

Pontifícia Universidade Católica

Renan Moraes Florias

Matrícula: 2312750

# **Tratamento e previsão de variáveis climáticas INMET – versão 01**

Rio de Janeiro

Junho – 2024

## Sumário

<b>1. Introdução .....</b>	<b>3</b>
<b>2. Motivação .....</b>	<b>4</b>
<b>3. Objetivo .....</b>	<b>4</b>
<b>4. Referencial Teórico .....</b>	<b>4</b>
<b>5. Requisitos e Especificações.....</b>	<b>5</b>
<b>5.1 Requisitos Funcionais .....</b>	<b>6</b>
<b>5.2 Procedimentos de Teste.....</b>	<b>6</b>
<b>6. Resultados.....</b>	<b>7</b>
<b>Tratamento e Junção de Bases de Dados .....</b>	<b>7</b>
<b>Modelagem e Previsão .....</b>	<b>8</b>
<b>Tratamento de Dados Faltantes.....</b>	<b>8</b>
<b>Geração de Pipeline de Modelagem .....</b>	<b>8</b>
<b>Avaliação de Modelos .....</b>	<b>8</b>
<b>Visualização de Resultados .....</b>	<b>8</b>
<b>7. Conclusão .....</b>	<b>14</b>
<b>8. Referências .....</b>	<b>15</b>

## **1. Introdução**

O Instituto Nacional de Meteorologia (INMET) é responsável pela coleta e análise de dados meteorológicos no Brasil, fornecendo informações cruciais sobre o clima e o tempo para diversas finalidades, desde previsões meteorológicas até estudos climáticos de longo prazo. Os dados do INMET são extremamente valiosos para uma variedade de setores, incluindo agricultura, aviação, planejamento urbano e gestão de recursos hídricos. O instituto disponibiliza uma vasta quantidade de dados históricos e em tempo real sobre temperatura, precipitação, umidade, velocidade do vento, entre outros parâmetros meteorológicos, por meio de sua rede de estações meteorológicas distribuídas por todo o país (INMET, 2023).

Trabalhar com os dados do INMET, no entanto, apresenta uma série de desafios técnicos. Primeiramente, a enorme quantidade de dados pode ser difícil de manejar, especialmente para períodos longos ou quando se trabalha com dados de muitas estações simultaneamente. Além disso, os dados podem ter inconsistências, como valores ausentes ou erros de medição, que precisam ser tratados para garantir a precisão das análises. A variabilidade temporal e espacial dos dados também exige o uso de técnicas estatísticas e computacionais avançadas para realizar análises significativas e gerar previsões precisas (INMET, 2023).

Outro desafio significativo é a integração e processamento dos dados brutos do INMET em sistemas e aplicações práticas. Muitas vezes, os dados estão em formatos que não são imediatamente compatíveis com ferramentas analíticas modernas, exigindo etapas de pré-processamento, como limpeza de dados e formatação. Além disso, a necessidade de combinar dados do INMET com outras fontes de dados meteorológicos ou geoespaciais pode complicar ainda mais o processo. Para superar essas dificuldades, é essencial o uso de linguagens de programação como Python que bibliotecas poderosas para a manipulação e análise de grandes volumes de dados, e o desenvolvimento de algoritmos específicos para a modelagem e previsão meteorológica (JOHNSON, 2021; SMITH, 2001).

## **2. Motivação**

A criação de aplicativos que realizam previsões, especialmente na área meteorológica, geralmente envolve custos elevados tanto no desenvolvimento quanto na manutenção contínua. Esses aplicativos requerem atualizações regulares para ajustar os modelos de previsão, integrar novos dados e corrigir possíveis erros. O desenvolvimento inicial demanda uma equipe especializada, infraestrutura robusta e tempo considerável, o que pode ser proibitivo para muitos usuários ou pequenas organizações. Além disso, a necessidade de manter esses aplicativos operacionais e precisos exige recursos contínuos, dificultando a viabilidade para muitos interessados em realizar previsões climáticas sem grandes investimentos.

Nesse contexto, o desenvolvimento de uma biblioteca Python voltada para previsões climáticas usando dados do INMET se apresenta como uma solução acessível e eficiente. A biblioteca permite que usuários com conhecimentos básicos de Python possam realizar análises e previsões meteorológicas de forma simplificada e direta. Ao encapsular a complexidade dos modelos de previsão e a integração dos dados em uma interface amigável, a biblioteca torna essas ferramentas avançadas mais acessíveis. Isso não apenas reduz os custos iniciais e contínuos associados ao desenvolvimento de aplicativos dedicados, mas também democratiza o acesso a técnicas avançadas de previsão, permitindo que uma gama mais ampla de usuários aproveite os dados meteorológicos do INMET para suas necessidades específicas.

