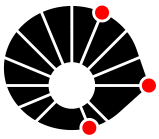

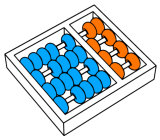

Gerenciador de Modelos para Séries Temporais Univariadas

Raphael de Oliveira Rodrigues Giron *Hélio Pedrini*

Relatório Técnico - IC-PFG-19-49
Projeto Final de Graduação
2019 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Gerenciador de modelos para séries temporais univariadas

Raphael de Oliveira Rodrigues Giron* Hélio Pedrini†

Resumo

Este projeto descreve a implementação de uma aplicação para análise, visualização, gerenciamento e seleção de modelos de aprendizado de máquina para séries temporais univariadas. A sua utilização visa facilitar a criação de modelos a partir de diferentes algoritmos, cada um com um conjunto de parâmetros específicos. A partir dos modelos gerados por meio de heurísticas baseadas em funções de erro e testes estatísticos, o melhor modelo é apresentado ao usuário, juntamente com os outros modelos criados, de modo que seja possível uma melhor interpretação do impacto que os parâmetros exercem sobre cada modelo.

*Instituto de Computação, Universidade Estadual de Campinas, 13083-852 Campinas, SP.
†Instituto de Computação, Universidade Estadual de Campinas, 13083-852 Campinas, SP.

Sumário

1 Introdução

Ciência de dados é o estudo de dados, no qual estão inseridos a extração, análise, visualização, gerenciamento e armazenamento de dados para a criação de *insights* e predições que potencialmente podem contribuir em decisões em um plano de negócios. Este campo de estudo é altamente multidisciplinar, com raízes em áreas como estatística, matemática e ciência da computação. Neste contexto, predições e análises de modelos de séries temporais geralmente requerem supervisão humana com a finalidade de selecionar a melhor combinação de parâmetros para a geração de modelos.

Para um melhor entedimento do projeto, faz-se necessária a apresentação de uma nomenclatura básica para a análise de séries temporais. Denominam-se *séries temporais* um conjunto de valores que uma variável assume no decorrer do tempo. Uma *tendência* ocorre quando existe, a longo prazo, um termo crescente ou decrescente nos dados. Um padrão *sazonal* ocorre quando a série temporal é afetada por fatores sazonais, como o período do ano ou dias da semana. A sazonalidade está sempre atrelada a uma frequência conhecida. Um *ciclo* ocorre quando os dados apresentam altas e quedas que não seguem uma frequência fixa, diferentemente do comportamento sazonal. *Resíduo* é a diferença entre o dado predito por algum modelo de aprendizado de máquina e o dado real.

Testes estatísticos fornecem mecanismos para a tomada de decisões quantitativas sobre um processo, com a finalidade de determinar se existe ou não evidências suficientes para a rejeição da conjectura ou hipótese sobre o processo em questão.

Este projeto fornece uma metodologia para a seleção e o gerenciamento de modelos de aprendizado de máquina para séries temporais, utilizando arquivos de configuração para facilitar a criação de modelos a partir de diferentes algoritmos, cada um com um conjunto de parâmetros específicos. A partir dos modelos gerados, por meio de heurísticas baseadas em funções de erro e testes estatísticos, o melhor modelo é apresentado ao usuário, juntamente com os outros modelos criados, para que assim seja possível uma melhor interpretação do impacto que os parâmetros exercem sobre cada modelo.

2 Objetivos

A finalidade do produto final é a criação de uma ferramenta computacional que agiliza o processo de avaliação e geração de modelos de aprendizado de máquina para séries temporais, facilitando o trabalho de um cientista de dados ou de *stakholders* que possuem somente conhecimento do domínio do problema, uma vez que se abstraiu grande parte do ferramental de programação necessário para a geração dos modelos. Atualmente, a ferramenta comporta os seguintes modelos de aprendizado de máquina: Prophet, Arima Sazonal (SARIMAX) e Holt-Winters.

O intuito deste projeto não é a criação de um modelo para predições ou análises de comportamento de um problema específico, mas sim a elaboração de um software que suporta a criação de diversos modelos para auxiliar um usuário em suas análises.

3 Metodologia

Nesta seção são apresentadas as principais etapas que compõem a metodologia da ferramenta desenvolvida.

3.1 Fluxograma

Para melhor explanar a ferramenta criada, pode-se separar suas ações em diferentes etapas, cada qual com uma função específica. Para tanto, inicialmente são necessários alguns arquivos de configuração:

- arquivo no formato `csv` com os dados pertinentes ao problema.
- arquivo `json`, chamado *model_params.json* com os dados para a geração dos modelos e predições. Entre os dados estão o nome da coluna do arquivo `csv` que a série temporal irá retratar, a frequência do agrupamento a ser realizado, a operação de agregação, período em que os modelos deverão realizar a predição e a heurística para a seleção do melhor modelo.
- arquivos de configuração no formato `json` contendo os parâmetros para cada um dos modelos referentes aos algoritmos pré-estabelecidos: Prophet (*prophet_params.json*), SARIMAX (*arima_params.json* e Holt-Winters (holtwinters_params.json).

A partir dos arquivos citados anteriormente, as etapas a seguir são seguidas.

3.1.1 Tratamento dos dados

- A partir do arquivo `csv`, gera-se a série temporal do problema em questão, que é analisada conforme especificações do arquivo *model_params*.
- Os dados analisados são separados em um conjunto de treinamento, com 70% dos dados totais, e um conjunto de teste com os dados restantes.

3.1.2 Geração de modelos, validação e predição

- Após o tratamento inicial dos dados, gera-se uma lista de modelos segundo os parâmetros especificados nos arquivos *prophet_params.json*, *holtwinters_params.json* e *arima_params.json*.
- Os modelos gerados são treinados com o mesmo conjunto de dados de treinamento e, em seguida, são testados sobre o método heurístico escolhido no arquivo *model_params.json*, selecionando o melhor.
- É realizada a predição do período especificado em todos os modelos construídos.

3.1.3 Visualização

- são criadas abas para a visualização e informações de cada modelo criado;
- são disponibilizadas tabelas com informações sobre as métricas de erro, testes estatísticos e parâmetros utilizados para a geração dos modelos.
- são criados cinco gráficos: o primeiro contendo a série temporal referente ao conjunto de treinamento e o modelo gerado a partir deste conjunto; o segundo contendo o resíduo do modelo criado neste intervalo de dados; o terceiro contendo o conjunto de dados integral (treinamento e teste) e a predição realizada para o conjunto de teste; o quarto contendo o resíduo entre o conjunto de teste real e a predição realizada; o quinto contendo a predição no intervalo especificado no arquivo *model_params.json*.

3.2 Arquivos de Implementação

Para a implementação da ferramenta, utilizou-se a linguagem de programação *Python*, encapsulando-o em diversos módulos. A seguir são listados os arquivos da aplicação, assim como suas funções no projeto:

- **dataframe_handler.py:** implementa rotinas para o tratamento dos dados. Para tanto, foi escolhida a biblioteca `Pandas` e os dados foram manipulados por meio de *dataframes*. Neste módulo, rotinas são implementadas para a divisão do conjunto de treinamento e teste, agrupamento e agregação dos dados.
- **io_handler.py:** implementa rotinas para a leitura dos arquivos de configuração e dados, fornecidos como entrada para a ferramenta.
- **base_model.py:** classe pai para classes de modelos específicas.
- **arima.py:** classe que implementa rotinas como *fit*, validação e predição para o modelo Arima Sazonal.
- **holt_winters.py:** classe que implementa rotinas como *fit*, validação e predição para o modelo Holt-Winters.
- **prophet.py:** classe que implementa rotinas como *fit*, validação e predição para o modelo Prophet.
- **model_tools.py:** implementa algumas funcionalidades extras genéricas para todos os diferentes modelos de aprendizado de máquina, como obter resíduos, gerar modelos, predições por *dataframes* ou períodos, seleção do melhor modelo.
- **metrics.py:** implementa rotinas que calculam as métricas de erro, testes estatísticos e heurísticas para a seleção dos modelos.
- **plots.py:** implementa rotinas para a geração dos gráficos dos modelos construídos.

- **tables.py:** implementa rotinas para a criação das tabelas contendo os dados referentes as métricas de erro e testes estatísticos e dados dos parâmetros escolhidos para a geração dos modelos.
- **model_tabs.py:** responsável pelo *layout* de cada aba, contendo os gráficos e tabelas referentes a cada modelo criado.
- **app.py:** chama as rotinas dos módulos anteriores de maneira a criar o fluxograma explicado na Seção 3.1.

4 Modelos

A ferramenta foi projetada de maneira a suportar alguns modelos de aprendizado de máquina para séries temporais pré-selecionados. Eles são *Prophet*, *Holt-Winters* e *Arima Sazonal*. Nas seções seguintes, uma breve introdução sobre cada um deles é fornecida.

4.1 Prophet

O modelo de predição Prophet é um arcabouço de código aberto criado pelo Facebook. Este modelo utiliza outro modelo de decomposição de séries temporais [1], que contém três componentes principais: tendência, sazonalidade e feriados. Elas são combinadas de acordo com a seguinte equação:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{1}$$

em que $g(t)$ é a função tendência que modela mudanças não periódicas nos valores da série temporal, $s(t)$ representa mudanças periódicas (com frequências semanais ou anuais, por exemplo) e $h(t)$ representa os efeitos de feriados, que podem acarretar em consequências por períodos irregulares, durante um ou mais dias. O termo ϵ_t representa qualquer mudança idiossincrática não comportada pelo modelo. Normalmente, assume-se que este termo possui uma distribuição normal. Além disso o modelo apresenta uma margem de confiança referente a predição. A seguir, algumas noções básicas acerca do funcionamento do modelo são fornecidas. Para uma melhor compreensão, o leitor pode consultar [].

4.1.1 Tendência

Prophet implementa dois tipos de modelos para a modelagem da componente referente a tendências: modelo de crescimento com saturação e modelo linear.

$$g(t) = \frac{C(t)}{1 + exp(-k + a(t)\delta)(t - (m + a(t)\gamma))} \tag{2}$$

Para o entendimento da equação do modelo de crescimento com saturação (Equação 3), algumas definições são necessárias. $C(t)$ é a capacidade intrínseca da modelagem em que é expressa a saturação da tendência. Alguns pontos de mudança (*changepoints*) são incorporados ao modelo, em que a taxa de crescimento pode ser alterada. Sejam S *changepoints* em tempos s_j, $j = 1, \ldots, S$. Um vetor de ajustes $\gamma \in \mathcal{R}^S$ é definido tal que γ_j é a mudança

na taxa que ocorre no tempo s_j. A taxa de crescimento em um tempo t é a soma da taxa base k somada a todos os ajustes até o tempo t. Isso pode ser representado por um vetor $a(t) \in \{0,1\}^S$ tal que

$$a_j(t) = \begin{cases} 1, & \text{se } t \geq s_j, \\ 0, & \text{caso contrário.} \end{cases}$$

Quando a taxa k é ajustada, o parâmetro de *offset* m também necessita ser ajustado. O ajuste em um *changepoint* j é computado por:

$$\gamma_j = \left(s_j - m - \sum_{l<j} \gamma_l \right) \left(1 - \frac{k + \sum_{l<j} \delta_l}{k + \sum_{l \leq j} \delta_l} \right)$$

Para problemas que não exibem comportamentos de saturação, a Equação 3 exprime a curva de modelagem.

$$g(t) = \frac{C(t)}{1 + exp(-k + a(t)\delta)(t - (m + a(t)\gamma))} \tag{3}$$

4.1.2 Sazonalidade

Os modelos se baseiam na série de Fourier para prover flexibilidade acerca de efeitos periódicos [2]. Seja P um período regular o qual é esperado que a série temporal possua (por exemplo, $P = 365,5$ para dados anuais ou $P = 7$ para dados semanais, em que a unidade de P são dias). É possível aproximar efeitos sazonais pela Equação 4

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P} \right) + b_n \text{sen}\left(\frac{2\pi nt}{P} \right) \right) \tag{4}$$

Para a modelagem da sazonalidade, é necessário estimar $2N$ parâmetros $\beta = [a_1, b_1, \ldots, a_N, b_N]^T$.

4.2 Holt-Winters

Este método utiliza três equações de suavização exponencial: uma para o nível l_t, uma para a tendência b_t e uma para a sazonalidade s_t, com parâmetros de suavização α, β^* e γ, respectivamente. Neste projeto, o modelo Holt-Winters utilizado foi implementado pela biblioteca `statsmodels`.

Existem duas variações deste método que diferem devido a propriedades de distintas sazonalidades. O método *aditivo* é utilizado quando as variações sazonais são constantes pela série temporal, enquanto o método *multiplicativo* é preferível quando as variações sazonais mudam proporcionalmente com o decorrer da série. Nas equações a seguir, m é usado para denotar a frequência da sazonalidade.

4.2.1 Método Aditivo

O método aditivo é modelado pela Equação 5.

$$\hat{y}_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)} \tag{5}$$

em que

$$l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1})$$
$$b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$$
$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m},$$

em que k é a parte inteira de $(h-1)/m$, que assegura que as estimativas dos índices sazonais usadas para as predições venham do final do período estipulado como frequência. A equação para nível mostra uma média ponderada entre a observação da sazonalidade ajustada $(y_t - s_{t-m})$ e a predição não sazonal $(l_{t-1} + b_{t-1})$ para o tempo t. A equação de tendência é idêntica ao método Holt linear clássico. A equação sazonal é uma média ponderada entre o índice sazonal corrente $(y_t - l_{t-1} - b_{t-1})$ e o índice sazonal do mesmo período no ciclo anterior (m períodos atrás).

4.2.2 Método Multiplicativo

O método multiplicativo é modelado pela Equação 6.

$$\hat{y}_{t+h|t} = (l_t + hb_t)s_{t+h-m(k+1)} \tag{6}$$

em que

$$l_t = \alpha(\frac{y_t}{s_{t-m}}) + (1-\alpha)(l_{t-1} + b_{t-1})$$
$$b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$$
$$s_t = \gamma(\frac{y_t}{l_{t-1} + b_{t-1}}) + (1-\gamma)s_{t-m},$$

4.3 Arima Sazonal

O modelo arima sazonal é a combinação do modelo arima tradicional adicionado de parâmetros extras que induzem a modelagem de uma ampla gama de dados sazonais. A aplicação desenvolvida neste projeto utiliza o modelo arima sazonal (`SARIMAX`) implementado pela biblioteca `statsmodel`.

O modelo arima tradicional é composto da combinação da operação diferença de primeira ordem nos dados ($y'_t = y_t - y_{t-1}$) com modelos de auto-regressão e média móvel. Este modelo pode ser escrito conforme a Equação 7.

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t \tag{7}$$

em que y'_t é a série temporal após a realização da operação de diferença (que pode ser realizada mais de uma vez). Os termos no lado direito da equação 7 incluem valores com *lag* de y_t e dos erros. Este modelo é denominado `ARIMA`(p,d,q), em que:

- p = ordem referente ao modelo auto-regressivo.
- d = grau da diferença de primeira ordem.
- q = ordem referente ao modelo de média móvel.

Este modelo também pode ser reescrito utilizando a notação de *backshift*, conforme a Equação 8.

$$(1 - \phi_1 B - \cdots - \phi_p B^p)(1 - B)^d y_t = c + (1 + \theta_1 B + \ldots \theta_q B^q)\epsilon_t \tag{8}$$

O modelo arima sazonal consiste no acréscimo de parâmetros $(P, D, Q)m$, em que m é a periodicidade do modelo. A parte sazonal do modelo consiste de termos similares aos componentes do arima tradicional, mas com *backshifts* no período sazonal. Por exemplo, um modelo $\texttt{ARIMA}(1,1,1)(1,1,1)_4$, sem constante, pode ser escrito conforme a seguinte equação:

$$(1 - \phi_1 B)(1 - \Phi_1 B^4)(1 - B)(1 - B^4)y_t = (1 + \theta_1 B)(1 + \Theta_1 B^4)\epsilon_t$$

em que os termos adicionais sazonais são apenas multiplicados pelos termos não-sazonais.

5 Métricas de Erro

Durante o processo de criação de modelos de aprendizado de máquina, torna-se importante mensurar a qualidade da modelagem de acordo com objetivos intrínsecos à natureza do domínio do problema. Existem funções matemáticas que auxiliam a avaliação da capacidade do modelo em prever valores corretos. A seguir, introduziremos duas métricas de erro utilizadas no projeto.

5.1 Erro Médio Absoluto

O erro médio absoluto (*mean absolute error* - MAE), calculado pela Equação 9, é a média das diferenças em módulo entre os valores preditos e os valores reais. Deste modo, todas as diferenças possuem o mesmo peso, de maneira linear.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{9}$$

5.2 Raiz do Erro Médio Quadrático

A raiz do erro médio quadrático (*root mean squared error* - RMSE), calculado pela Equação 10, é a raiz da média das diferenças elevadas ao quadrado entre os valores reais e preditos pelos modelos. Devido à operação de potência ser realizada antes da radiciação, o RMSE fornece um peso relativamente alto para erros maiores. Desta maneira, esta métrica é melhor utilizada quando erros mais distantes são particularmente indesejáveis no modelo.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{10}$$

6 Testes Estatísticos

Testes estatísticos são ferramentas que podem ser utilizadas para inferências serem feitas acerca dos dados estudados, mostrando se um determinado padrão observado é significativo ou simplesmente casual.

Neste projeto, os testes estatísticos são realizados sobre os resíduos dos dados preditos no conjunto de teste. Resíduos de um bom modelo devem possuir média zero e são estacionários, ou seja, cujas propriedades não dependem do tempo em que seus valores são observados. Estas características configuram o que é chamado de *ruído branco*, cujo comportamento é imprevisível. De fato, um modelo que fornece um resíduo que é um ruído branco é um bom modelo, pois o comportamento não capturado é impossível de ser retratado. Um resíduo que não possui média zero certamente possui uma tendência não capturada. Do mesmo modo, um resíduo que possui um comportamento periódico definido possui alguma sazonalidade não detectada pelo modelo.

6.1 Teste Z

É um tipo de teste de hipótese onde, no contexto deste trabalho, verifica-se se a média do resíduo dos modelos é equivalente a zero. Com isto, é possível afirmar se a tendência contida na série temporal foi capturada ou não. Neste trabalho, foi escolhido o valor de significância $\alpha = 0,05$ para a realização dos testes.

6.2 Teste Ljung-Box

Este teste verifica a ausência de autocorrelação no resíduo dos modelos até um *lag* k. Em outras palavras, com a escolha adequada de k, este teste verifica se os dados a serem testados são ruídos brancos. Isto pois quando existe tendência na série temporal, a autocorrelação entre valores adjacentes ou próximos é alta e decai com o tempo, fazendo com que o teste falhe. Já quando existe comportamento sazonal na série temporal com periodicidade p, a autocorrelação entre variáveis distantes umas das outras de um lag p é alta, fazendo com que o teste falhe. Neste trabalho, o valor de significância $\alpha = 0,05$ foi escolhido para a realização dos testes.

7 Heurísticas de Seleção

Como potenciais métodos de seleção dos modelos, a serem escolhidos pelo usuário no arquivo *model_params.json* estão:

- **MAE:** onde os pesos dos erros não são ponderados e portanto têm uma relação linear entre valores preditos e valores reais;
- **RMSE:** onde os pesos da diferença entre os valores preditos e reais são ponderados, penalizando modelos com *outliers*, fazendo com que erros maiores aumentem o erro calculado de maneira mais acentuada que o MAE;

- **Teste Z+ MAE:** Adiciona, com caráter eliminatório o teste Z a métrica MAE, onde se o modelo falha no teste, ele não é escolhido. Desta maneira elimina-se modelos onde tendência não são capturadas.

- **Teste Z + RMSE:** Adiciona, com caráter eliminatório o teste Z a métrica RMSE, onde se o modelo falha no teste, ele não é escolhido. Desta maneira elimina-se modelos onde tendência não são capturadas.

- **Ljung-Box + MAE:** Adiciona, com caráter eliminatório o teste Ljung-Box a métrica MAE, onde se o modelo falha no teste, ele não é escolhido. Desta maneira elimina-se modelos onde tendências e sazonalidades não são capturadas.

