosting	8,818	5,8412	7,2032	0,725
Elastic Net	8,820	5,8421	6,4675	0,744
Ridge	8,831	5,8494	6,4717	0,745
Least Angular Regressor	8,831	5,8494	6,4717	0,745
Orthogonal Matching Pursuit	8,831	5,8494	6,4717	0,745
Linear	8,831	5,8494	6,4717	0,745
Huber	8,855	5,8647	6,5003	0,743
K Neighbors	8,974	5,9432	6,6119	0,753
AdaBoost	9,099	6,0256	6,7725	0,753
Croston	9,253	6,1296	6,8935	0,788
Auto ARIMA	9,436	6,2550	7,9169	0,783
Lasso Least Angular Regressor	9,812	6,4989	8,6020	0,806
Naive Forecaster	12,013	7,9556	9,5993	1,070
Seasonal Naive Forecaster	13,321	8,4555	10,8745	1,154

Fonte: Elaborada pelo autor.

Tabela 5 - Valores das métricas resultantes da predição da série temporal CRUD, utilizando o modelo Random Forest

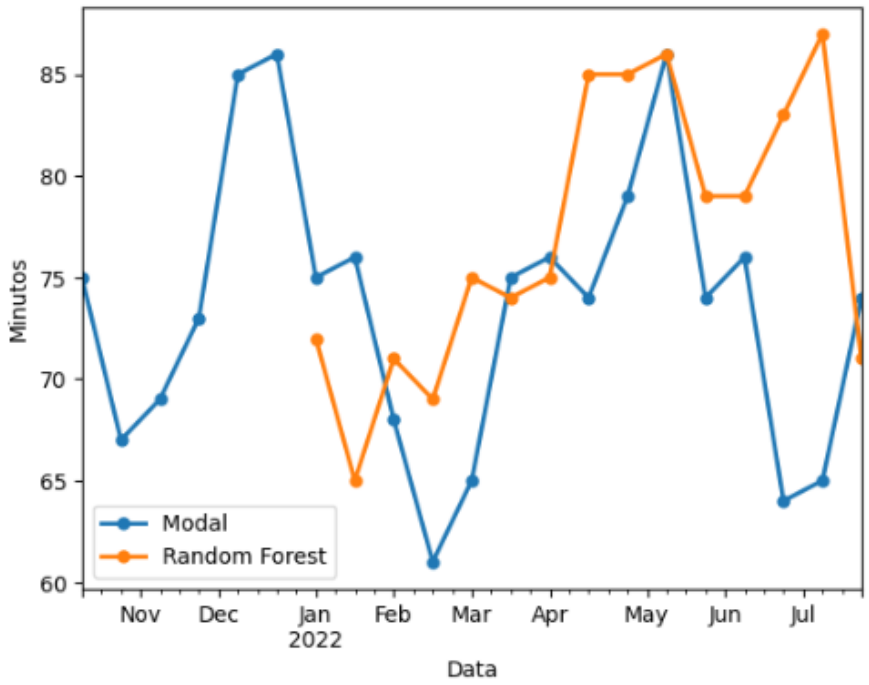
Modelo	MASE	MAE	RMSE	MAPE
Random Forest	2,403	9,8191	5,9134	0,222
K Neighbors	4,987	10,1975	7,5237	0,282
Extra Trees	5,134	10,2553	9,1460	0,317
Decision Tree	5,234	14,3226	10,2435	0,347
Gradient Boosting	5,281	15,5206	12,7983	0,372
Huber	6,365	16,2763	12,8776	0,411
AdaBoost	6,372	17,3121	14,6664	0,429
Light Gradient Boosting	7,199	17,5921	18,4783	0,464
Least Angular Regressor	8,251	17,6683	18,3553	0,517
Linear	9,610	18,8250	29,6770	0,572
Ridge	9,640	30,6087	38,1699	0,738
Elastic Net	9,712	38,8518	102,590	0,753
Bayesian Ridge	9,755	57,5398	108,626	0,757
Lasso	10,134	58,9839	108,717	0,798
Naive Forecaster	13,182	59,3397	119,101	0,838
ETS	13,481	62,8978	120,918	0,890
Theta Forecaster	14,735	78,8917	138,189	0,943
Auto ARIMA	16,682	127,970	189,102	0,959
Exponential Smoothing	16,821	128,687	192,666	0,997
Orthogonal Matching Pursuit	16,902	157,558	193,783	0,999
ARIMA	17,953	158,423	203,812	1,097
Croston	18,887	204,608	286,236	1,120
Seasonal Naive Forecaster	18,893	218,609	295,247	1,149
Lasso Least Angular Regressor	18,942	223,365	299,309	1,267
Grand Means Forecaster	18,954	233,374	300,666	1,289
Polynomial Trend Forecaster	18,991	233,428	301,667	1,398

Fonte: Elaborada pelo autor.