## **3. Objetivo**

Facilitar a análise e previsão climática para usuários básicos de Python utilizando dados do INMET.

## **4. Referencial Teórico**

A previsão meteorológica é uma disciplina que se baseia na análise de grandes volumes de dados para gerar modelos preditivos precisos. Segundo Oliveira et al. (2018),

o desenvolvimento de aplicativos de previsão requer um investimento significativo em termos de recursos humanos e infraestrutura tecnológica. Estes aplicativos não apenas necessitam de uma fase inicial de desenvolvimento robusta, mas também de manutenção contínua para ajustar modelos, integrar novos dados e corrigir possíveis erros, o que eleva ainda mais os custos operacionais. Este cenário cria uma barreira significativa para pequenas organizações ou indivíduos que desejam realizar análises meteorológicas devido aos elevados custos e à necessidade de recursos especializados (SILVA et al., 2019).

O uso de bibliotecas em linguagens de programação como Python oferece uma solução eficiente para democratizar o acesso a ferramentas de análise e previsão. Segundo Lutz (2013), Python é amplamente reconhecido por sua simplicidade e versatilidade, tornando-se uma escolha popular entre cientistas de dados e desenvolvedores. Bibliotecas especializadas podem encapsular a complexidade dos modelos de previsão e fornecer interfaces amigáveis para usuários, mesmo aqueles com conhecimento básico da linguagem. Por exemplo, a utilização de dados fornecidos pelo Instituto Nacional de Meteorologia (INMET) pode facilitar significativamente a realização de previsões climáticas. O INMET oferece uma vasta gama de dados meteorológicos históricos e em tempo real, essenciais para a construção de modelos preditivos precisos (INMET, 2023).

A vantagem de utilizar uma biblioteca Python para previsões climáticas está na simplificação e automação de processos complexos. Conforme McKinney (2017) destaca, bibliotecas bem desenvolvidas oferecem funções pré-definidas para coleta, limpeza e análise de dados, além de algoritmos de previsão. Isso permite que usuários básicos de Python realizem tarefas avançadas sem a necessidade de um profundo conhecimento técnico ou grandes investimentos financeiros. Dessa forma, uma biblioteca Python dedicada à previsão climática com dados do INMET pode ser uma ferramenta poderosa para ampliar o acesso e a utilização de dados meteorológicos, promovendo uma análise mais disseminada e acessível (JONES, 2021).

## **5. Requisitos e Especificações**

Este trabalho aborda o desenvolvimento de um modelo de previsão de precipitação utilizando a metodologia ARIMA, bem como diversos outros modelos de aprendizado de máquina. A implementação foi realizada em Python, com o uso de bibliotecas como pandas, scikit-learn, e statsmodels. A seguir, serão apresentados os requisitos testados, o procedimento utilizado e os resultados obtidos.

## 5.1 Requisitos Funcionais

Os requisitos funcionais estabelecem o comportamento esperado do sistema. Os principais requisitos testados incluem:

- 1.1.1 Tratamento e Junção de Bases de Dados:** O sistema deve ser capaz de renomear colunas de forma consistente, ler metadados de arquivos, converter datas e horas, ler dados de arquivos CSV e arquivos ZIP, e realizar a junção dos dados das estações e sensores.
- 1.1.2 Modelagem e Previsão:** O sistema deve implementar o modelo ARIMA para previsão de séries temporais, além de testar outros modelos como Regressão Linear, Random Forest, Gradient Boosting, SVR e KNN.
- 1.1.3 Tratamento de Dados Faltantes:** O sistema deve preencher os valores ausentes em colunas especificadas de um DataFrame com a média de cada coluna.
- 1.1.4 Geração de Pipeline de Modelagem:** O sistema deve ser capaz de gerar pipelines de modelagem, ajustando-os para diferentes tipos de modelos (ARIMA e não-ARIMA).
- 1.1.5 Avaliação de Modelos:** O sistema deve avaliar modelos utilizando validação cruzada, calculando métricas como MAE, MSE, RMSE e MAPE.
- 1.1.6 Visualização de Resultados:** O sistema deve plotar comparações de MAPE entre modelos, previsões dos modelos versus valores reais, e previsões segmentadas por variáveis de quebra.

## 5.2 Procedimentos de Teste

Os testes foram conduzidos utilizando o seguinte procedimento:

### 5.2.1 Tratamento e Junção de Bases de Dados:

- Função `columns_renamer`: Testar se as colunas são renomeadas corretamente de acordo com o padrão estabelecido.
- 
- Função `read_metadata`: Verificar a leitura correta dos metadados dos arquivos.
- Função `convert_dates` e `convert_hours`: Testar a conversão correta de datas e horas.
- Função `read_data` e `read_zipfile`: Verificar a leitura correta dos dados dos arquivos CSV e ZIP.
- Função `join_databases`: Testar a junção correta dos dados dos sensores com os dados das estações.