- **Ljung-Box + RMSE:** Adiciona, com caráter eliminatório o teste Ljung-Box a métrica RMSE, onde se o modelo falha no teste, ele não é escolhido. Desta maneira elimina-se modelos onde tendências e sazonalidades não são capturadas.

8 Resultados e Visualização

Este projeto visa a criação de uma aplicação que facilita a análise, visualização, gerenciamento e seleção de modelos de machine learning para séries temporais univariadas. Portanto não temos como meta a criação de um tipo de modelos específico para a resolução de um tipo de problema direcionado. Para testar a aplicação foi utilizado um conjuntos de dados fictício que representa a o número de passageiros de uma companhia aérea dos anos de 1949 a 1960 [4]. Os modelos apresentados neste trabalho tiveram como método para seleção a heurística o Teste Ljung-Box + RMSE. Foi utilizada a biblioteca Bokeh para a visualização e interação em web browsers. O layout geral é exemplificado pela figura 1.

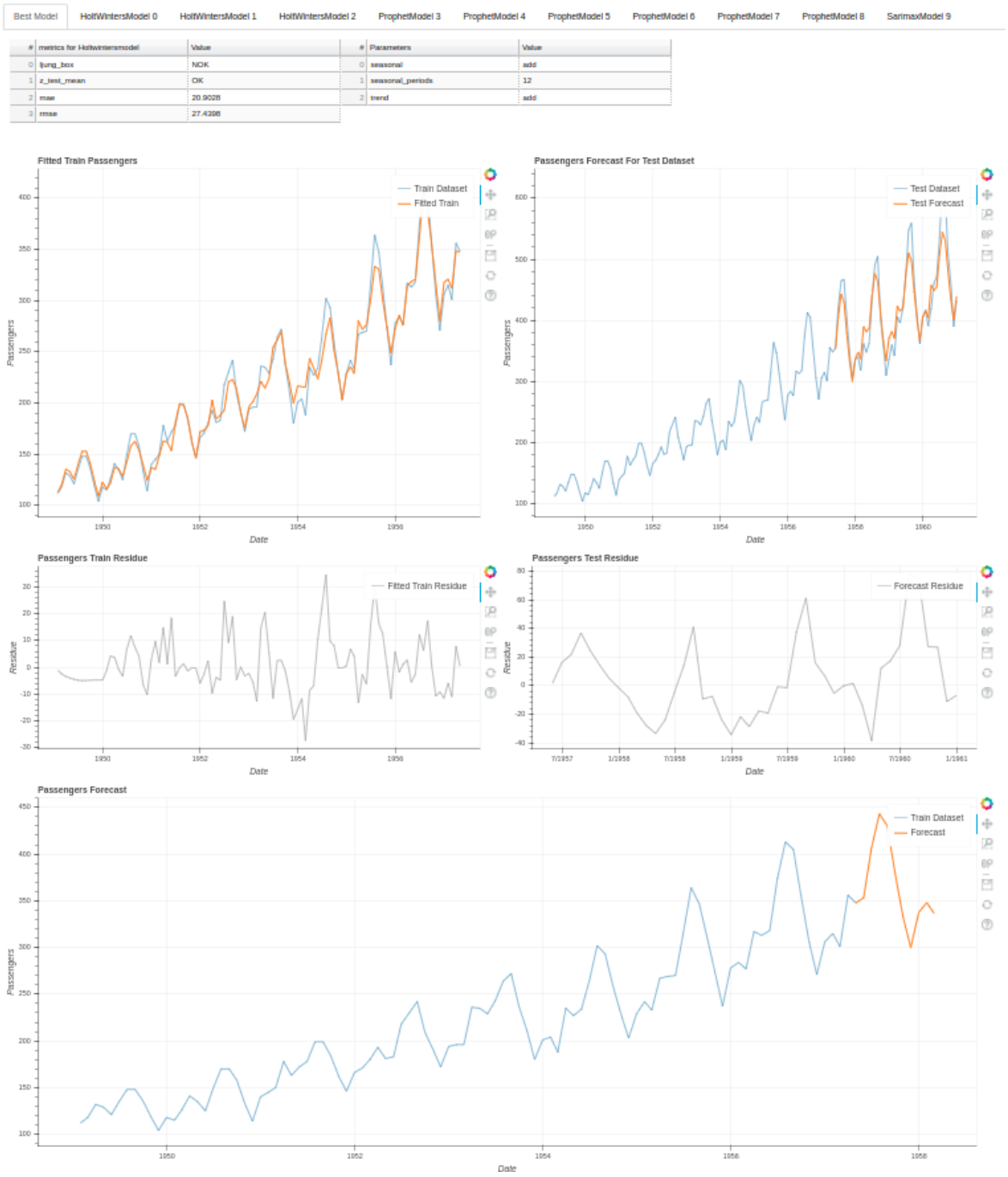

Figura 1: Layout da aplicação, exibindo os elementos de um modelo sobre a página web.

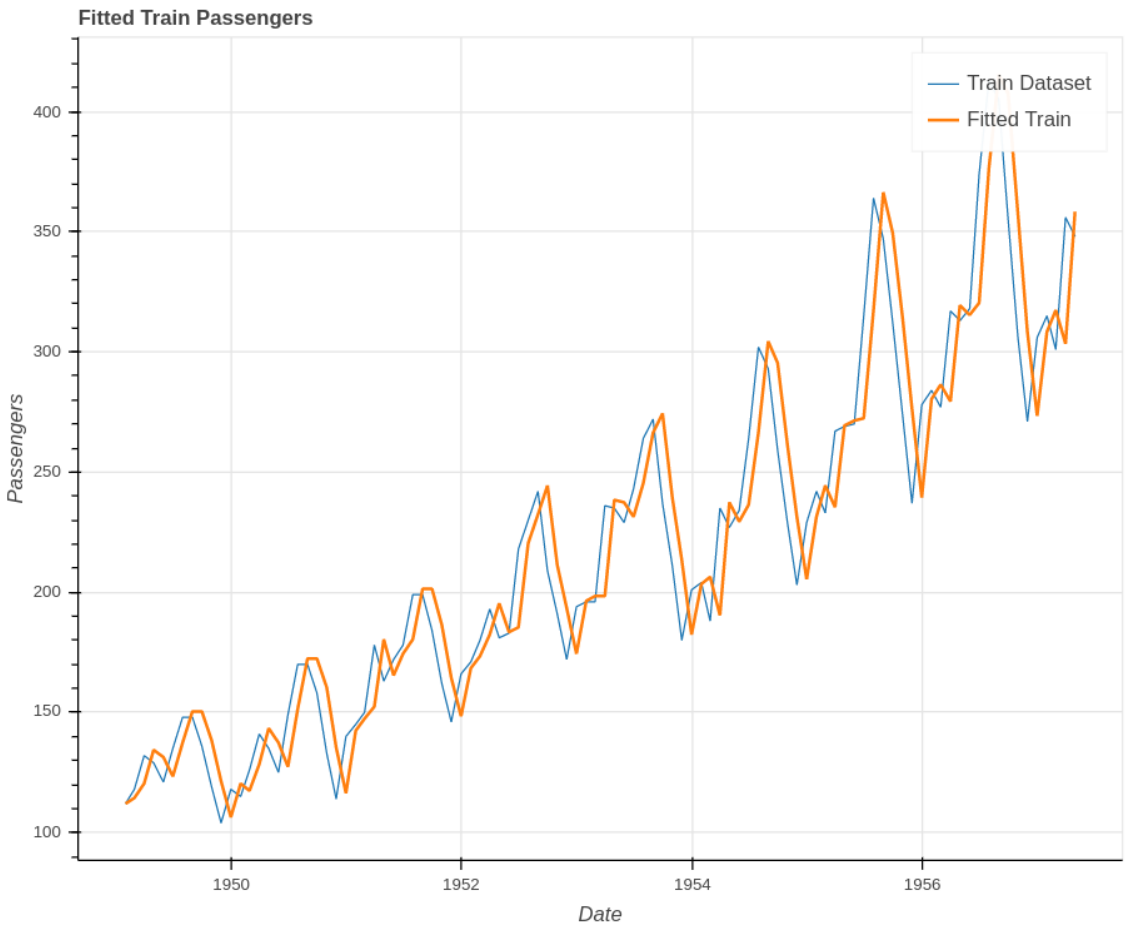

Figura 2: Modelo Holt Winters 0 fitado com o conjunto de treino. Em azul temos os dados de treino original e em laranja o fit do modelo.

8.1 Holt-Winters 0

Métricas	**Valor**
Teste Z	OK
Teste Ljung Box	NOK
MAE	53,9323
RMSE	69,9779

Parâmetros	Valor
seasonal$_p eriods$	6
trend	add

Tabela 1: A primeira tabela mostra as métricas e testes estatísticos para o modelo Holt-Winters 0. A segunda tabela mostra os parâmetros para a geração do modelo.

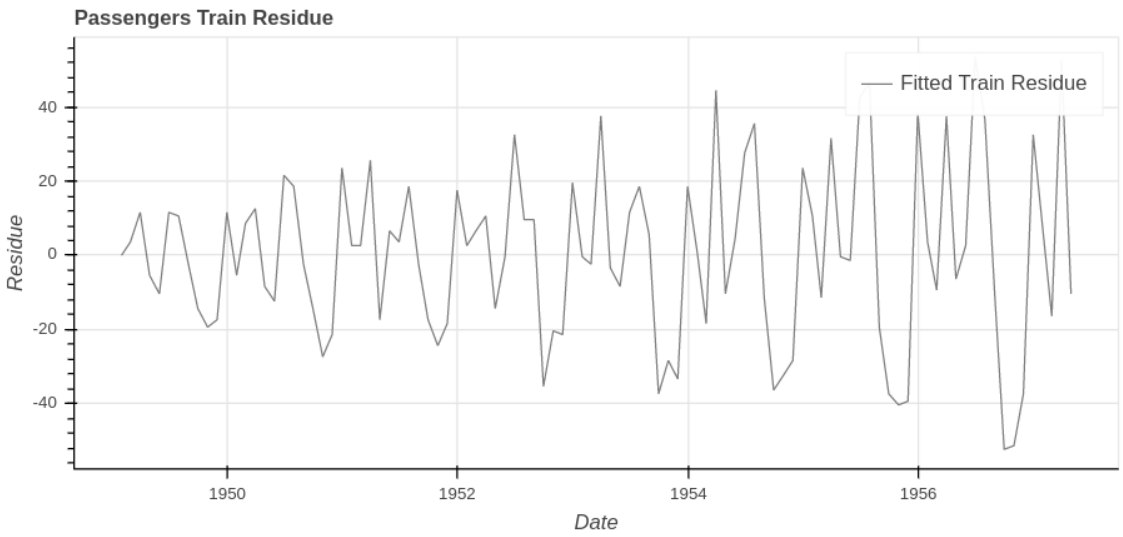

Figura 3: Resíduo do gráfico apresentado pela Figura 2.

8.2 Holt-Winters 1

Métricas	**Valor**
Teste Z	NOK
Teste Ljung Box	NOK
MAE	56,5064
RMSE	65,4493

Parâmetros	Valor
seasonal	add
$\text{seasonal}_p eriods$	6
trend	add

Tabela 2: A primeira tabela mostra as métricas e testes estatísticos para o modelo Holt-Winters 1. A segunda tabela mostra os parâmetros para a geração do modelo.

8.3 Holt-Winters 2

Métricas	**Valor**
Teste Z	OK
Teste Ljung Box	NOK
MAE	20,9028
RMSE	27,4398

Parâmetros	Valor
seasonal	add
$\text{seasonal}_p eriods$	12
trend	add

Tabela 3: A primeira tabela mostra as métricas e testes estatísticos para o modelo Holt-Winters 2. A segunda tabela mostra os parâmetros para a geração do modelo.

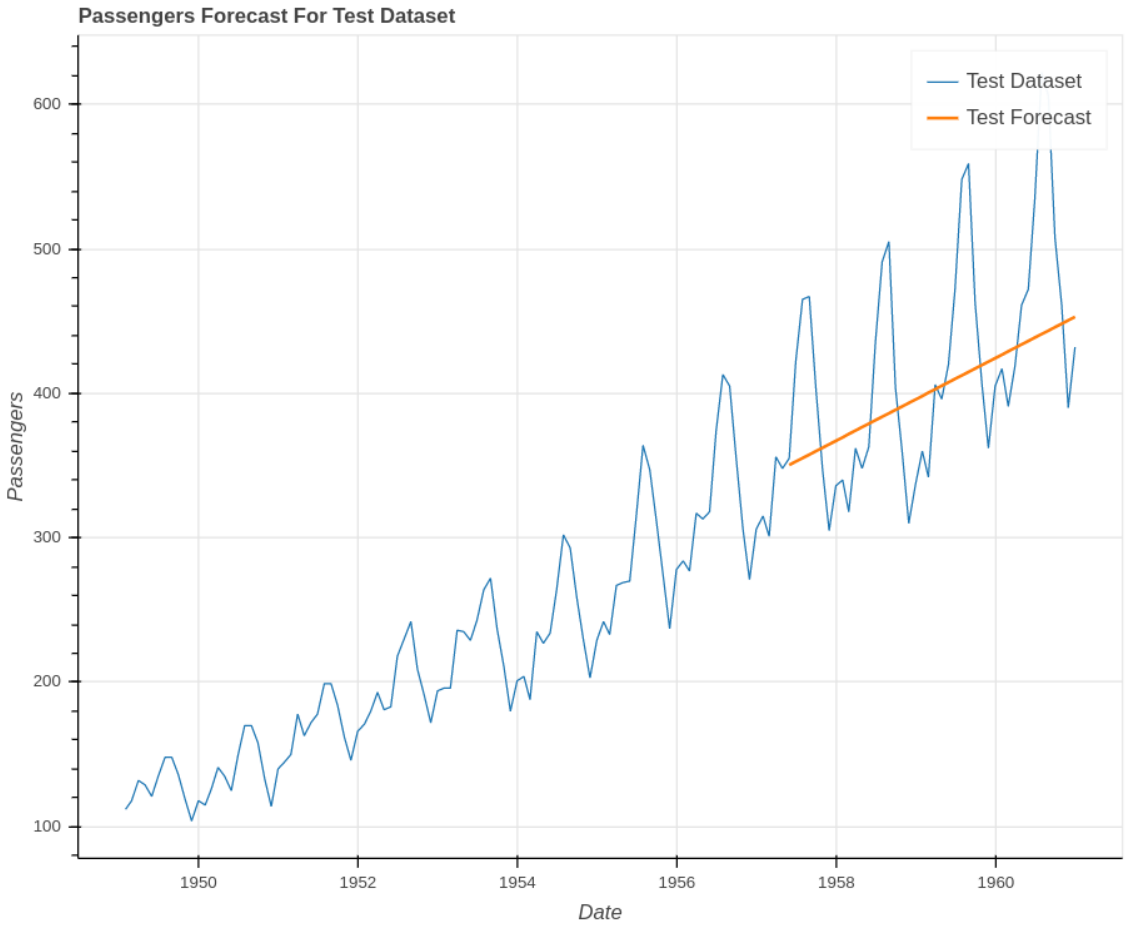

Figura 4: Projeção do modelo Holt Winters 0 sobre o conjunto de teste. Em azul temos o conjunto de dados original e em laranja a projeção do modelo.

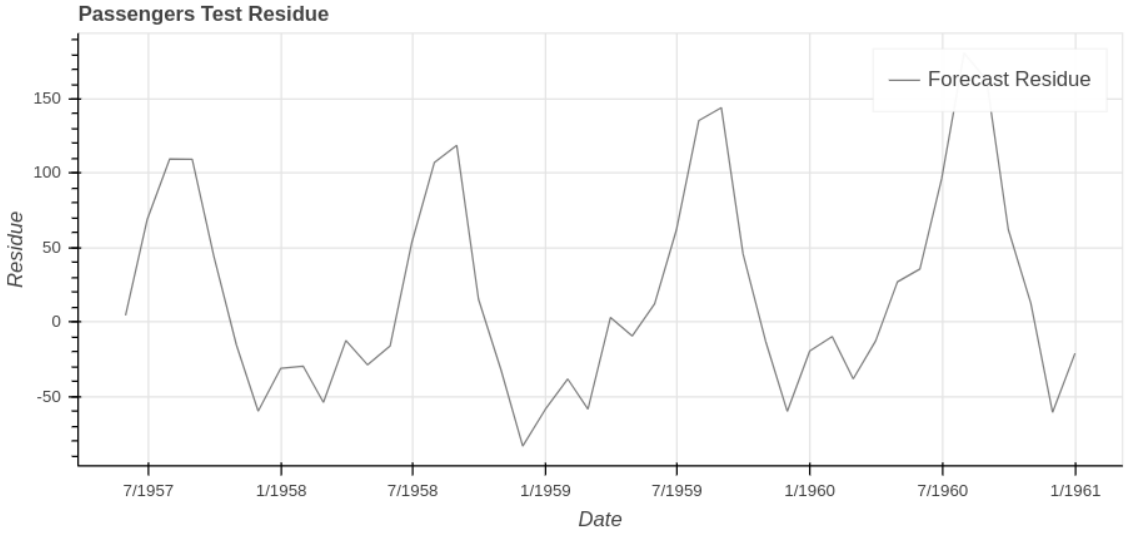

Figura 5: Resíduo do gráfico apresentado pela Figura 4.

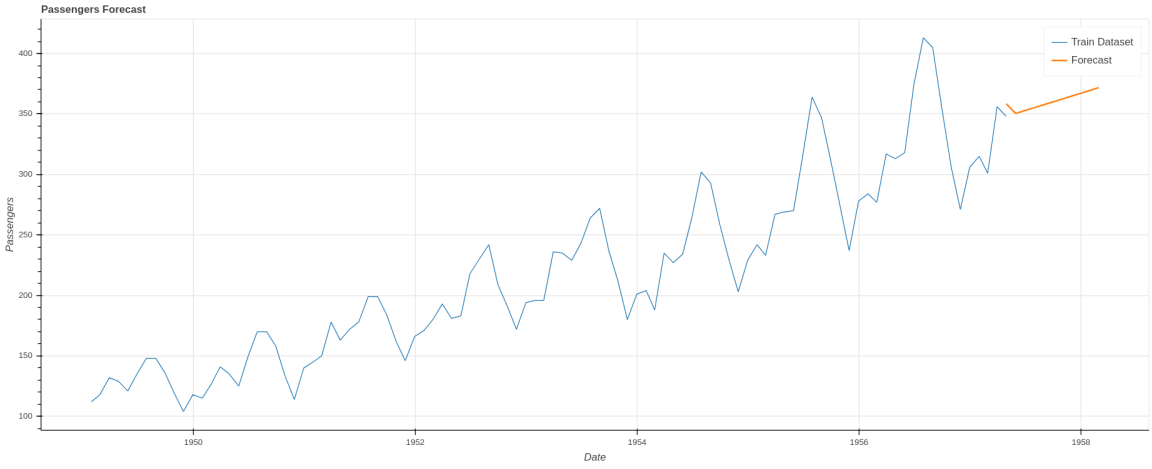

Figura 6: Projeção do modelo Holt-Winters 0 dez meses a frente do último dado fornecido pelo dataset.