O modelo Random Forest obteve os melhores resultados nas 3 séries temporais junto a outros modelos que também utilizam *decision trees*. Vale ressaltar o modelo K Neighbors (Knn), que também obteve bons resultados. Nas **Figuras 7**, **8** e **9**, os dados preditos pelo modelo Random Forest foram representados pela cor laranja, junto aos dados reais em azul, é

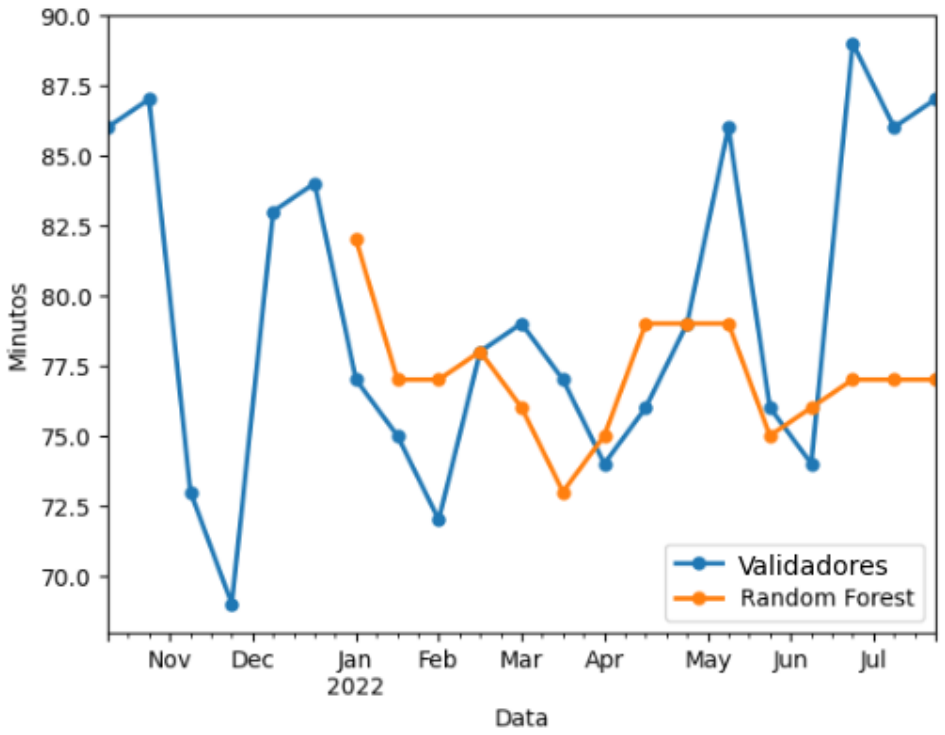
possível notar que, apesar de poucos dados para treino e teste, em vários pontos o modelo conseguiu seguir chegar a resultados precisos ou um comportamento muito parecido mesmo não acertando nenhum valor, que pode ser observado na **Figura 9** entre os meses de Fevereiro e Abril de 2022.

Figura 7 - Comparação entre os valores previstos pelo modelo Random Forest e os valores reais da série temporal Modal