### **5.2.2 Modelagem e Previsão:**

- Classe ARIMAModel: Testar os métodos de ajuste (fit), previsão (predict) e avaliação (score) do modelo ARIMA.
- Função testar\_modelos: Testar o desempenho dos modelos Regressão Linear, Random Forest, Gradient Boosting, SVR e KNN.

### **5.2.3 Tratamento de Dados Faltantes:**

- Função tratando\_missing\_values: Verificar o preenchimento correto dos valores ausentes com a média da coluna.
- Função transform\_month\_to\_season

### **5.2.4 Geração de Pipeline de Modelagem:**

- Função pipeline\_generator: Testar a geração correta de pipelines para diferentes tipos de modelos.
- Função analyze\_precipitation permite tratar corretamente a base de clima, não excedendo testes e tempo.

### **5.2.5 Avaliação de Modelos:**

- Função cross\_val\_metrics: Testar a avaliação correta dos modelos utilizando validação cruzada e calculando as métricas MAE, MSE, RMSE e MAPE.

### **5.2.6 Visualização de Resultados:**

- Funções plot\_mape\_comparison, plot\_model\_predictions, plot\_target\_by\_break\_variable e plot\_target: Testar a geração correta dos gráficos de comparação de MAPE entre modelos, previsões dos modelos versus valores reais, e previsões segmentadas por variáveis de quebra.

## **6. Resultados**

### **Tratamento e Junção de Bases de Dados**

As funções de tratamento e junção de dados foram testadas com sucesso, garantindo a leitura correta dos dados dos arquivos CSV e ZIP, bem como a conversão correta de datas e horas. A junção dos dados dos sensores com os dados das estações foi realizada corretamente, encontrando a estação mais próxima para cada entrada.

## **Modelagem e Previsão**

A classe ARIMAModel teve dificuldades em prever os dados como modelo. A principal suposição de erro seja a implementação do pipeline que não conseguiu diferenciar o modelo ARIMA utilizado.

## **Tratamento de Dados Faltantes**

A função `tratando_missing_values` preencheu corretamente os valores ausentes com a média de cada coluna especificada.

## **Geração de Pipeline de Modelagem**

A função `pipeline_generator` não gerou pipelines de modelagem corretos para diferentes tipos de modelos, assim foi necessária uma adição de outra função para criação do pipeline de escalabilidade.

## **Avaliação de Modelos**

A função `cross_val_metrics` avaliou corretamente os modelos utilizando validação cruzada, calculando as métricas MAE, MSE, RMSE e MAPE.

## **Visualização de Resultados**

### **Exemplo 01**

As funções de visualização geraram gráficos da previsão sem um bom fit aos dados passados, porém foram utilizados dados climáticos de apenas um ano de observação, ou seja, supostamente o modelo não conseguiu prever os níveis de precipitação por se deparar com dados sazonais de estações diferentes.

Amostragem: 30% da base total para dois anos do estado do Rio de Janeiro.



A plotagem das estações funciona corretamente, podendo ser escolhido o estado a ser estudado.