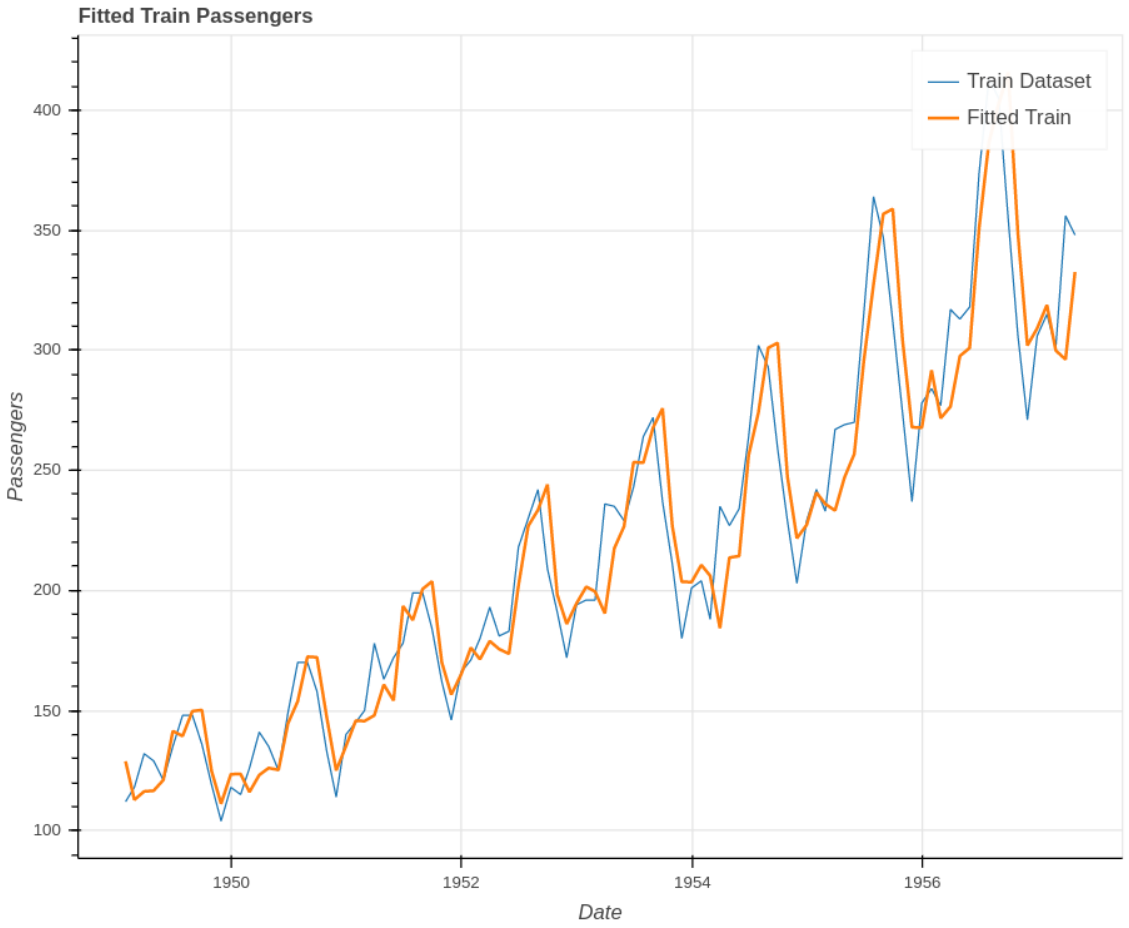

Figura 7: Modelo Holt Winters 1 fitado com o conjunto de treino. Em azul temos os dados de treino original e em laranja o fit do modelo.

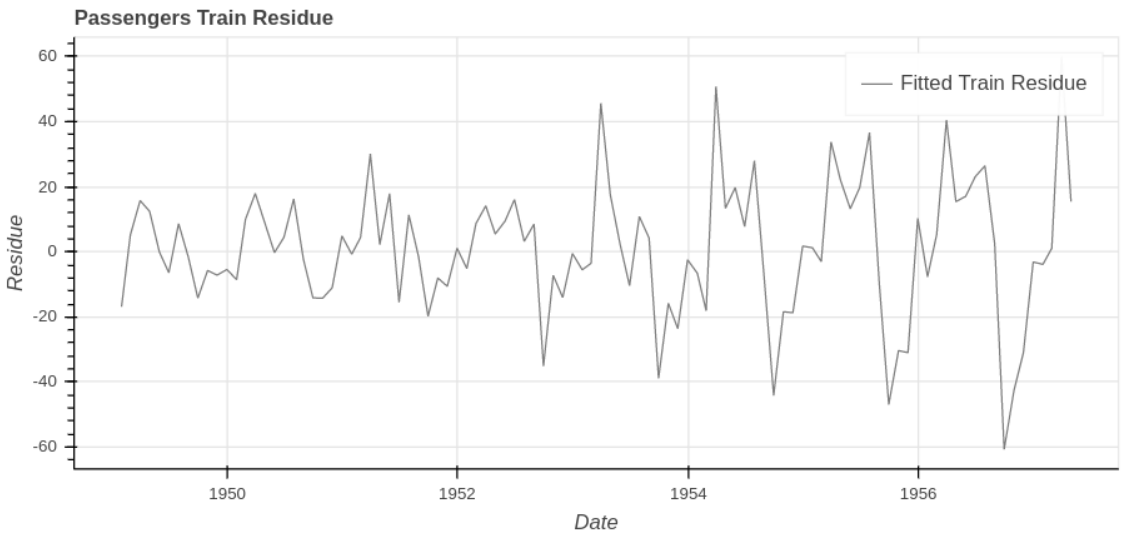

Figura 8: Resíduo do gráfico apresentado pela Figura 7.

8.4 Prophet 3

Métricas	**Valor**
Teste Z	NOK
Teste Ljung-Box	NOK
MAE	77,2119
RMSE	95,3972

Parâmetros Sazonais	Valor
fourier	5
name	monthly
period	30

Parâmetros	Valor
changepoint$_{p}rior_{s}cale$	0,01
mcmc$_{s}amples$	10
weekly$_{s}easonality$	5
yearly$_{s}easonality$	10

Tabela 4: A primeira tabela mostra as métricas e testes estatísticos para o modelo Prophet 3. A segunda e terceira tabelas mostram os parâmetros para a geração do modelo.

8.5 Prophet 4

Métricas	**Valor**
Teste Z	NOK
Teste Ljung-Box	NOK
MAE	37,6105
RMSE	53,1449

Parâmetros	Valor
changepoint$_{p}rior_{s}cale$	0,01
weekly$_{s}easonality$	5

Parâmetros Sazonais	Valor
Padrão	Padrão

Tabela 5: A primeira tabela mostra as métricas e testes estatísticos para o modelo Prophet 4. A segunda e terceira tabelas mostram os parâmetros para a geração do modelo.

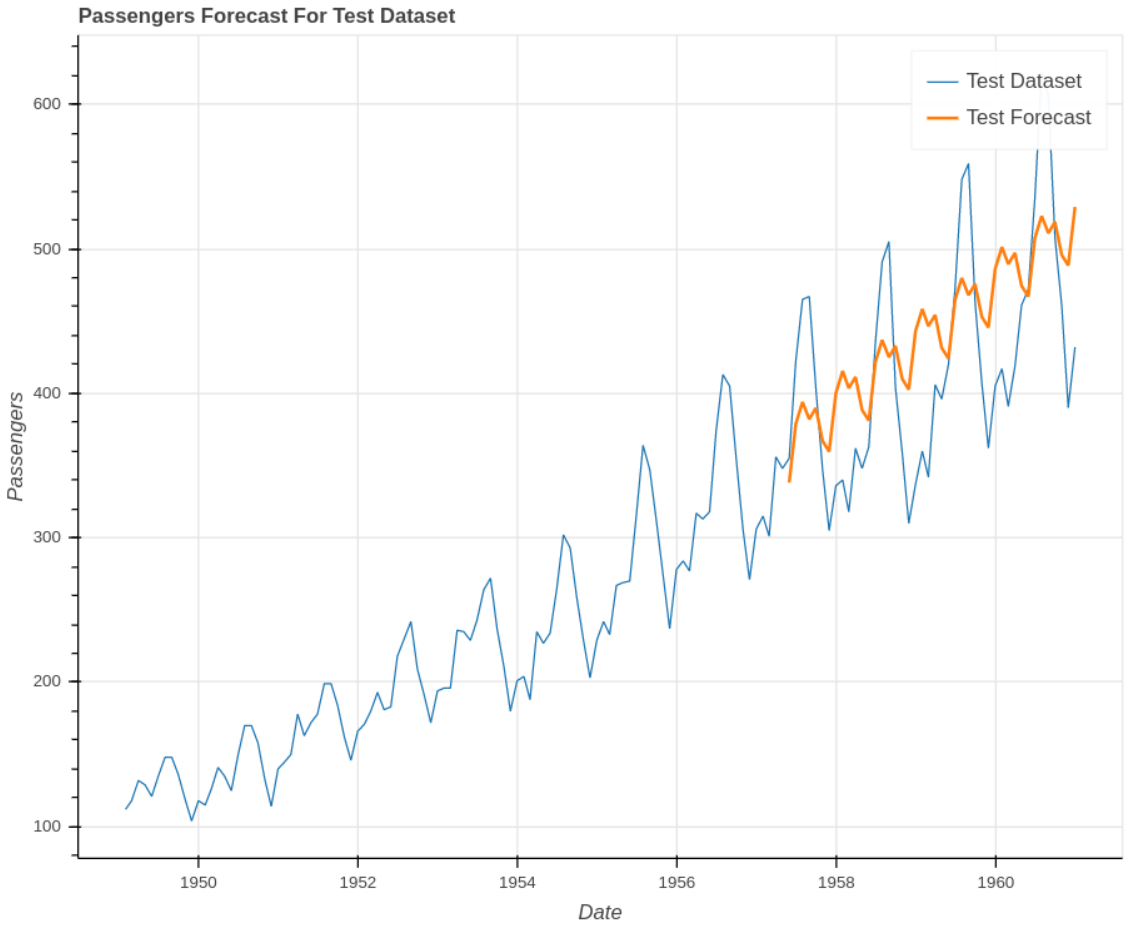

Figura 9: Projeção do modelo Holt Winters 1 sobre o conjunto de teste. Em azul temos o conjunto de dados original e em laranja a projeção do modelo.

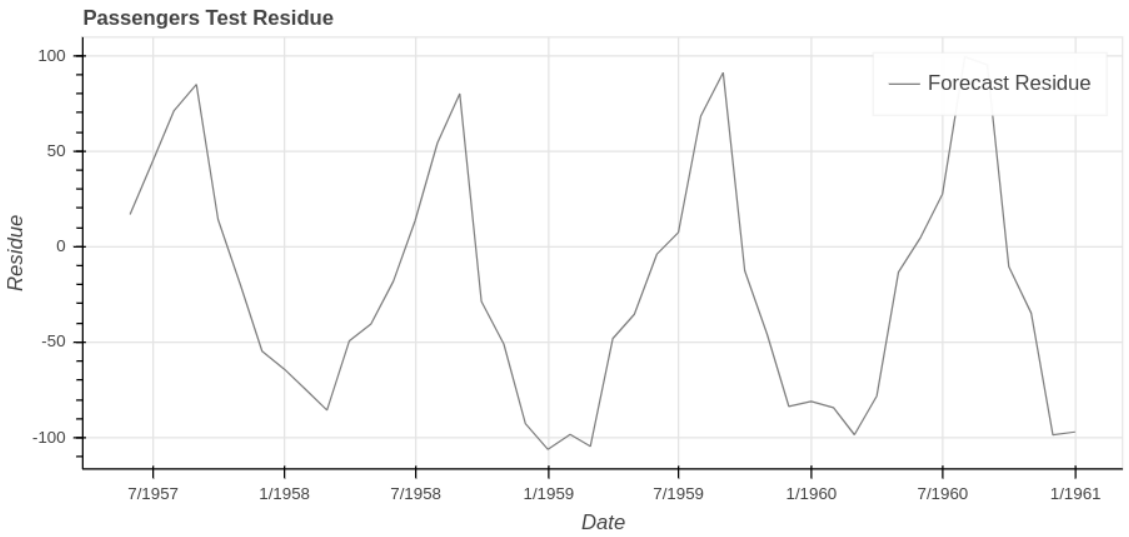

Figura 10: Resíduo do gráfico apresentado pela Figura 9.

8.6 Prophet 5

Métricas	**Valor**
Teste Z	OK
Teste Ljung-Box	NOK
MAE	35,3104
RMSE	45,1371

Parâmetros Sazonais	Valor
fourier	3
name	monthly
period	30

Parâmetros	Valor
Padrão	Padrão

Tabela 6: A primeira tabela mostra as métricas e testes estatísticos para o modelo Prophet 5. A segunda e terceira tabelas mostram os parâmetros para a geração do modelo.

8.7 Prophet 6

Métricas	**Valor**
Teste Z	OK
Teste Ljung-Box	NOK
MAE	35,1096
RMSE	41,8507

Parâmetros Sazonais	Valor
Padrão	Padrão

Parâmetros	Valor
Padrão	Padrão

Tabela 7: A primeira tabela mostra as métricas e testes estatísticos para o modelo Prophet 6. A segunda e terceira tabelas mostram os parâmetros para a geração do modelo.

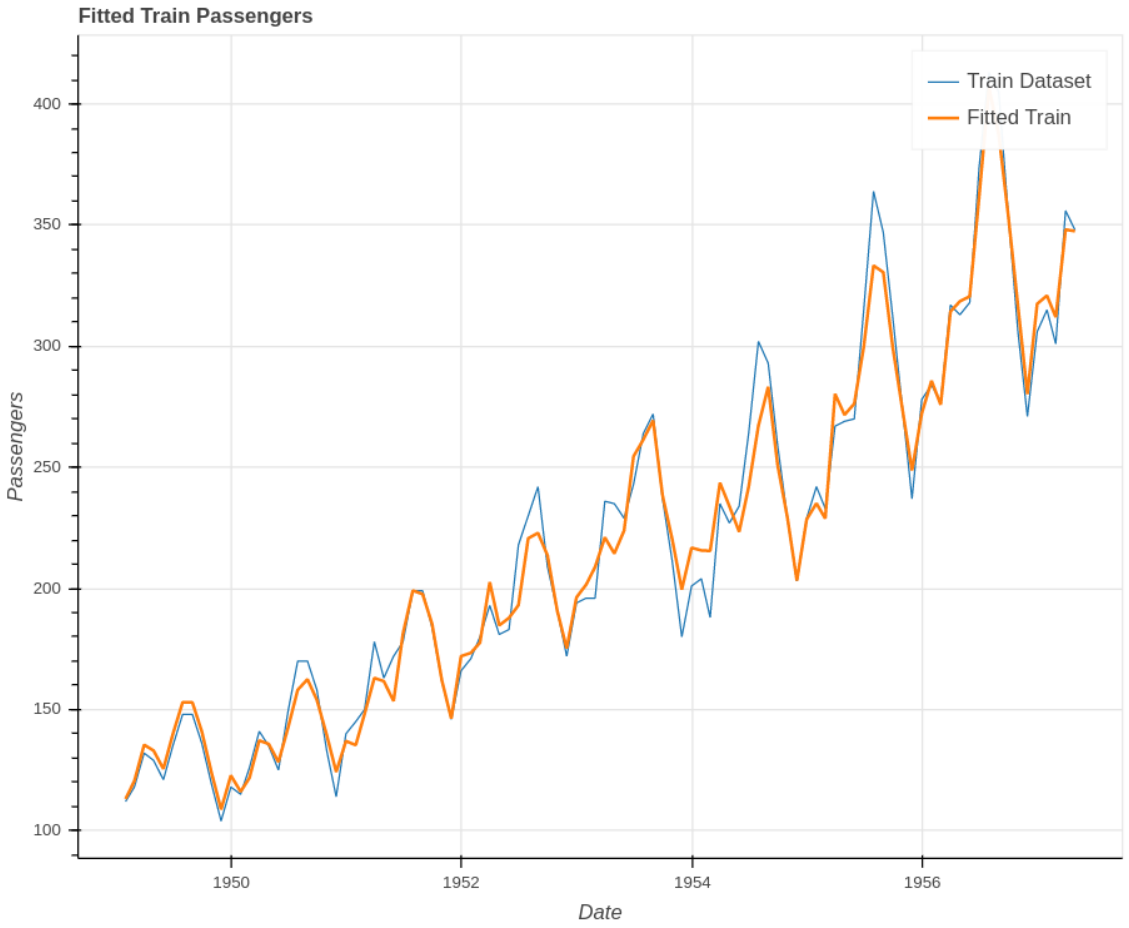

Figura 11: Modelo Holt Winters 2 fitado com o conjunto de treino. Em azul temos os dados de treino original e em laranja o fit do modelo.

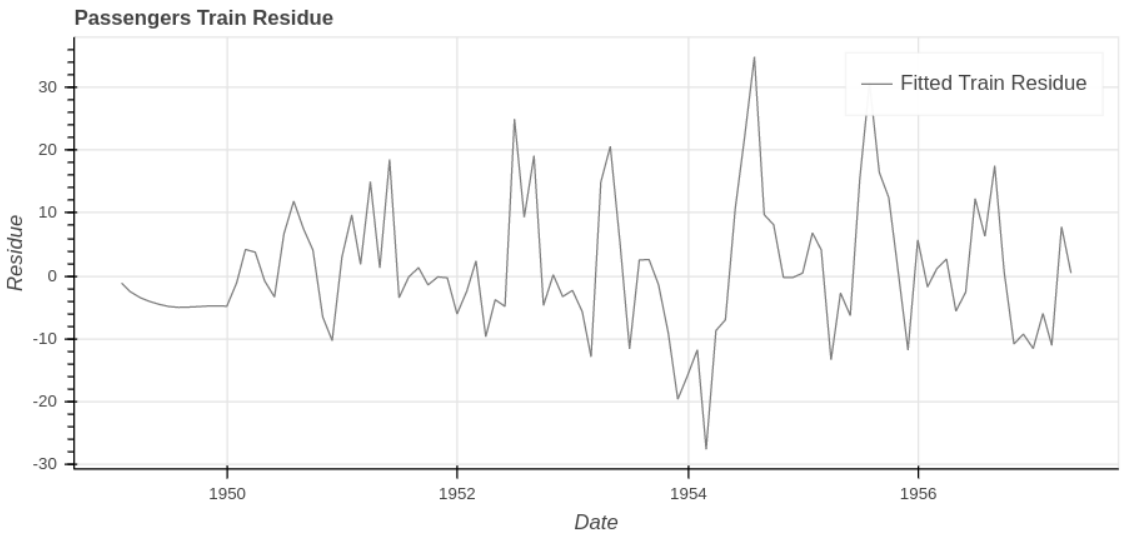

Figura 12: Resíduo do gráfico apresentado pela Figura 11.

8.8 Prophet 7

Métricas	**Valor**
Teste Z	NOK
Teste Ljung-Box	NOK
MAE	25,1206
RMSE	28,9492

Parâmetros Sazonais	Valor
mode	multiplicative
fourier	10
name	daily
period	365

Parâmetros	Valor
Padrão	Padrão

Tabela 8: A primeira tabela mostra as métricas e testes estatísticos para o modelo Prophet 7. A segunda e terceira tabelas mostram os parâmetros para a geração do modelo.