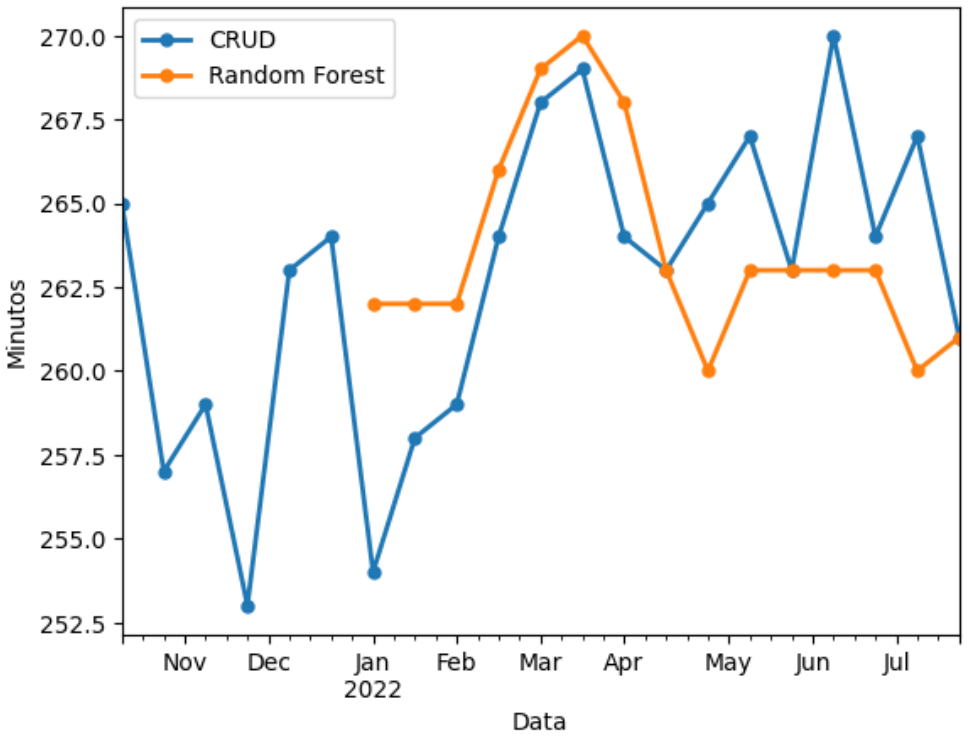
Fonte: Elaborada pelo autor.

Figura 8 - Comparação entre os valores previstos pelo modelo Random Forest e os valores reais da série temporal Validadores



Fonte: Elaborada pelo autor.

Figura 9 - Comparação entre os valores previstos pelo modelo Random Forest e os valores reais da série temporal CRUD



Fonte: Elaborada pelo autor.

Ao analisar as métricas, a ST Validadores obteve melhor resultado com o modelo, alcançando 5,3 na métrica MAE. A métrica é a média da diferença absoluta entre o realizado e o previsto, e, com a visualização da **Tabela 4**, é possível interpretar que o modelo tem uma média de erro de aproximadamente 5 minutos.

As STs Modal e CRUD, ambas tiveram 9,8 na métrica MAE, mas abertas a diferentes interpretações. Se utilizarmos o mesmo raciocínio da ST Validadores e interpretar que o modelo tem uma média de erro de aproximadamente 10 minutos, o modelo teve um desempenho um pouco melhor na ST CRUD. Se a análise for feita a partir dos dados das duas STs, a ST CRUD está na faixa dos 260 minutos, e a ST Modal está na faixa dos 80 minutos. Caso o modelo tenha esses 10 minutos de erro, é mais impactante em tarefas que levam menos tempo para serem realizadas, refletindo de maneira indireta em outras métricas em que a ST CRUD obteve melhores resultados.

2.4 Resultados: Random Forest

Após a análise dos testes, o modelo Random Forest foi criado e treinado pelas funções *create_model* e *finalize_model*, e os próximos valores da série temporal previstos pela função *predict_model*.

```
# Treina e avalia o desempenho do modelo Random Forest
model = create_model('rf_cds_dt')

# Treina o modelo em todo o conjunto de dados
final = finalize_model(model)

# Prevê próximos valores utilizando o modelo treinado
predict = predict_model(final, fh = 1)
```

Durante o desenvolvimento deste trabalho 8 *sprints* se passaram, e seus valores não foram utilizados nos treinamentos e testes do modelo. Os valores foram comparados com os valores preditos pela função *predict_model*, obtendo os valores das métricas na **Tabela 6**. Para cada execução da função, a série temporal foi atualizada com os novos valores, nesse caso, ao prever os valores da sprint de 19/08/2022, o valor real da sprint de 05/08/2022 foi adicionada na série temporal. As 8 execuções da função levaram em média 0,1 segundo para retornar o resultado.

Tabela 6 - Valores da métrica MAE para cada valor previsto utilizando o modelo Random Forest em cada série temporal.