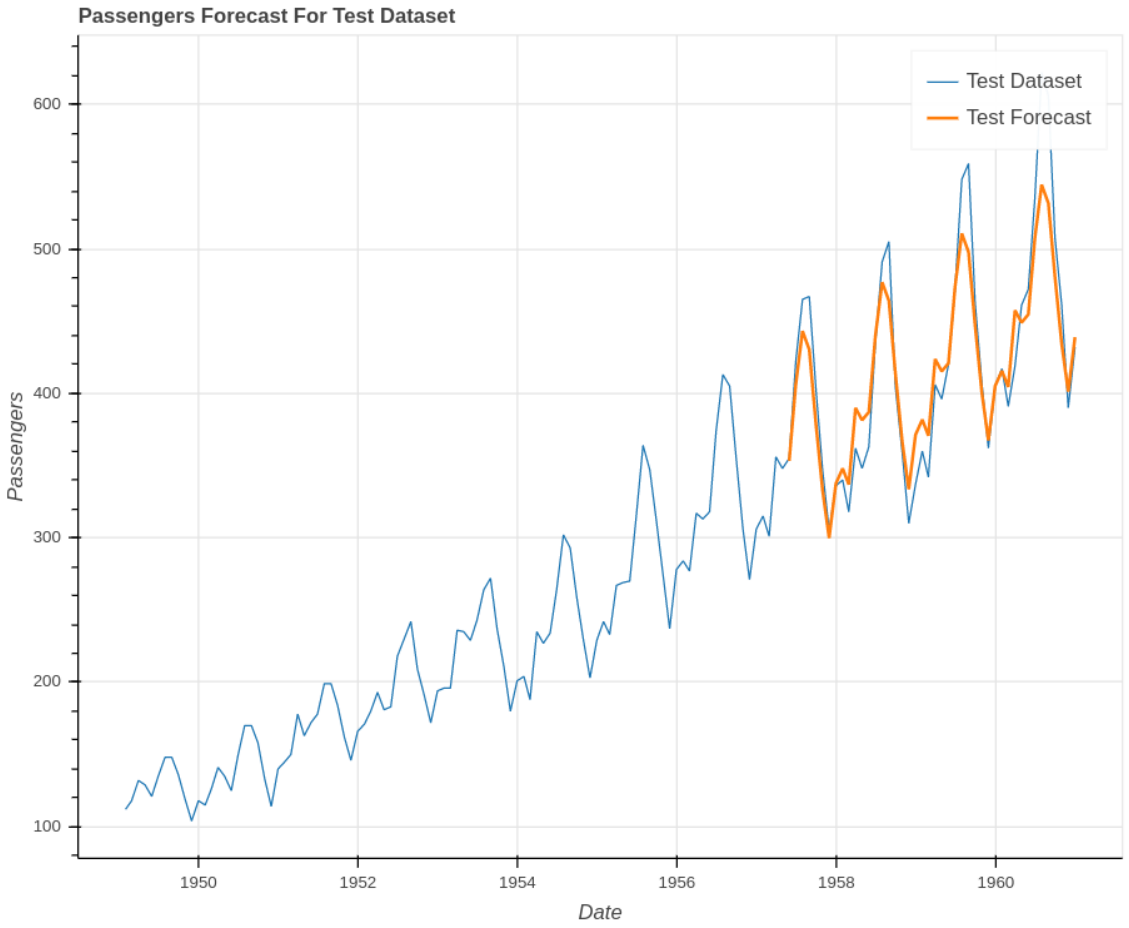

Figura 13: Projeção do modelo Holt Winters 2 sobre o conjunto de teste. Em azul temos o conjunto de dados original e em laranja a projeção do modelo.

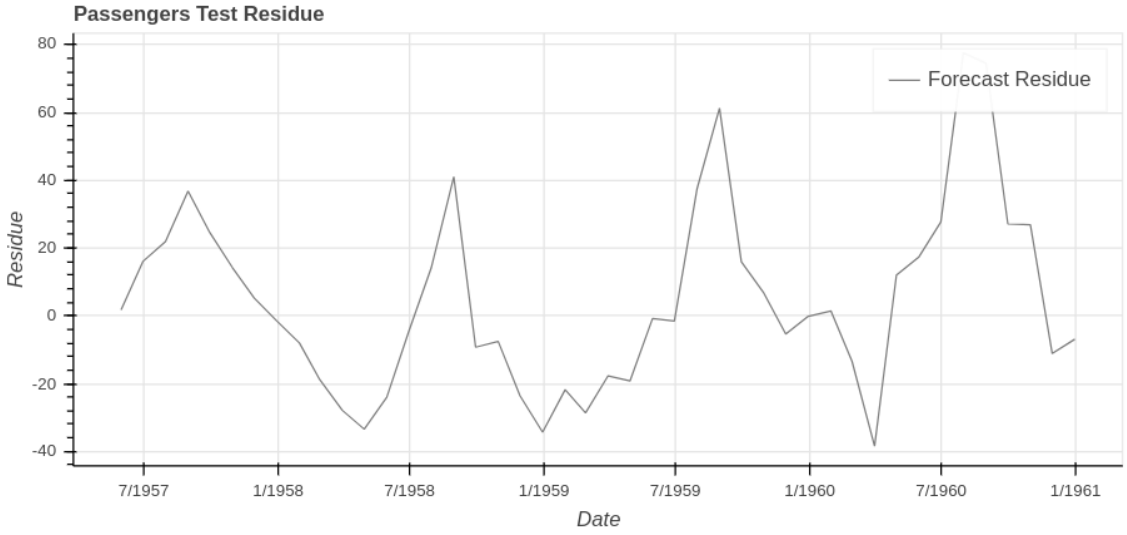

Figura 14: Resíduo do gráfico apresentado pela Figura 13.

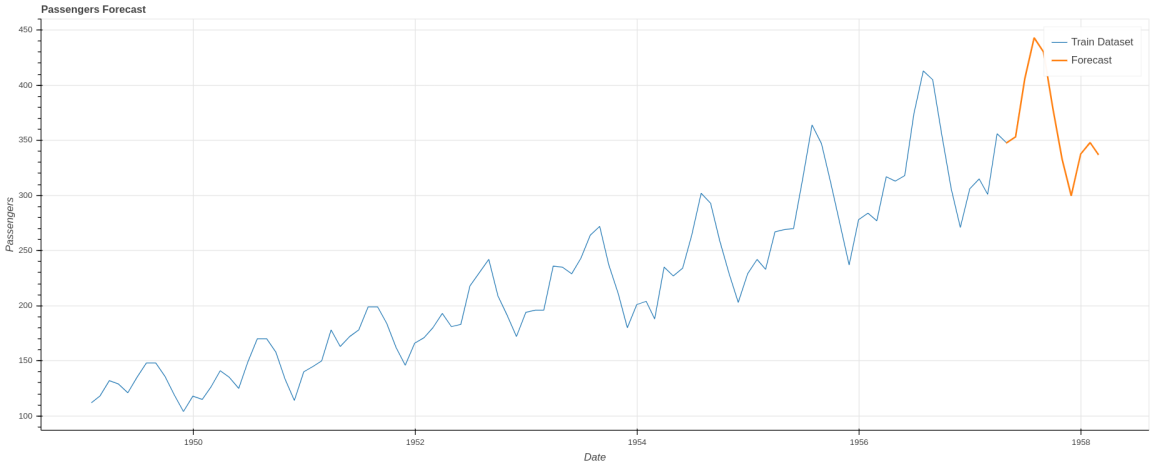

Figura 15: Projeção do modelo Holt-Winters 2 dez meses a frente do último dado fornecido pelo dataset.

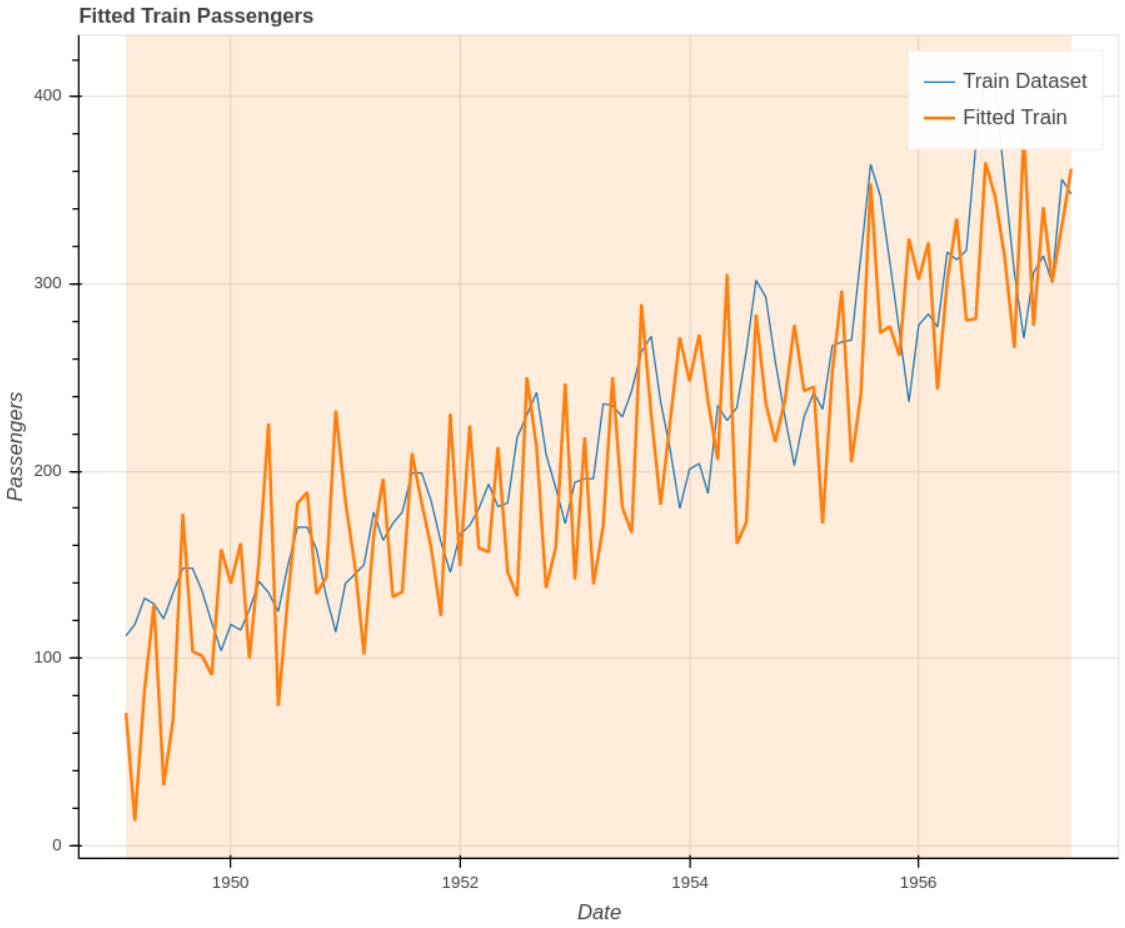

Figura 16: Modelo Prophet 3 fitado com o conjunto de treino. Em azul temos os dados de treino original e em laranja o fit do modelo.

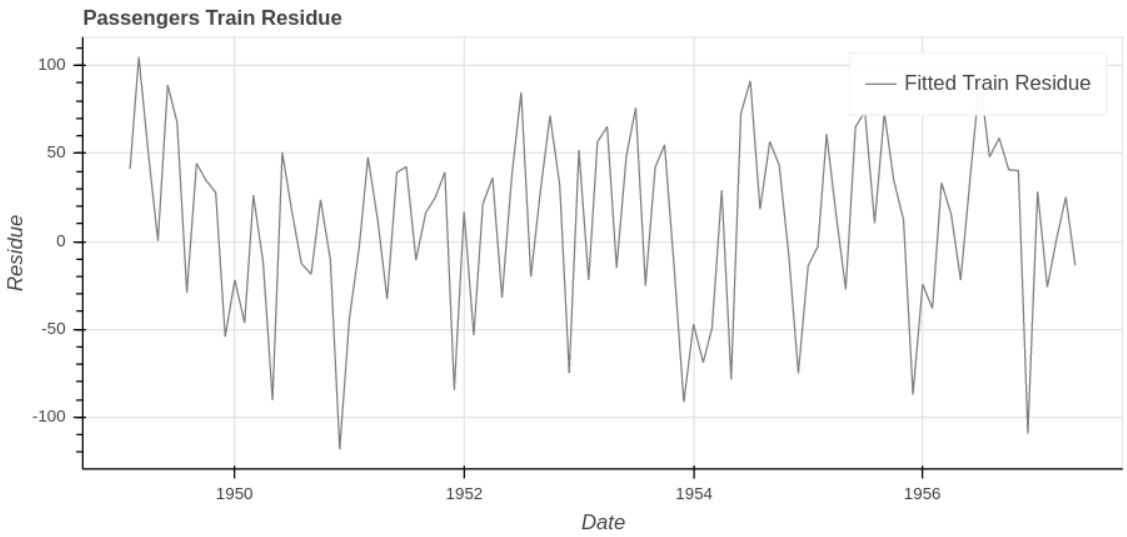

Figura 17: Resíduo do gráfico apresentado pela Figura 16.

8.9 Prophet 8

Métricas	**Valor**
Teste Z	NOK
Teste Ljung-Box	NOK
MAE	26,0113
RMSE	30,2395

Parâmetros Sazonais	Valor
fourier	5
mode	multiplicative
name	daily
period	720

Parâmetros	Valor
Padrão	Padrão

Tabela 9: A primeira tabela mostra as métricas e testes estatísticos para o modelo Prophet 8. A segunda e terceira tabelas mostram os parâmetros para a geração do modelo.

8.10 Arima Sazonal 9

Métricas	**Valor**
Teste Z	NOK
Teste Ljung-Box	NOK
MAE	23,9114
RMSE	28,2132

Parâmetros	Valor
d	1
p	1
q	0

Parâmetros Sazonais	Valor
D	1
P	2
Q	1
s	12

Tabela 10: A primeira tabela mostra as métricas e testes estatísticos para o modelo Arima Sazonal 9. A segunda e terceira tabelas mostram os parâmetros para a geração do modelo.

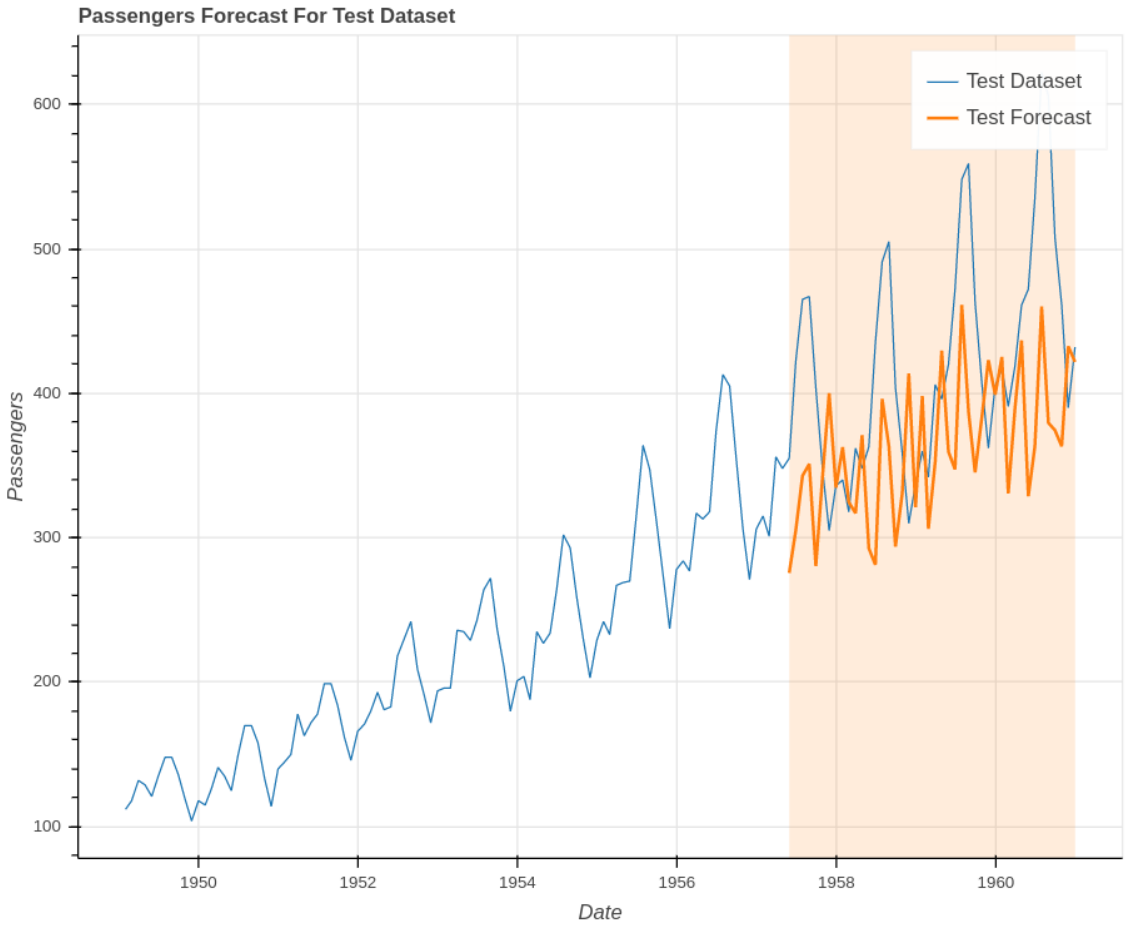

Figura 18: Projeção do modelo Prophet 3 sobre o conjunto de teste. Em azul temos o conjunto de dados original e em laranja a projeção do modelo.

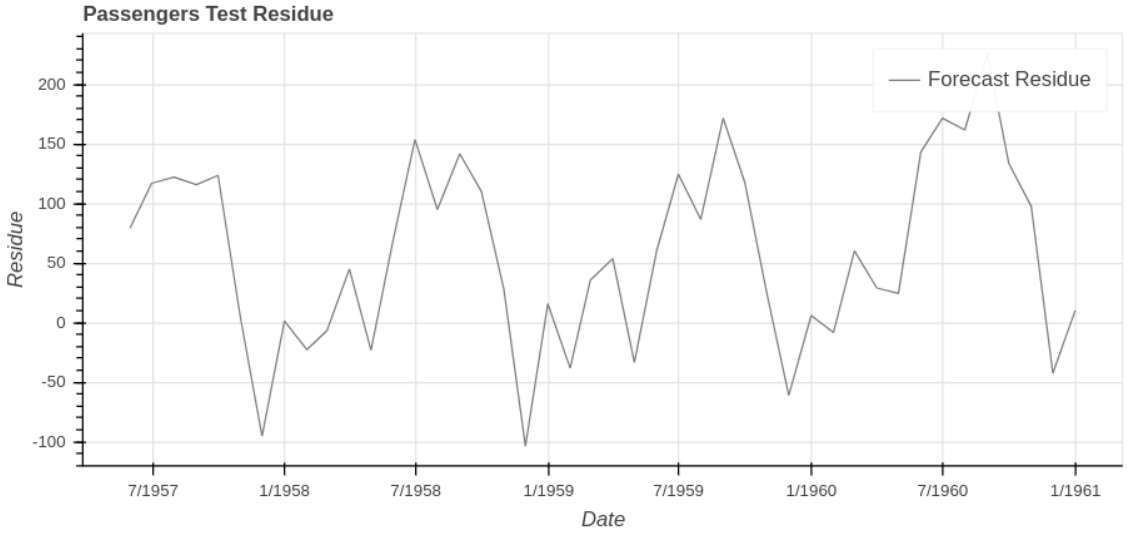

Figura 19: Resíduo do gráfico apresentado pela Figura 18.

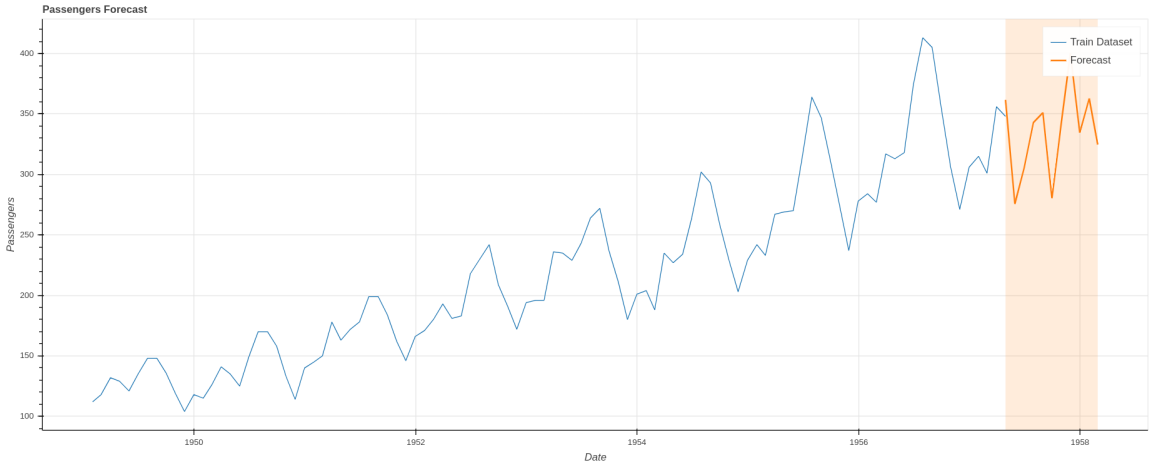

Figura 20: Projeção do modelo Prophet 3 dez meses a frente do último dado fornecido pelo dataset.

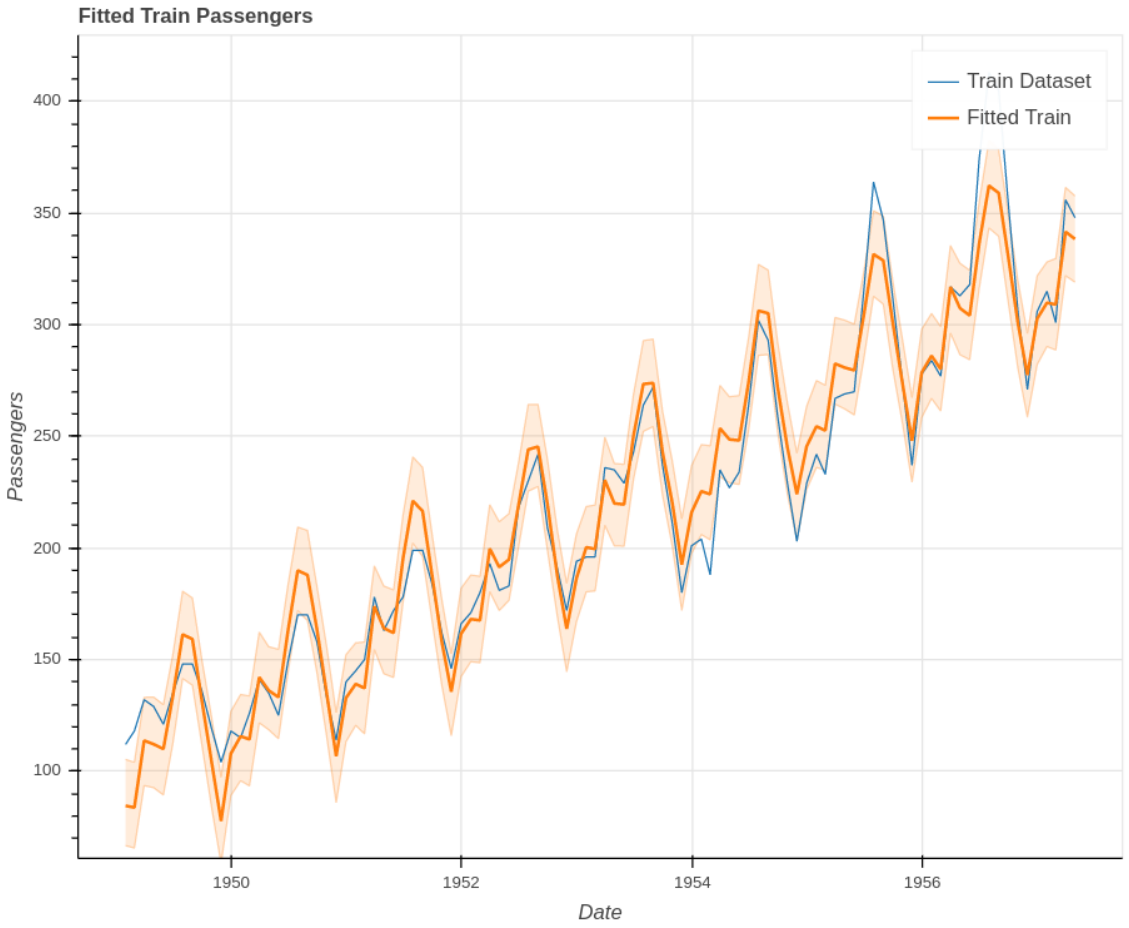

Figura 21: Modelo Prophet 4 fitado com o conjunto de treino. Em azul temos os dados de treino original e em laranja o fit do modelo.

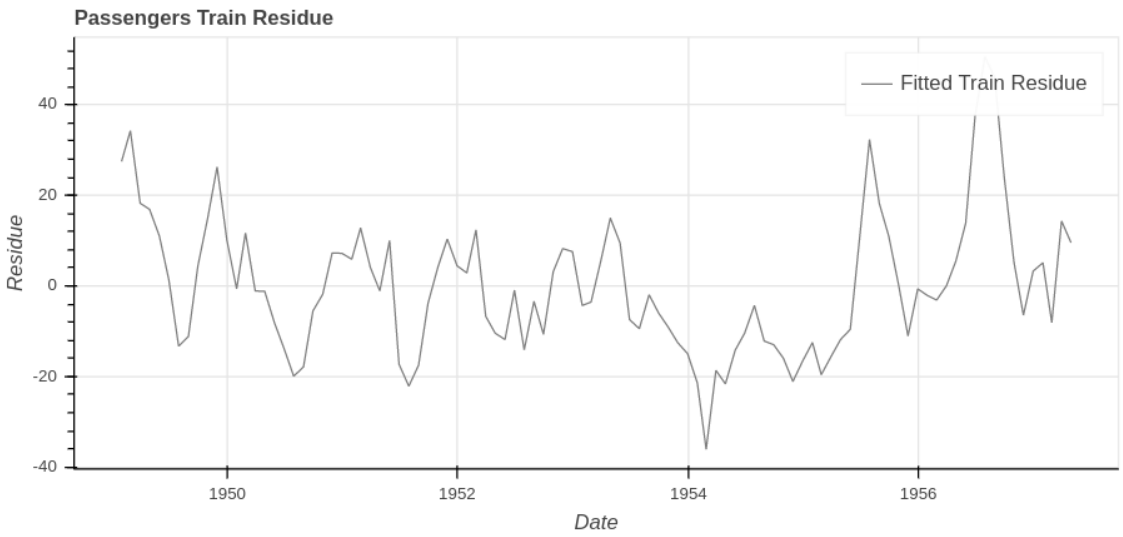

Figura 22: Resíduo do gráfico apresentado pela Figura 21.

9 Conclusões

A aplicação gerada como resultado final deste trabalho pode auxiliar analistas de dados e stakeholders a melhor compreender modelos de machine learning para séries temporais, facilitando a sua geração, o processo de tunelamento de parâmetros e seleção automática entre diversos modelos que podem ser requeridos. Conforme mostra a seção 8, o esquema de visualização dos modelos é simples e auxilia na comparação de modelos pelo usuário.

A maneira que o código foi implementado também facilita possíveis expansões da aplicação, com adições de novas métricas de erro, testes estatísticos, além de outros algoritmos de modelagem sem que o código já escrito seja alterado. Entretanto existem alguns pontos onde o trabalho poderia ser melhorado, especialmente no que se diz respeito ao tratamento de dados. Seria possível estender o trabalho e incorporar conceitos de big data, utilizando algumas ferramentas do ecossistema do *Hadoop*, como HDFS e Spark por exemplo, para aumentar o volume de dados processados, tornando o projeto muito mais escalável.

Referências

[1] A. Harvey and S. Peters, *Estimation procedures for structural time series models*, Journal of Forecasting (9), 89–108 (1990).

[2] A. C. Harvey and N. Shephard , *Structural time series models.* Handbook of Statistics, **11**, Elsevier, (10), 261–302 (1993).

[3] H. Wickham, *Elegant graphics for data analysis.* (2nd ed). Springer (2016).

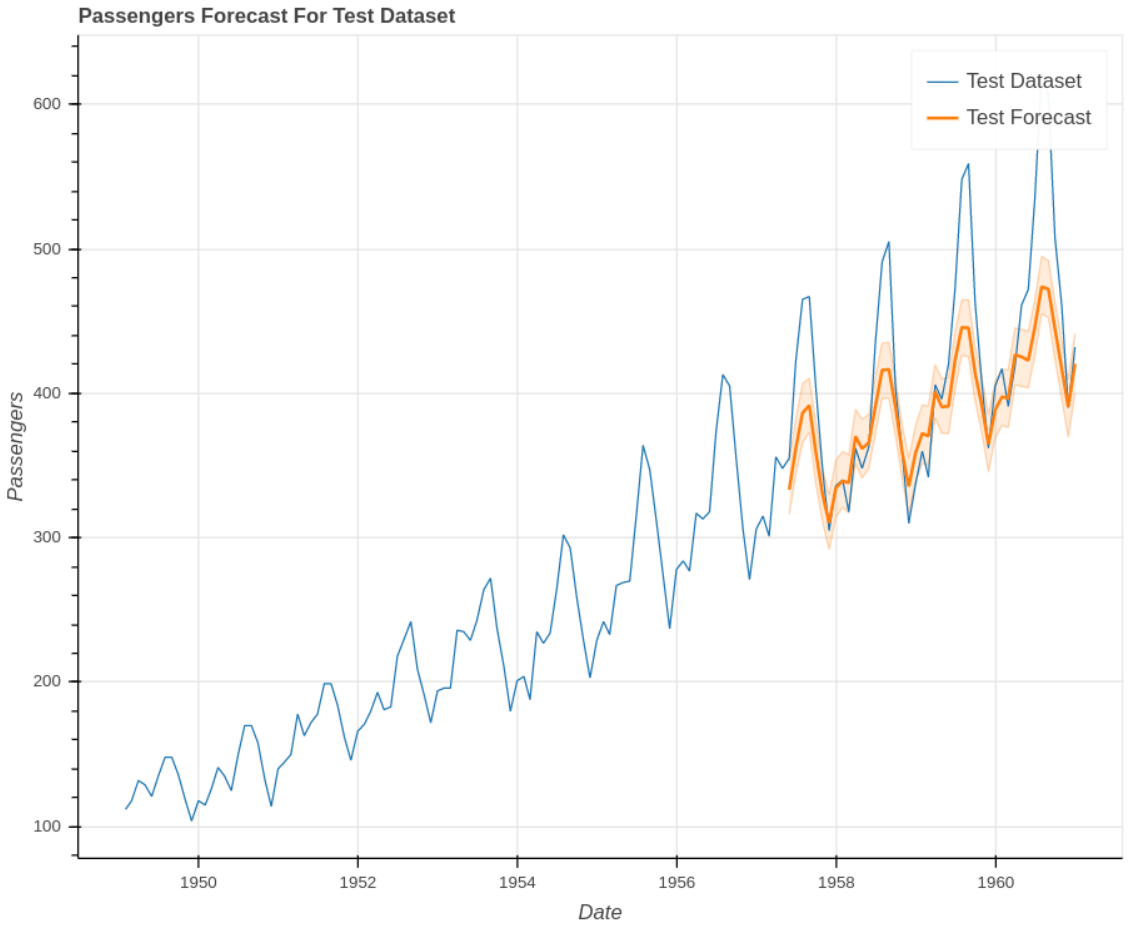

Figura 23: Projeção do modelo Prophet 4 sobre o conjunto de teste. Em azul temos o conjunto de dados original e em laranja a projeção do modelo.

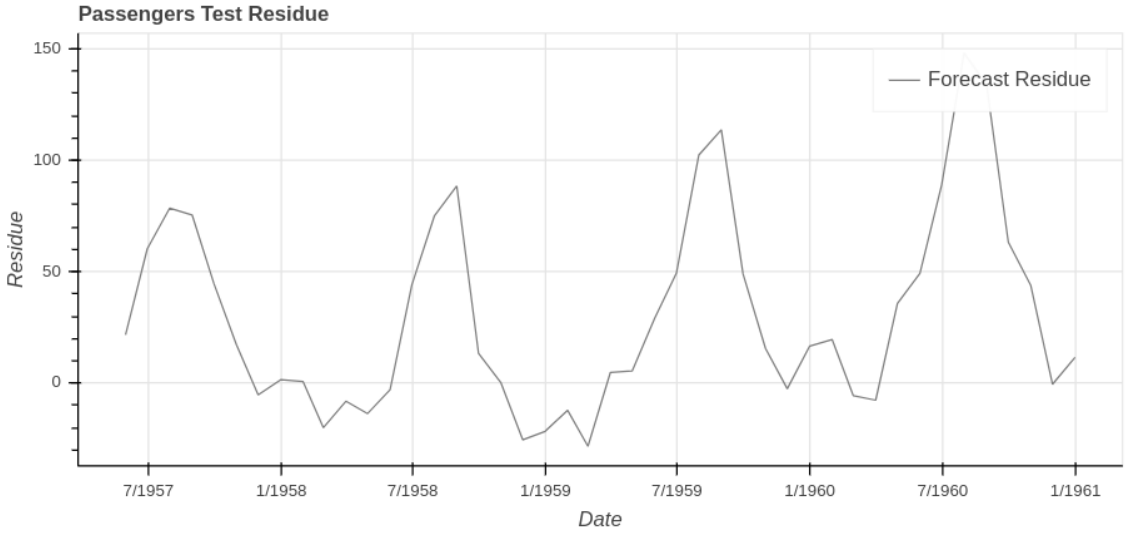

Figura 24: Resíduo do gráfico apresentado pela Figura 23.

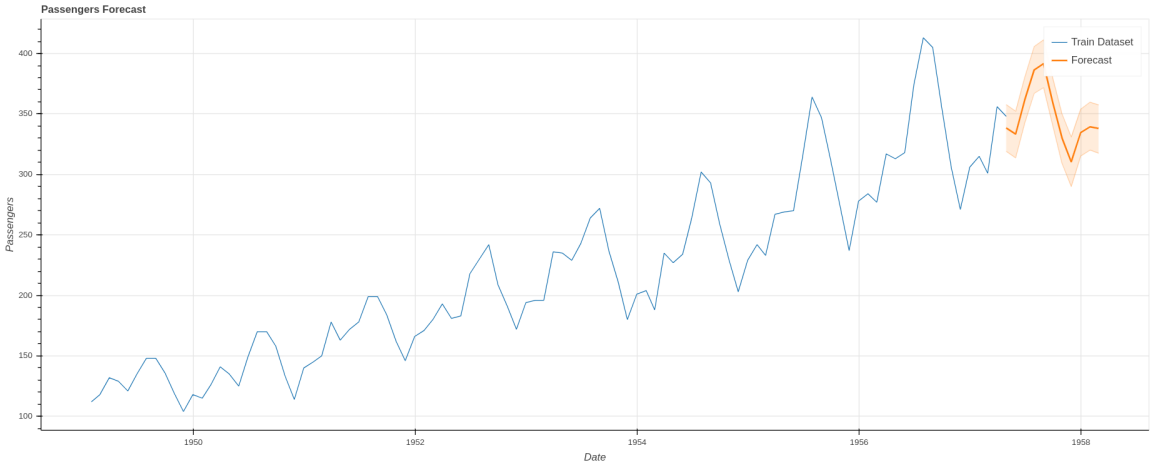

Figura 25: Projeção do modelo Prophet 4 dez meses a frente do último dado fornecido pelo dataset.

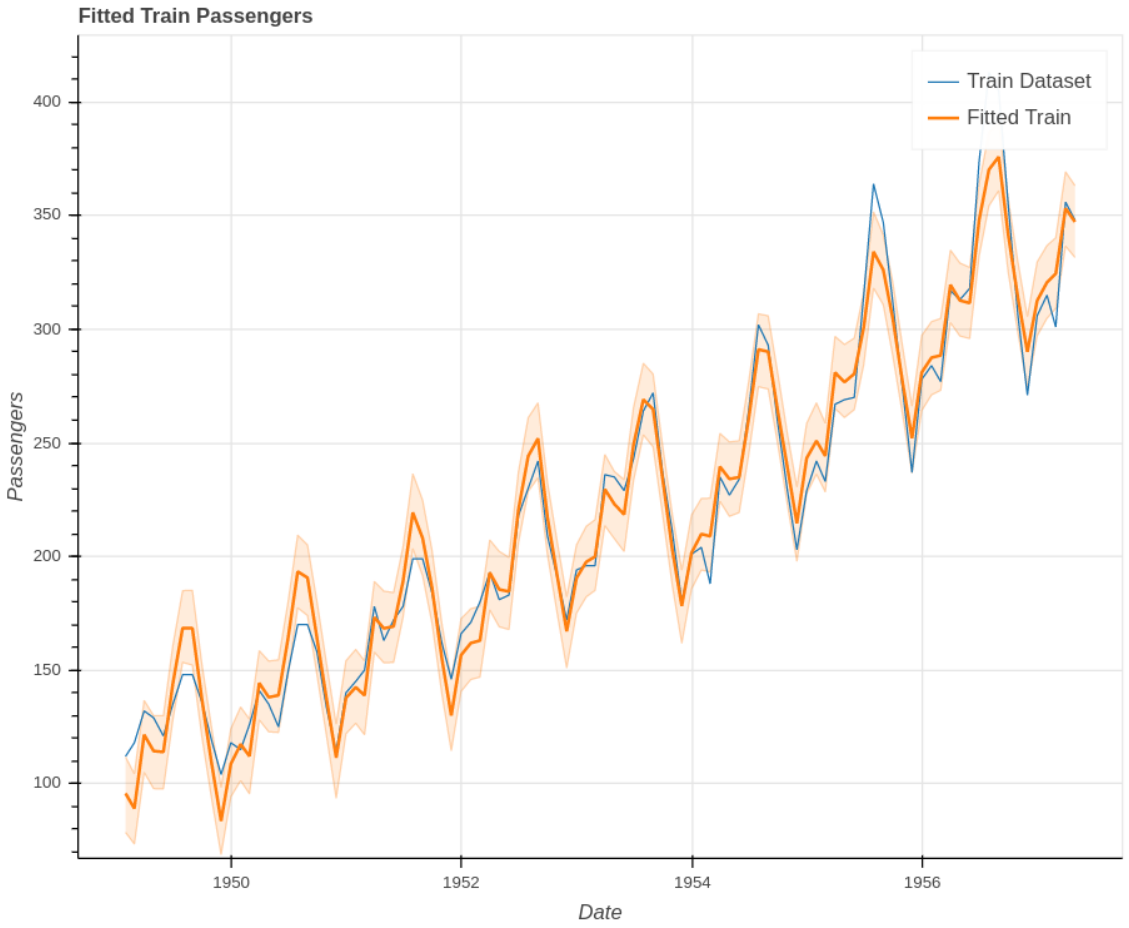

Figura 26: Modelo Prophet 5 fitado com o conjunto de treino. Em azul temos os dados de treino original e em laranja o fit do modelo.

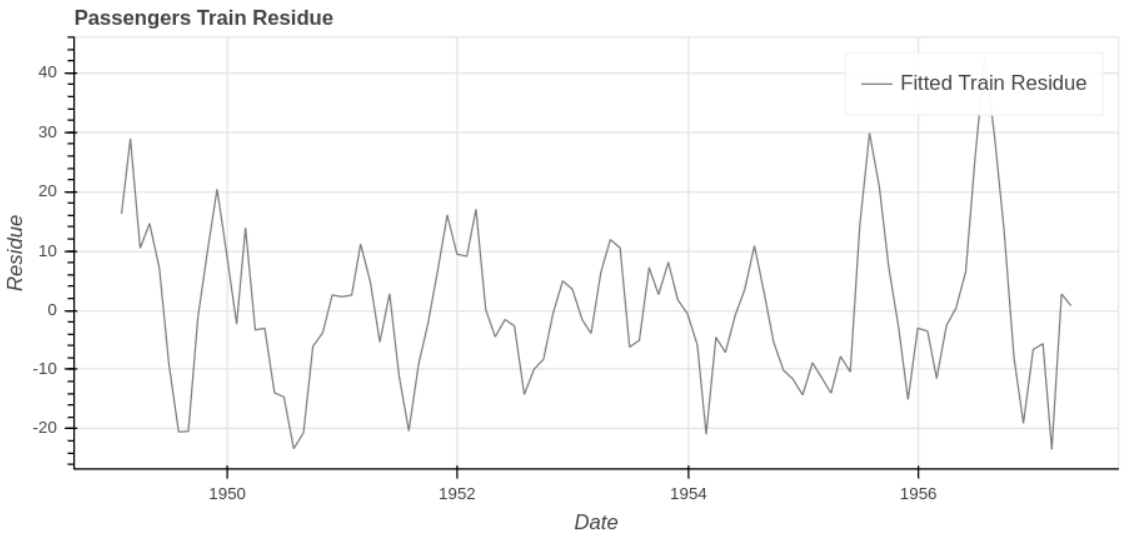

Figura 27: Resíduo do gráfico apresentado pela Figura 26.