Data	Modal (MAE)	Validadores (MAE)	CRUD (MAE)
05/08/2022	12,6428	5,7954	7,4853
19/08/2022	16,3738	5,1948	7,3456
02/09/2022	8,4563	7,5438	8,1352
16/09/2022	10,456	5,1462	15,1346
30/09/2022	8,6389	6,1865	11,5163
14/10/2022	11,4868	5,2345	9,4234
28/10/2022	9,3565	5,8765	10,3684
11/11/2022	15,7867	6,3967	9,3526

Fonte: Elaborada pelo autor.

Analisando a **Tabela 6**, a série temporal de Validadores teve o melhor resultado com o Random Forest, atingindo 5,9 na métrica MAE, um pouco a mais que nos testes. CRUD continuou com 9,8 e Modal, que teve o pior desempenho entre as 3 séries, obteve 11,65 na métrica MAE.

Por fim, é possível concluir que, apesar da quantidade de dados e da falta de padrão deles, foi possível obter bons resultados, mesmo não tendo uma precisão de acerto, é possível utilizar esse modelo em projetos ágeis.

4. Conclusão

O presente trabalho apresentou o desenvolvimento de um modelo para predizer o tempo estimado de execução de tarefas de projetos ágeis, utilizando dados da empresa JExperts. Durante o desenvolvimento do TCC, foi analisada a fundamentação teórica em relação ao conceito de séries temporais e *machine learning* aplicado às séries temporais, levantando trabalhos relacionados à predição de séries temporais para contribuir com os estudos.

O modelo teve o objetivo de avaliar a viabilidade de utilizar predição de séries temporais para determinar o tempo gasto para executar tarefas em projetos ágeis. A partir dos dados disponibilizados pela empresa JExperts, foram criadas séries temporais, as quais foram utilizadas em modelos preditivos da biblioteca PyCaret, dentro do Jupyter Notebook, e seus resultados, avaliados.

Para aplicá-la na prática, a primeira dificuldade foi encontrar uma base de dados. Não há vantagens para as empresas deixarem dados dessa natureza públicos, portanto, os dados testados permanecem privados. O segundo obstáculo, o qual demandou mais tempo de desenvolvimento, foi a montagem das séries temporais. Entretanto, devido à impossibilidade de montar uma consulta SQL que resultasse em uma série temporal pronta para os testes (devido aos tipos de informações), os dados foram separados manualmente. Após a separação prévia por SQL, o resultado foi a transformação do grande volume de 10 mil dados, em três séries temporais de aproximadamente 110 dados cada.

Felizmente, o número baixo de dados em cada ST não impediu que os testes obtivessem bons resultados. A desvantagem de possuir poucos dados mostrou que o modelo *Random Forest*, além de retornar resultados rapidamente, soube trabalhar com menor volume de dados, chegando em uma média de erro de 5 minutos em uma das STs, não sendo necessário ter uma grande quantidade, mas sim dados consistentes que sigam uma periodicidade (nesse caso, de 2 semanas). Para tanto, a equipe de desenvolvimento precisa estar comprometida em especificar e atualizar as tarefas, de forma que seja possível contribuir com a evolução dos planejamentos de sprint em equipes de desenvolvimento.

Para trabalhos futuros, pode-se citar a criação de um modelo preditivo específico para séries temporais de projetos ágeis, a criação de uma ferramenta que monte séries temporais com dados de projetos ágeis, capaz de realizar a análise estatística da série antes de aplicação dos métodos e a automação do processo como um todo.

5. Referências

DIGITAL.IA. **15th State of the Agile Report**, 2021. Disponível em: <https://itnove.com/wp-content/uploads/2021/07/15th-state-of-agile-report.pdf>. Acesso em: Maio de 2022.

EVANS, J. R. **Business Analytics: The Next Frontier for Decision Sciences**, 2012. Disponível em: <http://faculty.cbpp.uaa.alaska.edu/afef/business_analytics.htm> Acesso em: Maio 2022.

GONÇALVES, L. **Scrum: The methodology to become more agile**. Universidade de Coimbra, 2018.

KOECH, D. KIMUTAI. **A Complete Guide on How to Impute Missing Values in Time Series in Python**. Section, 2022. Disponível em: <https://www.section.io/engineering-education/missing-values-in-time-series/> Acesso em: Novembro 2022.

LATORRE, M. R. D. O. ;CARDOSO, M. R. A. **Análise de séries temporais em epidemiologia: uma introdução sobre os aspectos metodológicos**. Revista Brasileira de Epidemiologia, 2001.