[4] https://machinelearningmastery.com/time-series-data-stationary-python/, (Acesso em 05/12/2019)

[5] https://docs.bokeh.org/en/latest/docs/dev_guide/documentation.html , (Acesso em 05/12/2019)

[6] https://www.itl.nist.gov/div898/handbook/prc/section1/prc13.htm , (Acesso em 05/12/2019)

[7] https://peerj.com/preprints/3190.pdf , (Acesso em 05/12/2019)

[8] https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html , (Acesso em 05/12/2019)

[9] https://www.statsmodels.org/dev/generated/statsmodels.tsa.holtwinters.ExponentialSmoothing.html , (Acesso em 05/12/2019)

[10] https://www.statisticshowto.datasciencecentral.com/z-test/ , (Acesso em 05/12/2019)

[11] https://www.statisticshowto.datasciencecentral.com/ljung-box-test/ , (Acesso em 05/12/2019)

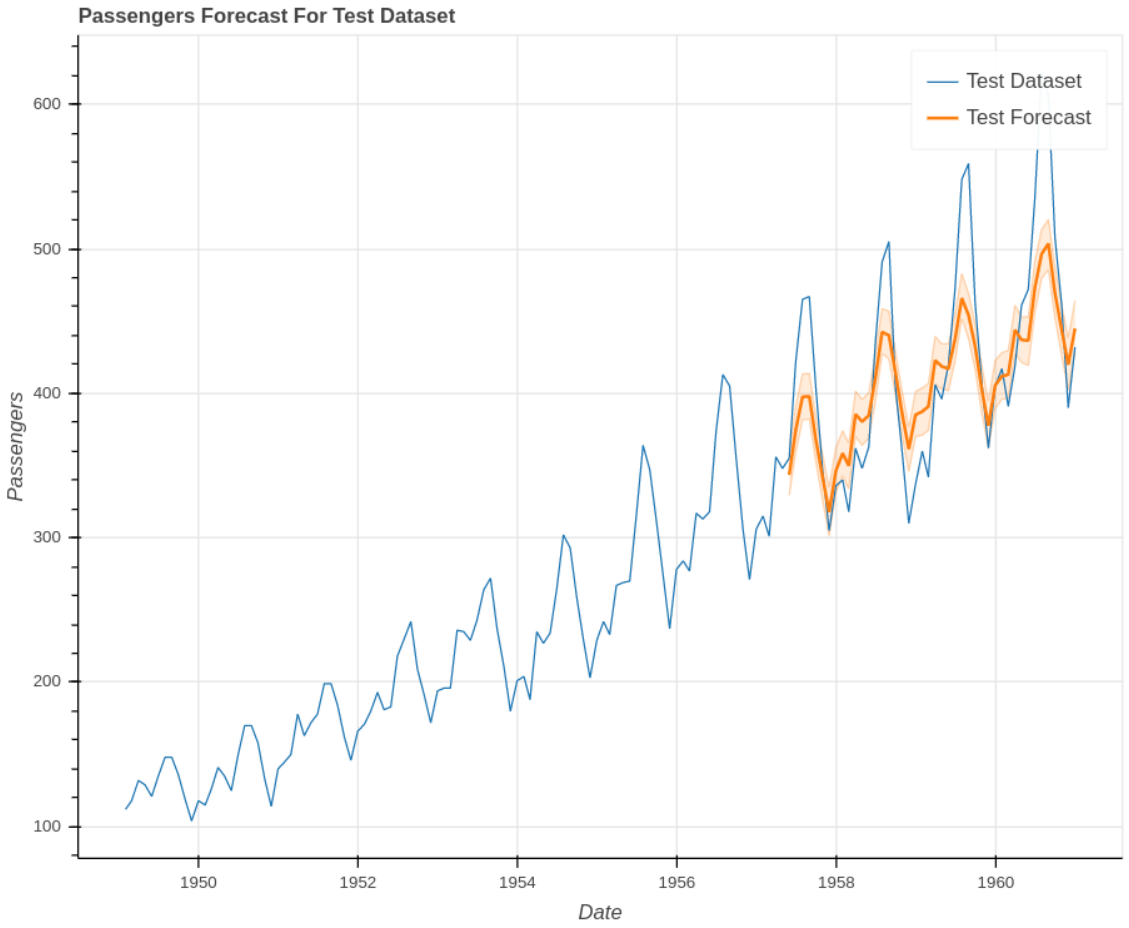

Figura 28: Projeção do modelo Prophet 5 sobre o conjunto de teste. Em azul temos o conjunto de dados original e em laranja a projeção do modelo.

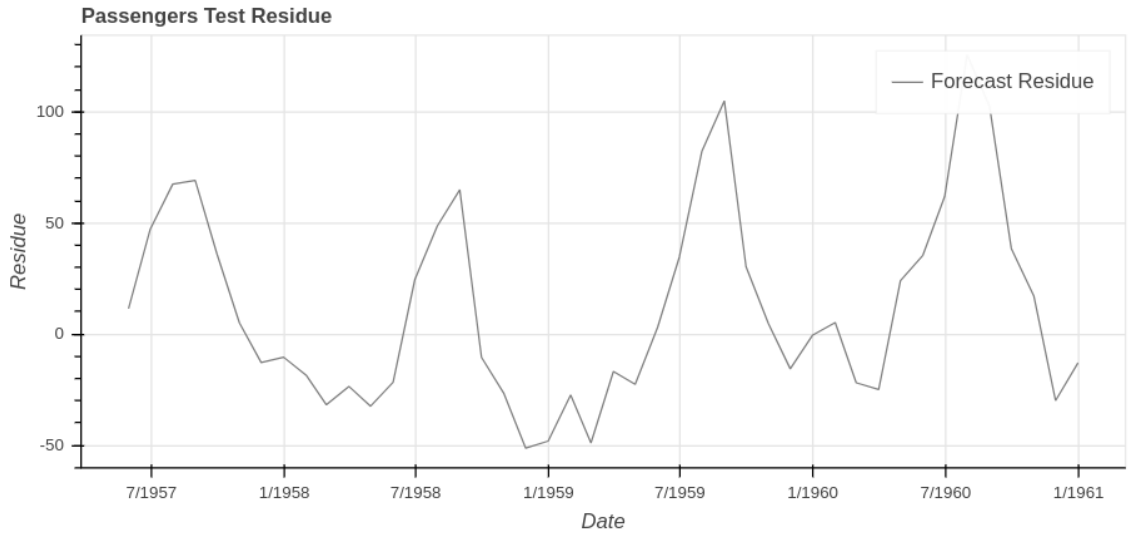

Figura 29: Resíduo do gráfico apresentado pela Figura 28.

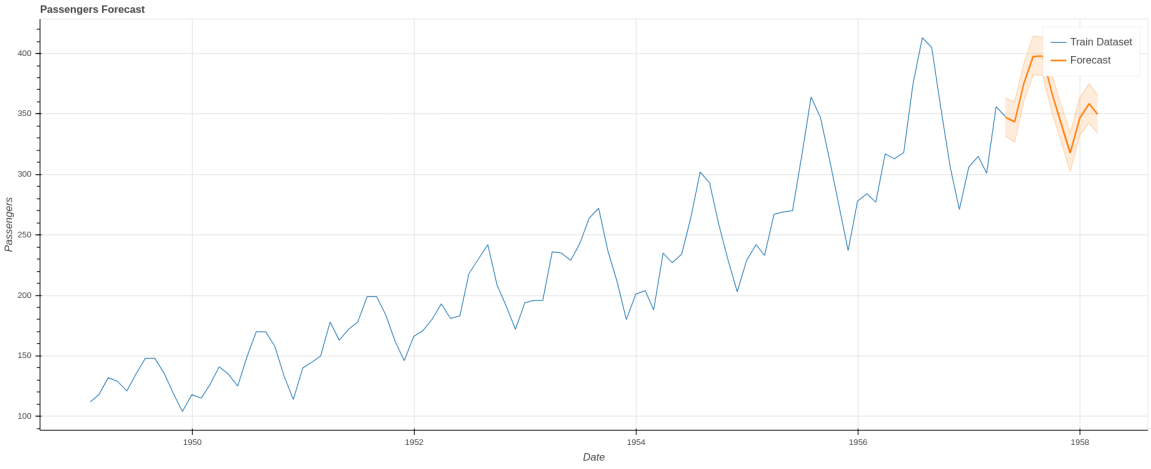

Figura 30: Projeção do modelo Prophet 5 dez meses a frente do último dado fornecido pelo dataset.

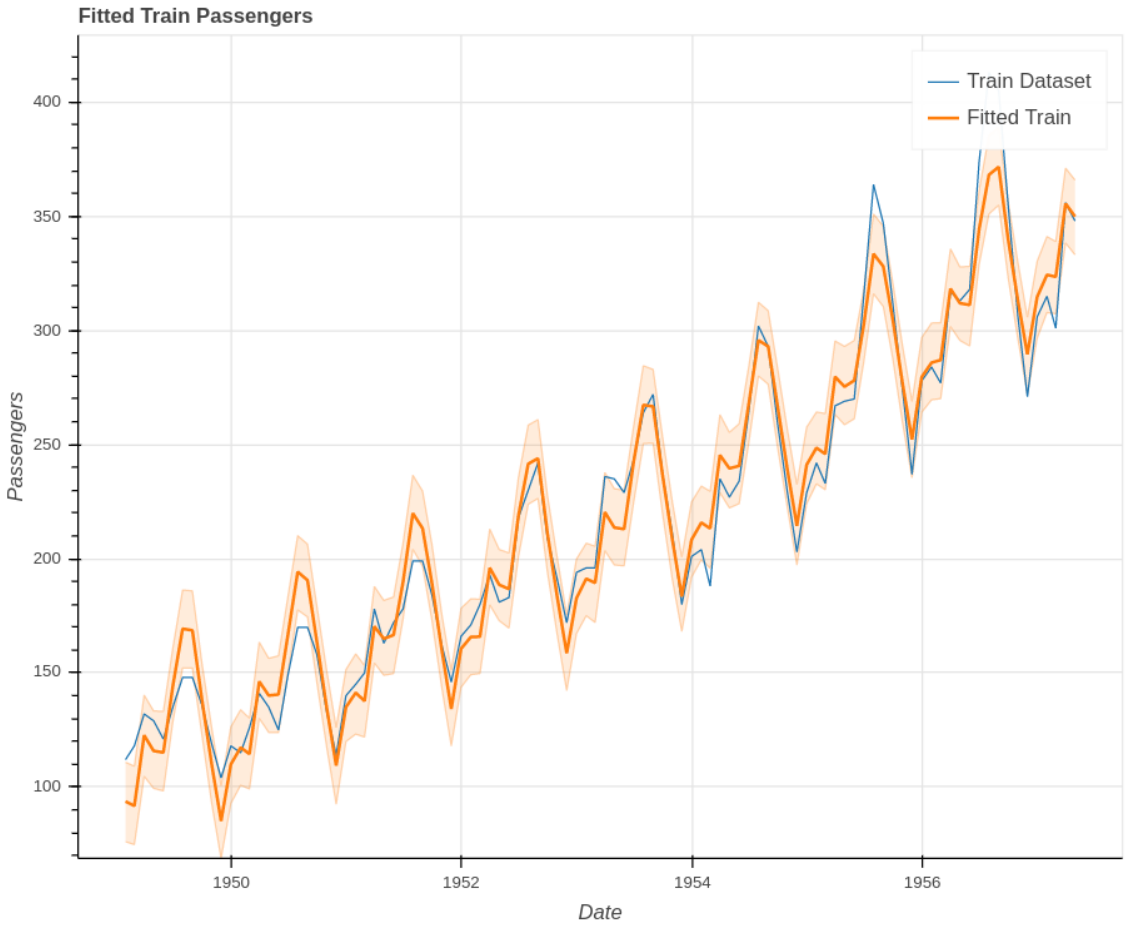

Figura 31: Modelo Prophet 6 fitado com o conjunto de treino. Em azul temos os dados de treino original e em laranja o fit do modelo.

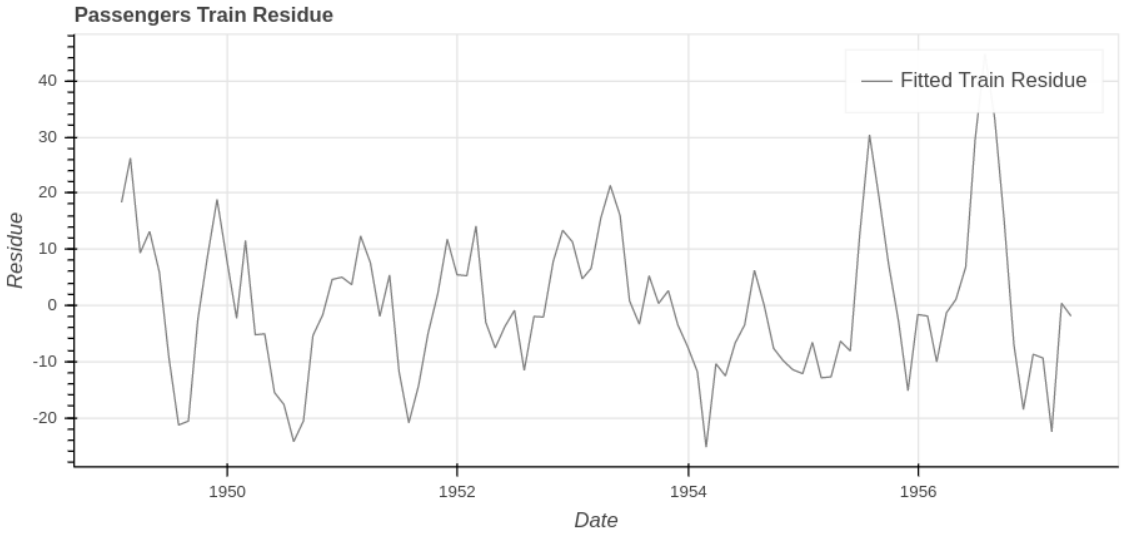

Figura 32: Resíduo do gráfico apresentado pela Figura 31.

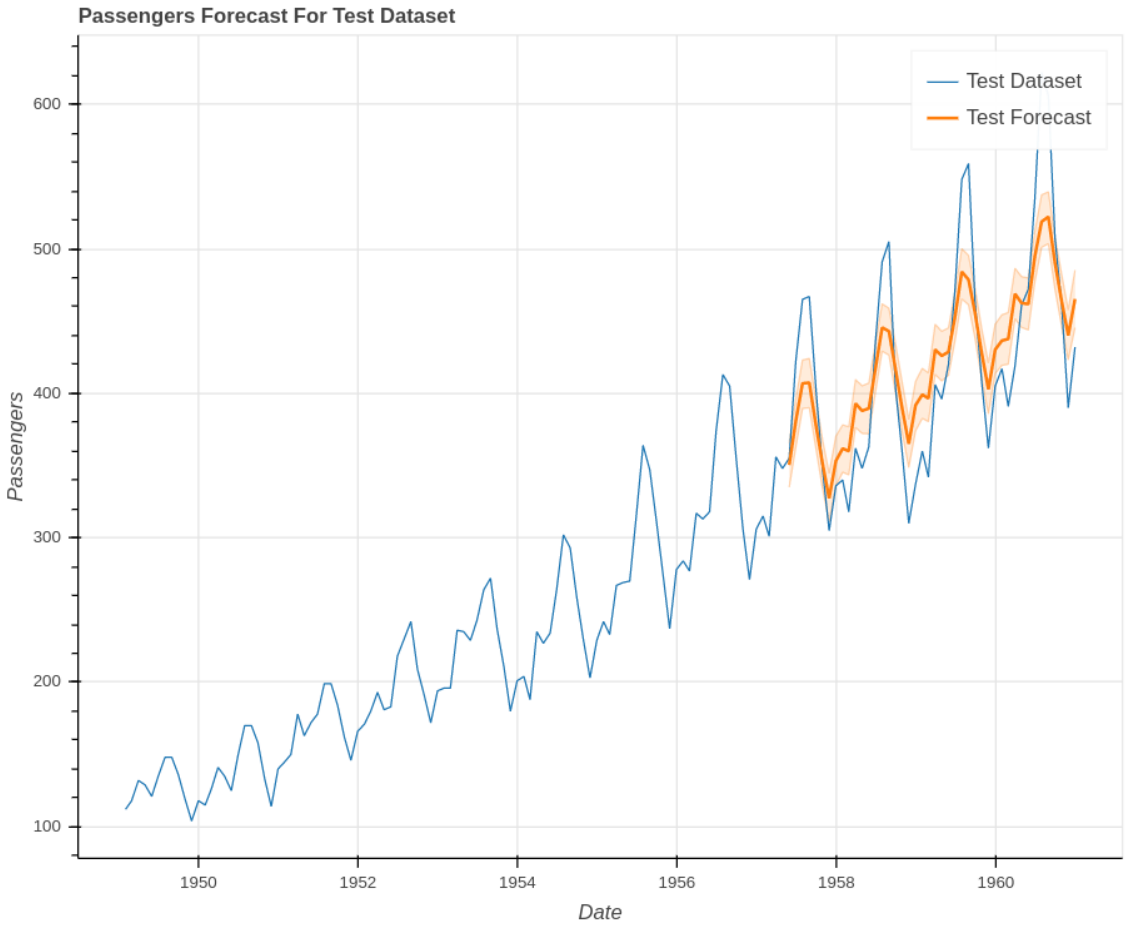

Figura 33: Projeção do modelo Prophet 6 sobre o conjunto de teste. Em azul temos o conjunto de dados original e em laranja a projeção do modelo.

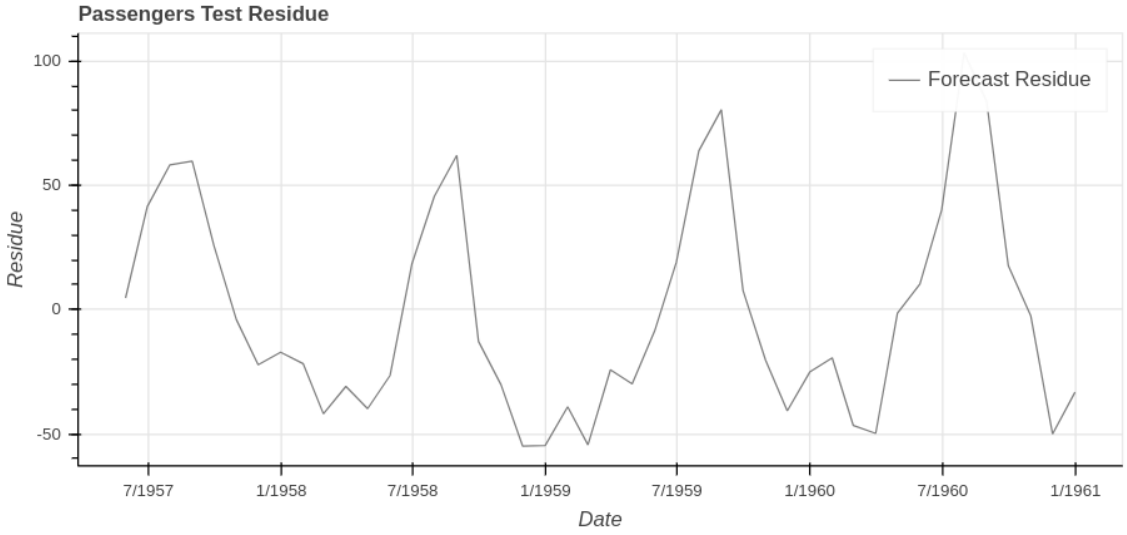

Figura 34: Resíduo do gráfico apresentado pela Figura 33.

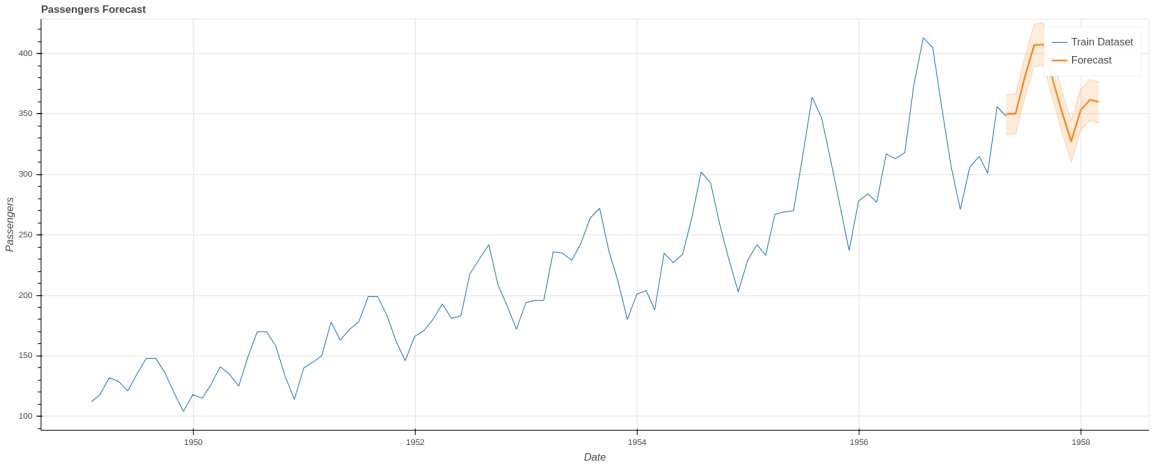

Figura 35: Projeção do modelo Prophet 6 dez meses a frente do último dado fornecido pelo dataset.

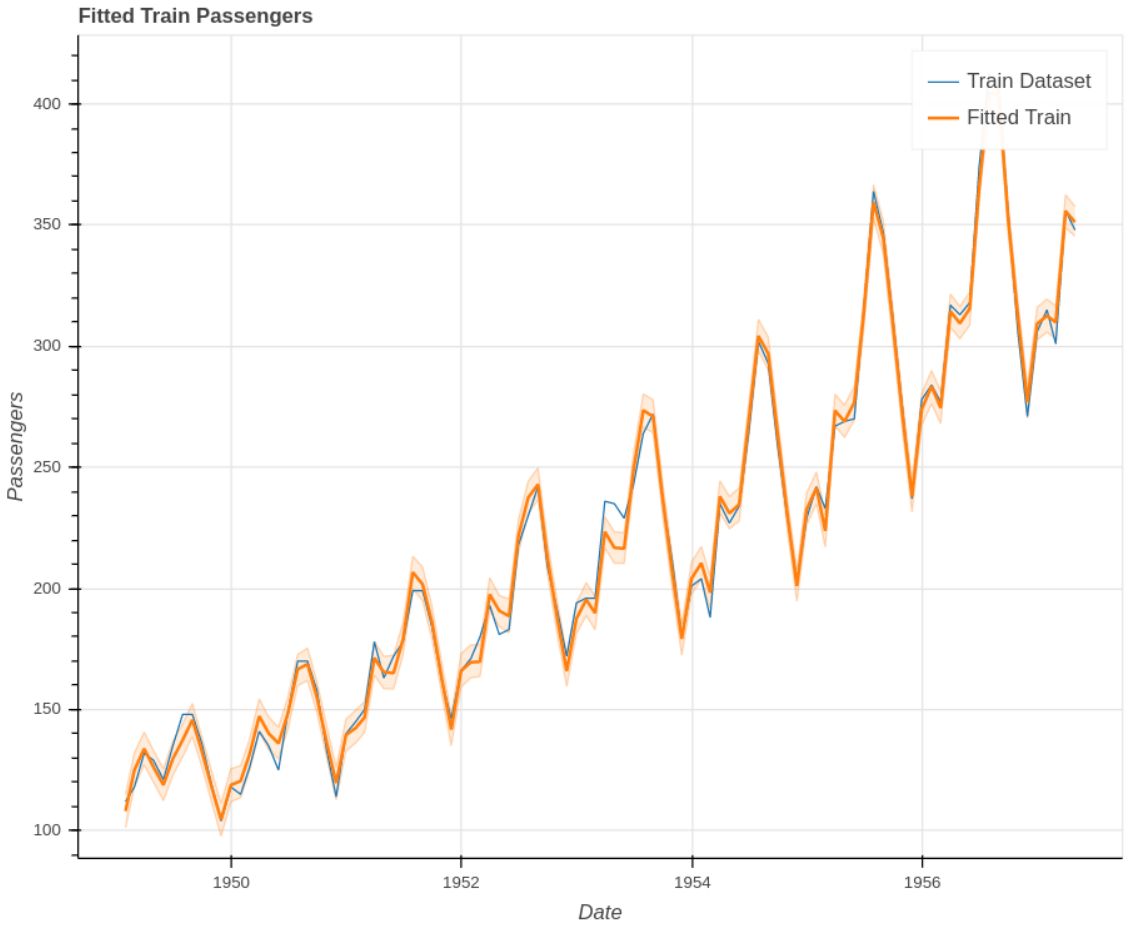

Figura 36: Modelo Prophet 7 fitado com o conjunto de treino. Em azul temos os dados de treino original e em laranja o fit do modelo.

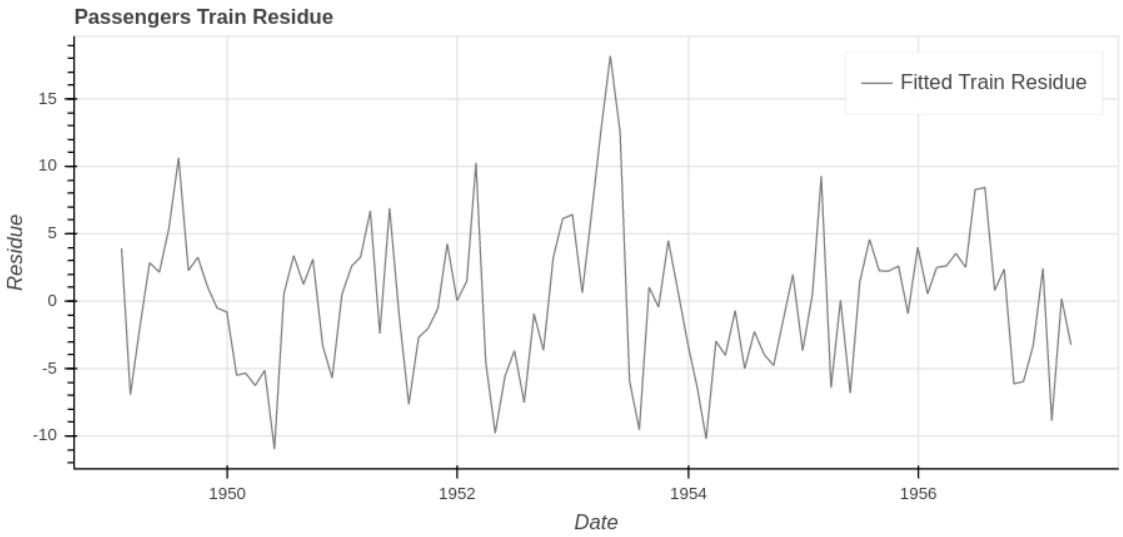

Figura 37: Resíduo do gráfico apresentado pela Figura 36.

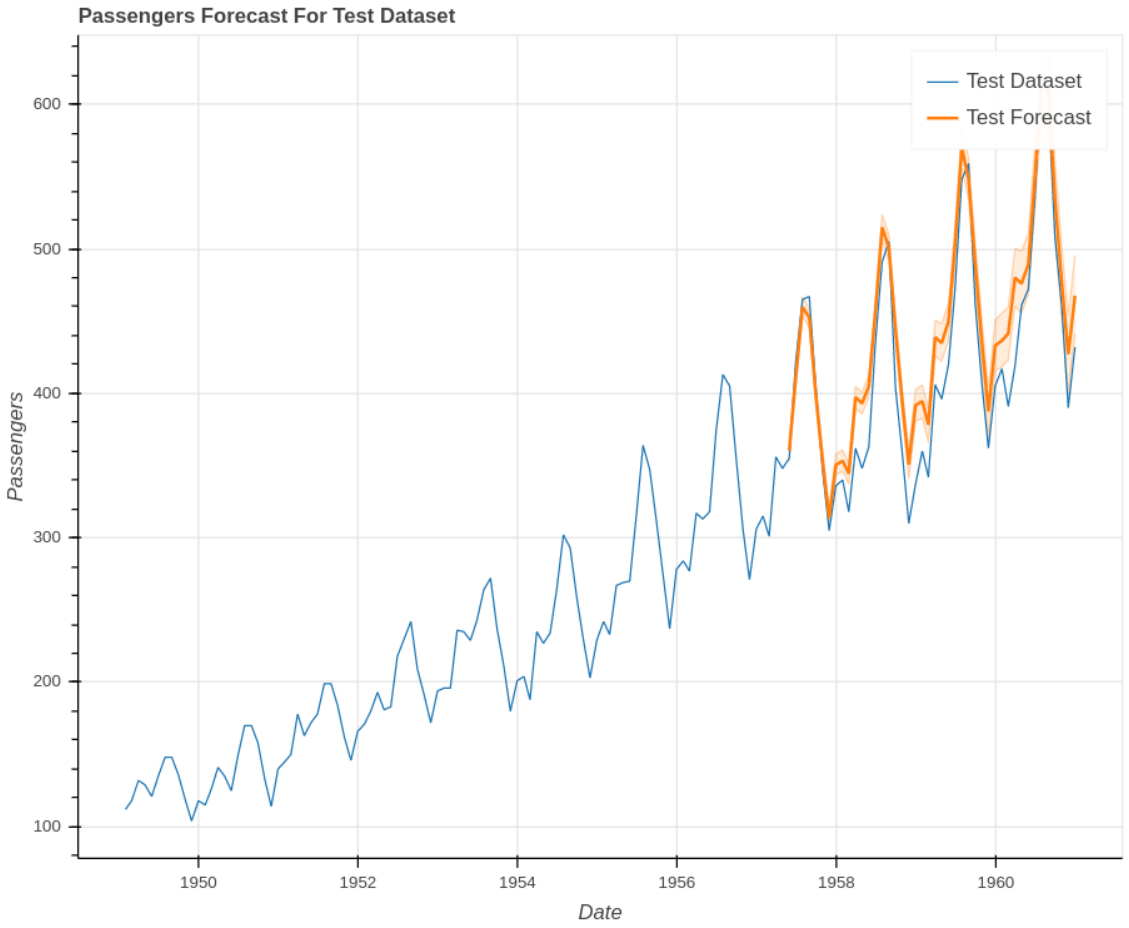

Figura 38: Projeção do modelo Prophet 7 sobre o conjunto de teste. Em azul temos o conjunto de dados original e em laranja a projeção do modelo.

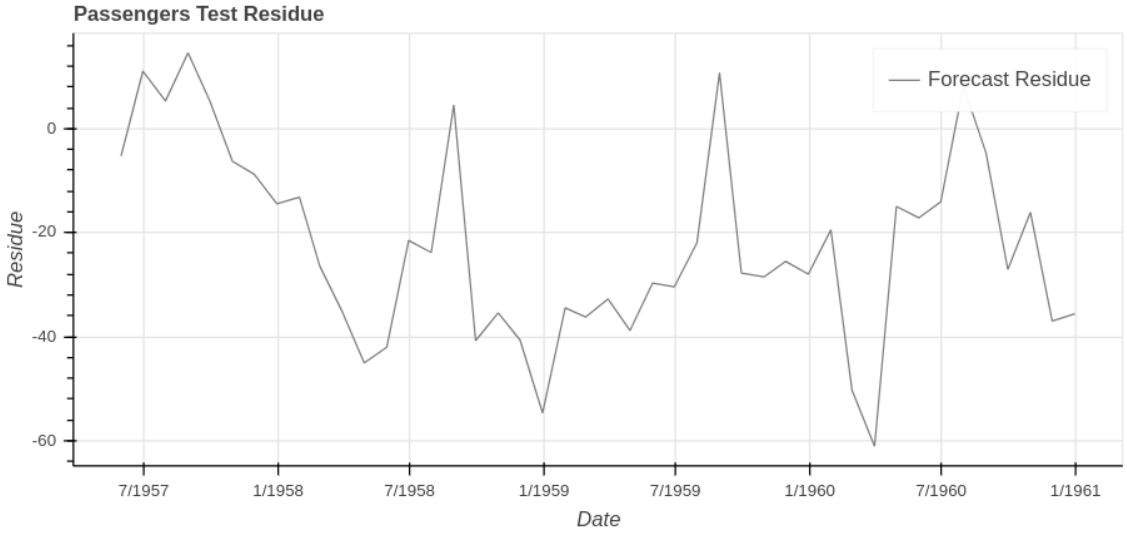

Figura 39: Resíduo do gráfico apresentado pela Figura 38.

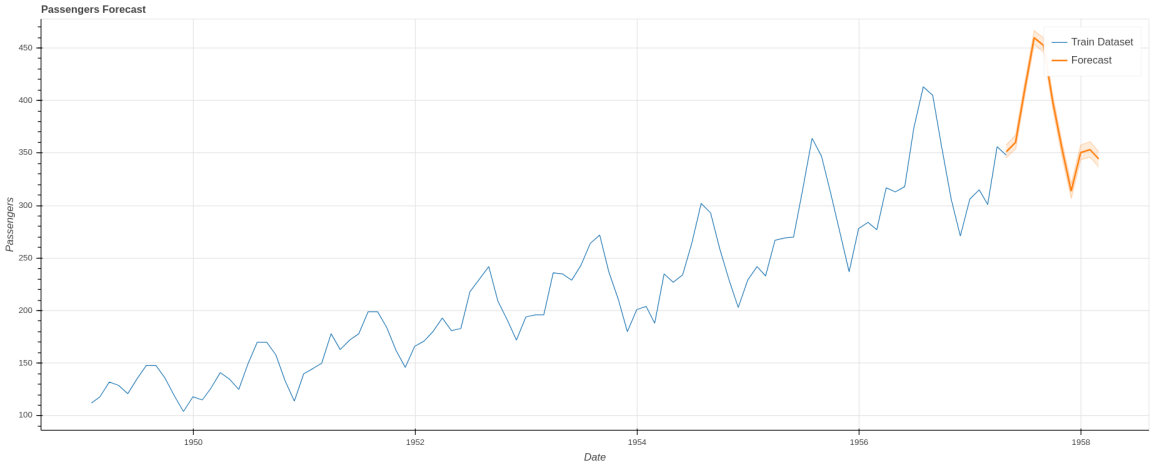

Figura 40: Projeção do modelo Prophet 7 dez meses a frente do último dado fornecido pelo dataset.

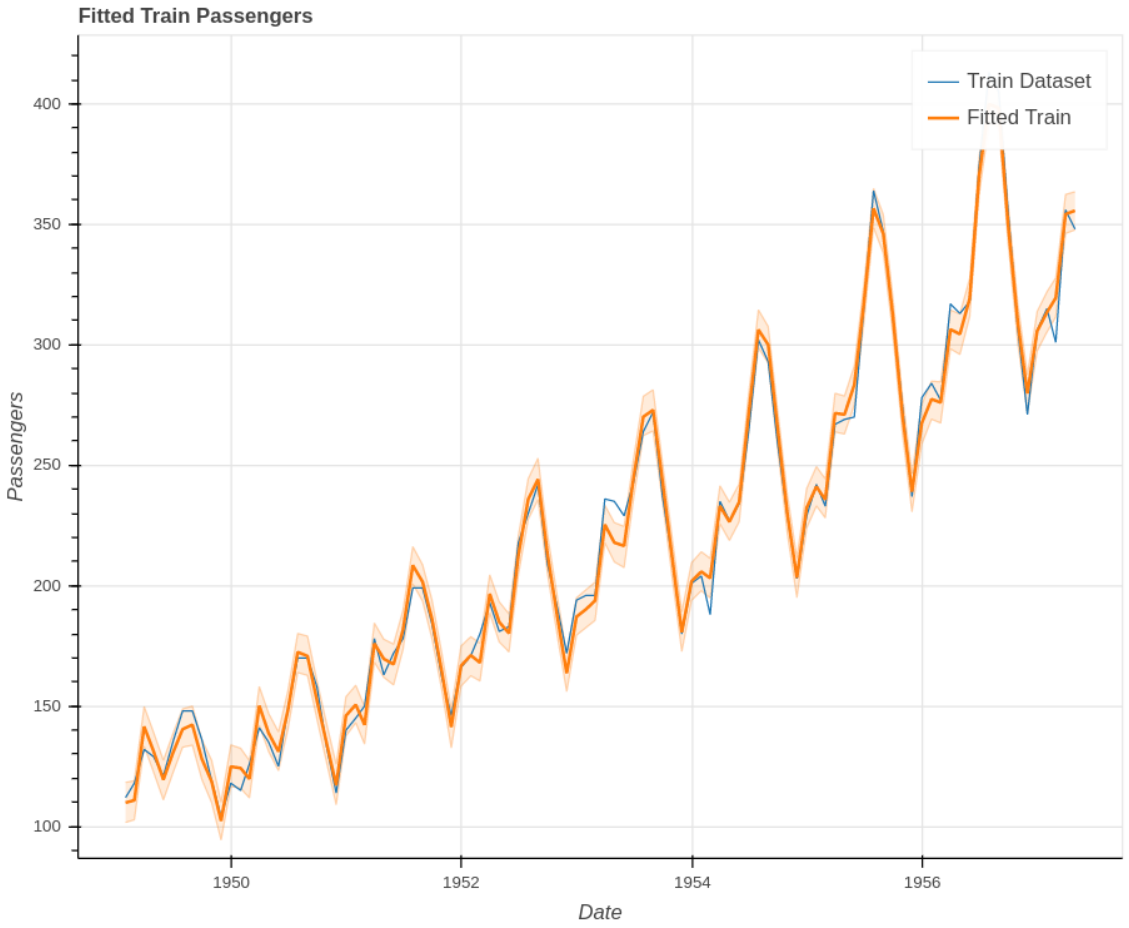

Figura 41: Modelo Prophet 8 fitado com o conjunto de treino. Em azul temos os dados de treino original e em laranja o fit do modelo.

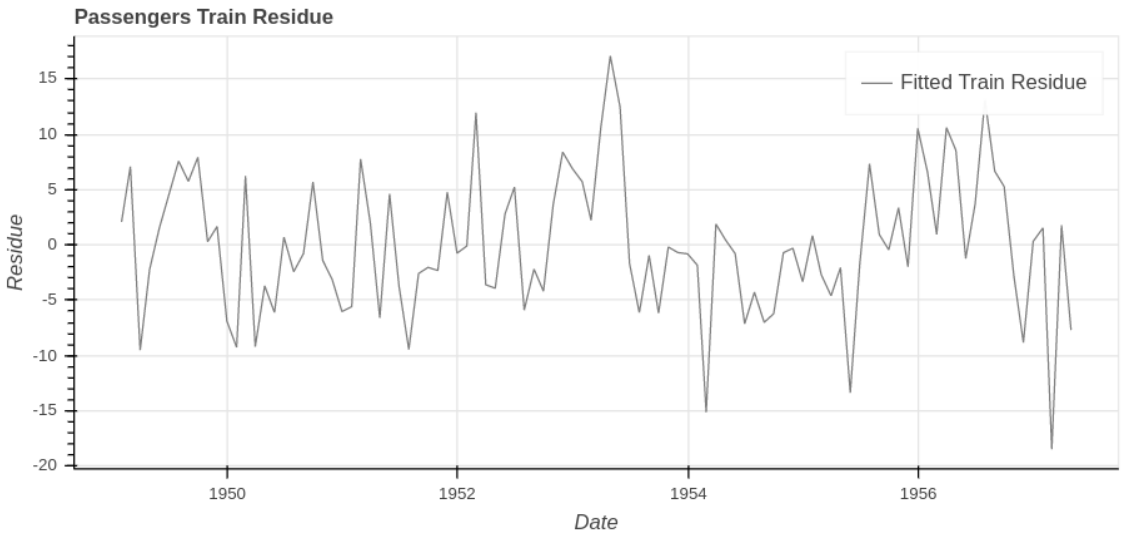

Figura 42: Resíduo do gráfico apresentado pela Figura 41.

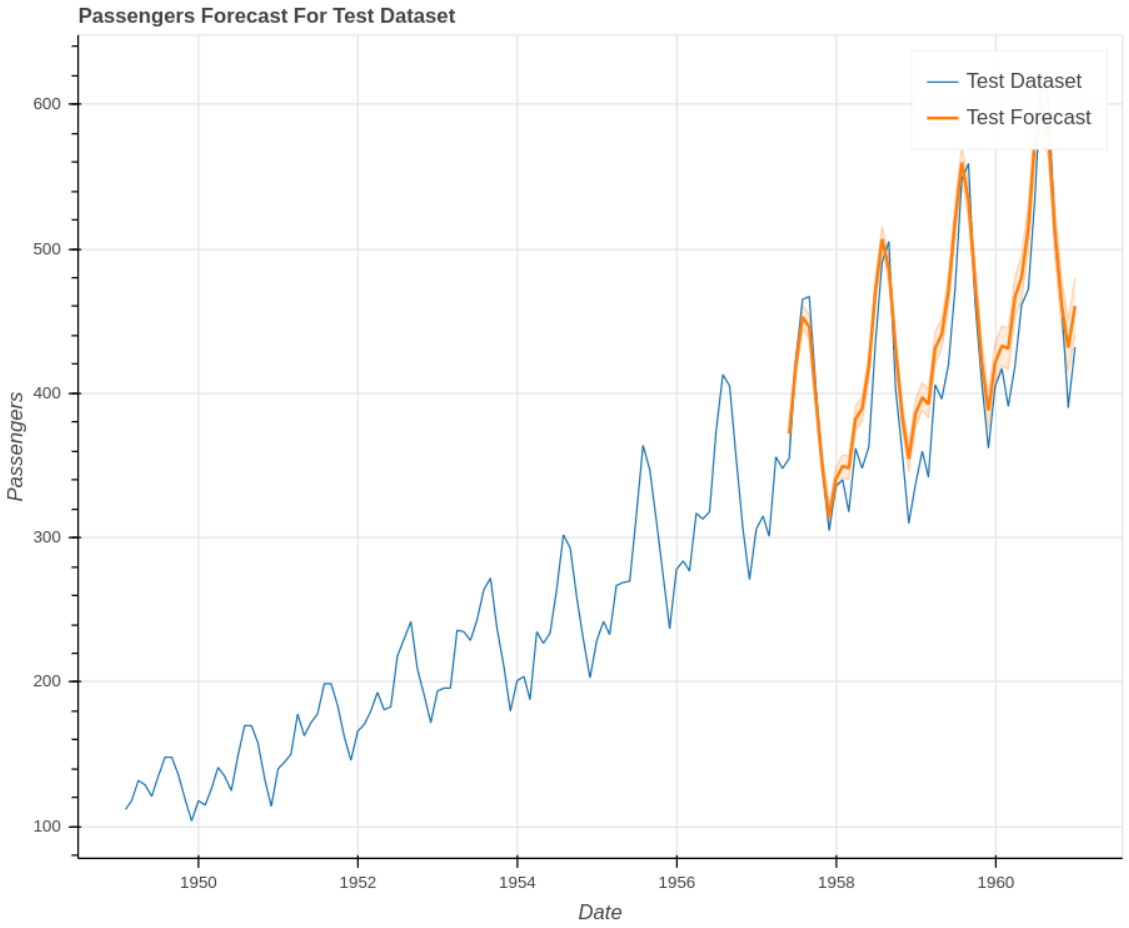

Figura 43: Projeção do modelo Prophet 8 sobre o conjunto de teste. Em azul temos o conjunto de dados original e em laranja a projeção do modelo.

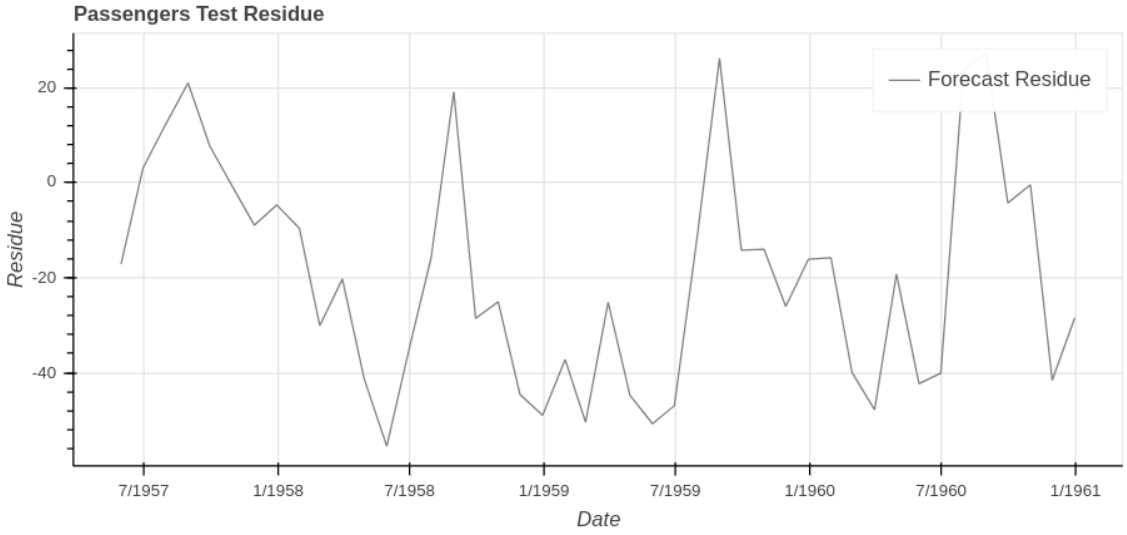

Figura 44: Resíduo do gráfico apresentado pela Figura 43.

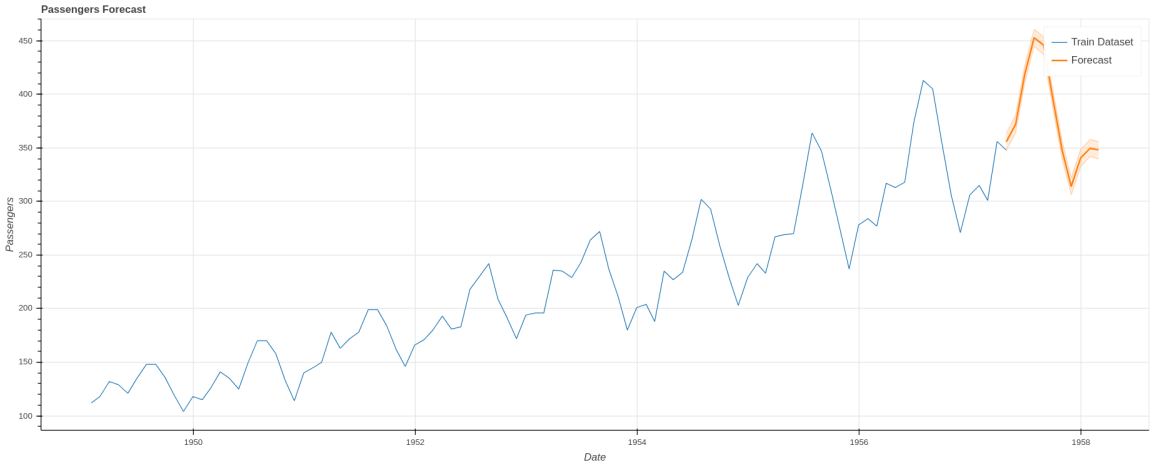

Figura 45: Projeção do modelo Prophet 8 dez meses a frente do último dado fornecido pelo dataset.

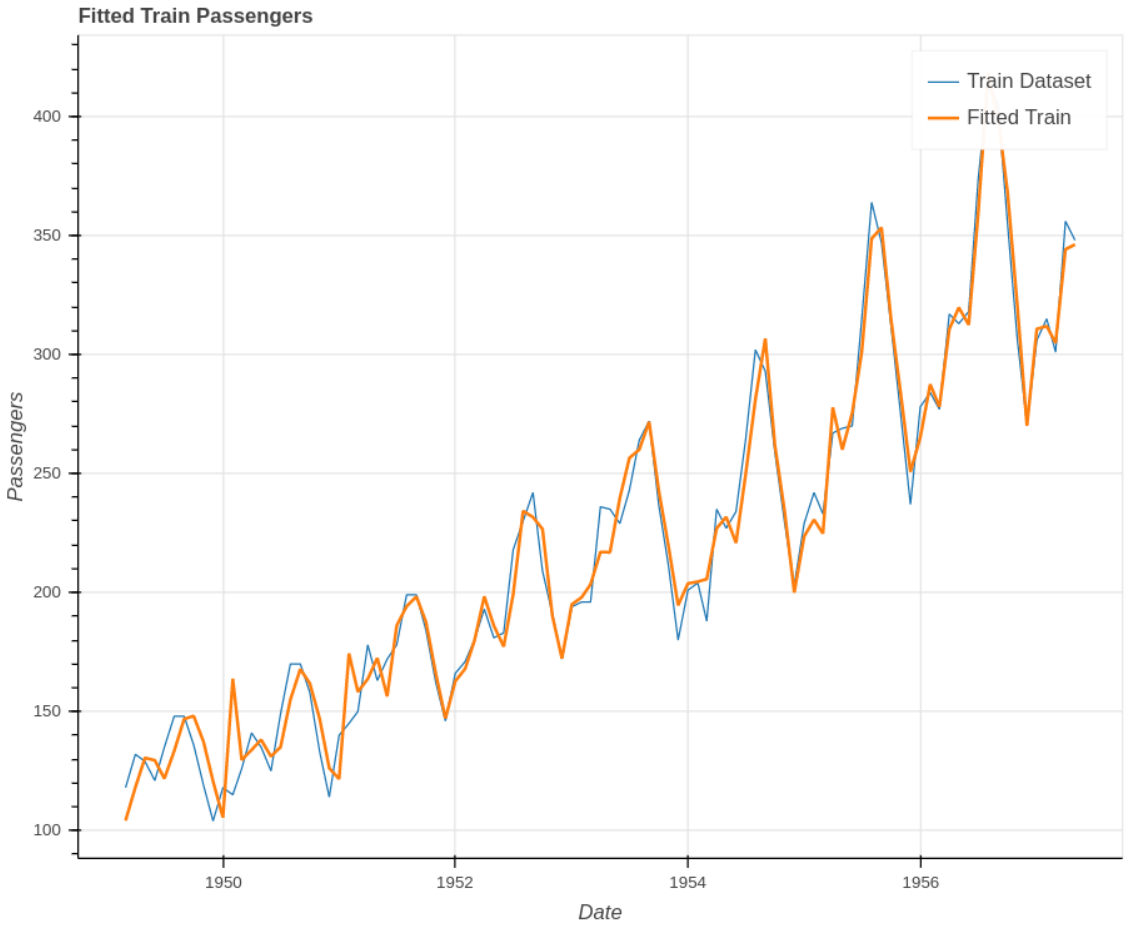

Figura 46: Modelo Arima Sazonal 9 fitado com o conjunto de treino. Em azul temos os dados de treino original e em laranja o fit do modelo.

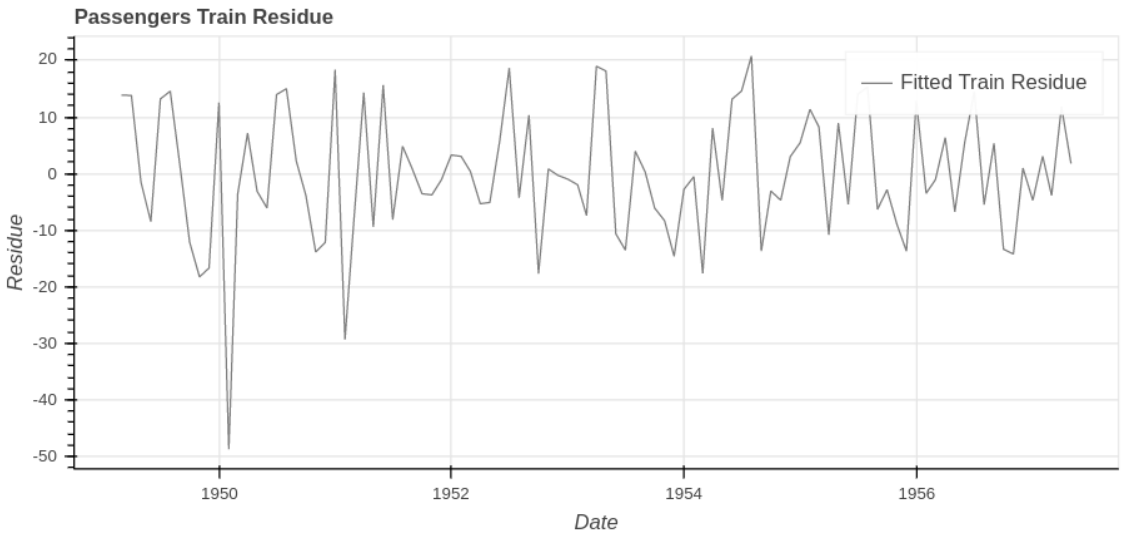

Figura 47: Resíduo do gráfico apresentado pela Figura 46.

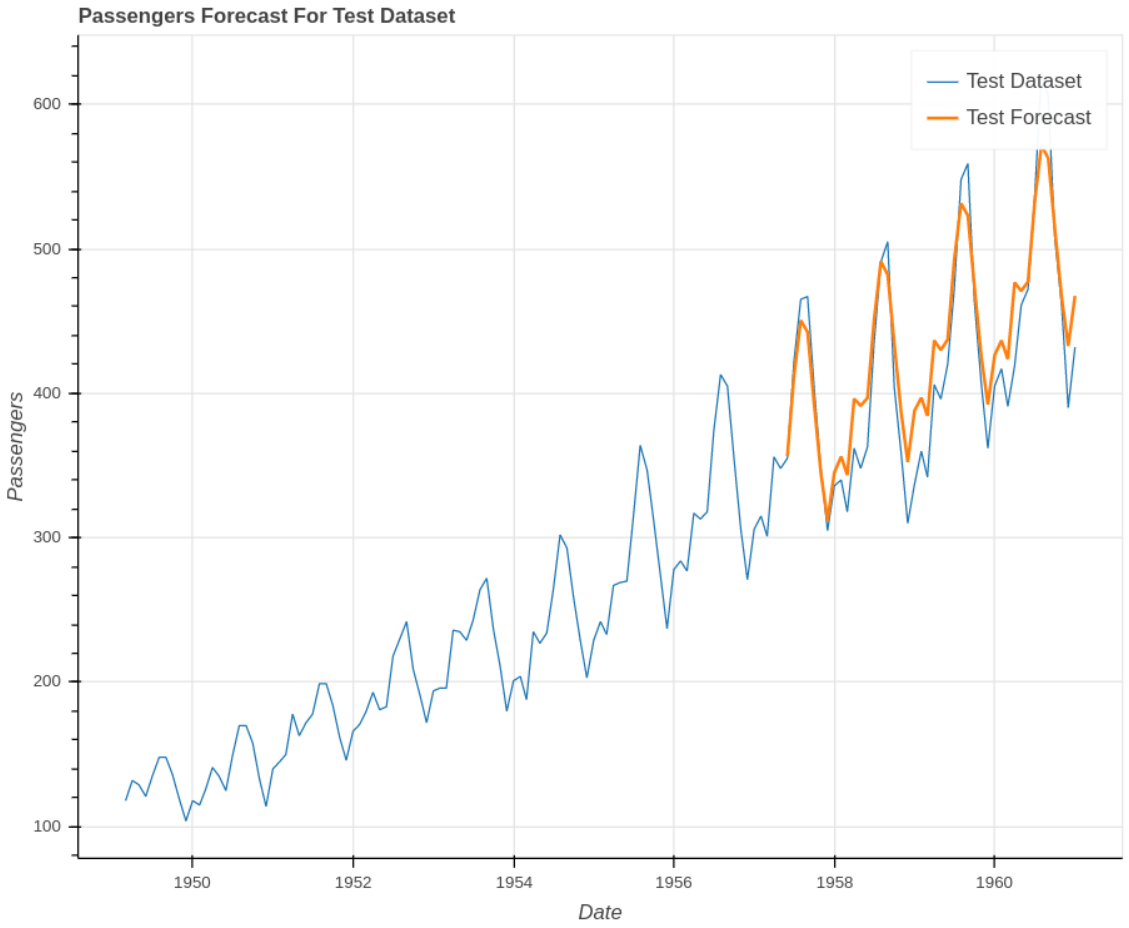

Figura 48: Projeção do modelo Arima Sazonal 9 sobre o conjunto de teste. Em azul temos o conjunto de dados original e em laranja a projeção do modelo.

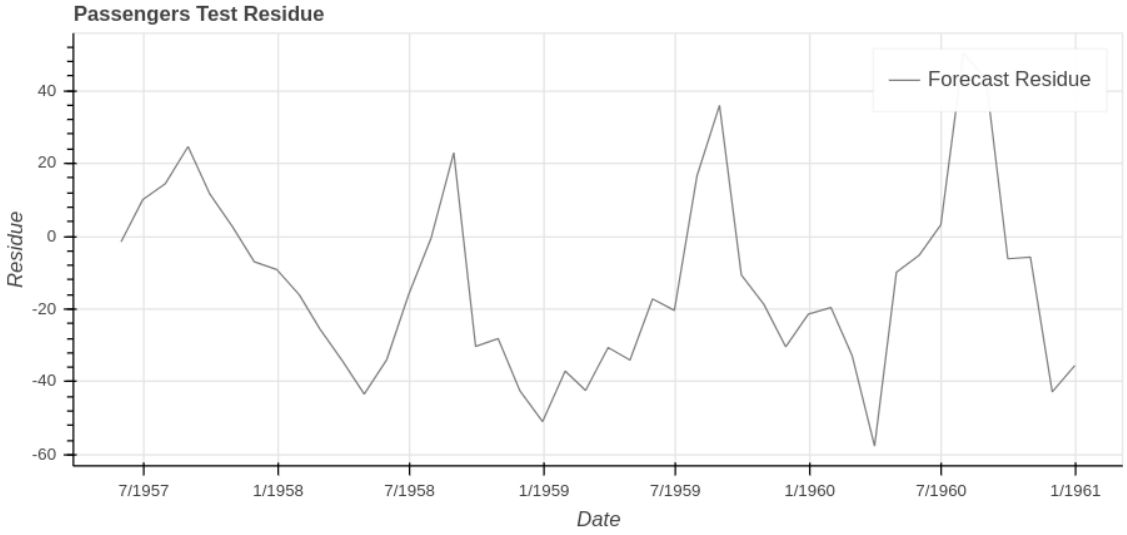

Figura 49: Resíduo do gráfico apresentado pela Figura 48.

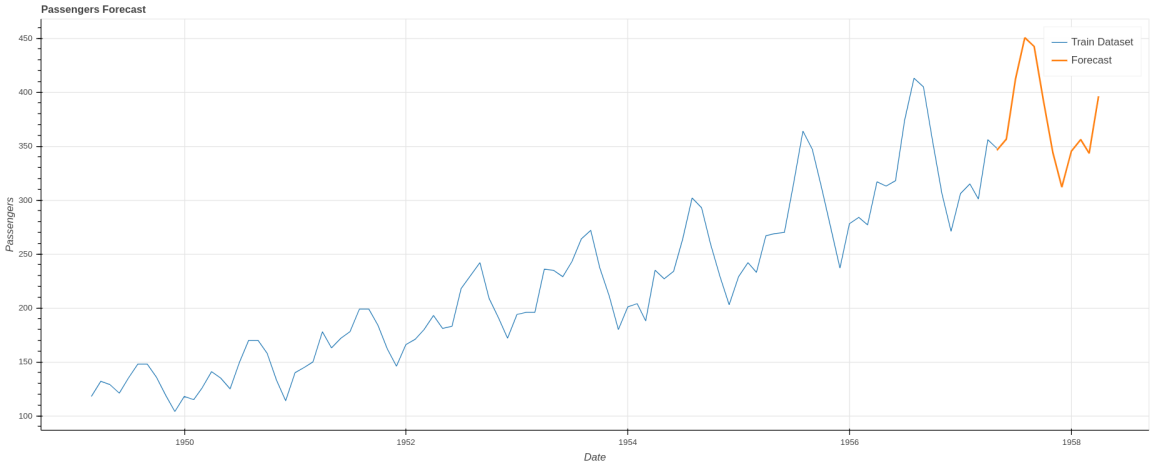

Figura 50: Projeção do modelo Arima Sazonal 9 dez meses a frente do último dado fornecido pelo dataset.