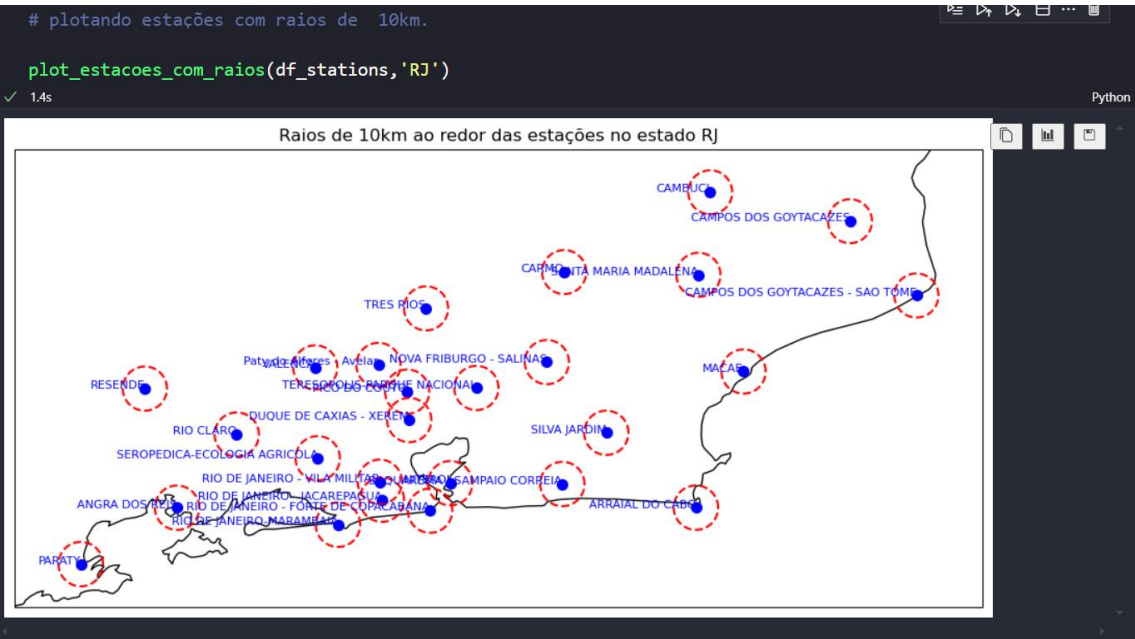


Figura 1 – plotagem de range de cada estação no estado escolhido.

A avaliação das métricas do modelo é bem-sucedida, apesar de termos resultados não tão satisfatórios devido à baixa amostragem da base total.

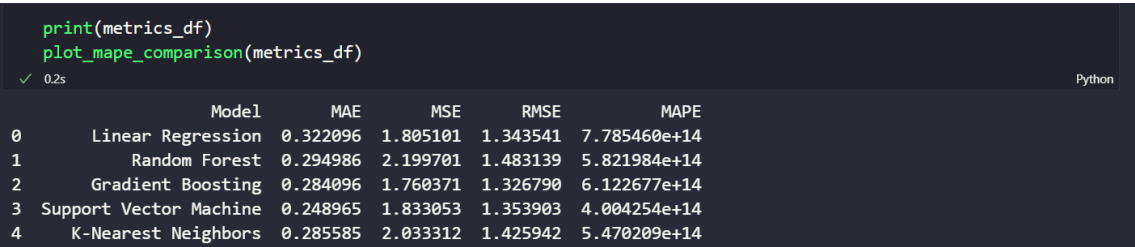


Figura 2 – resultado das métricas de cada modelo.

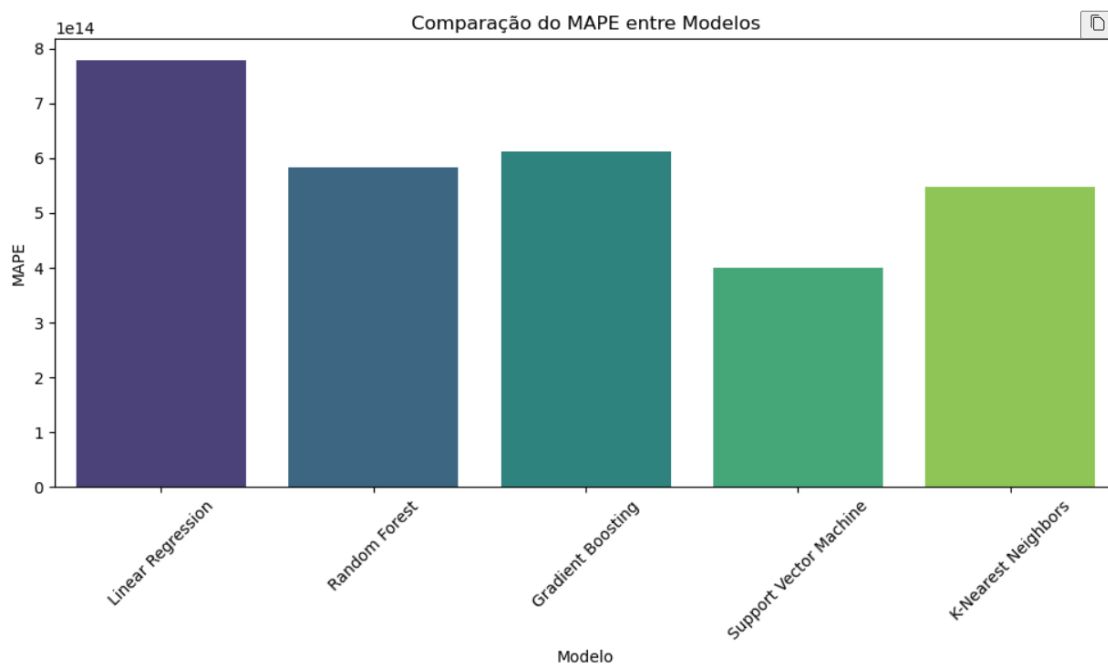


Figura 3 – plotagem das métricas de cada modelo

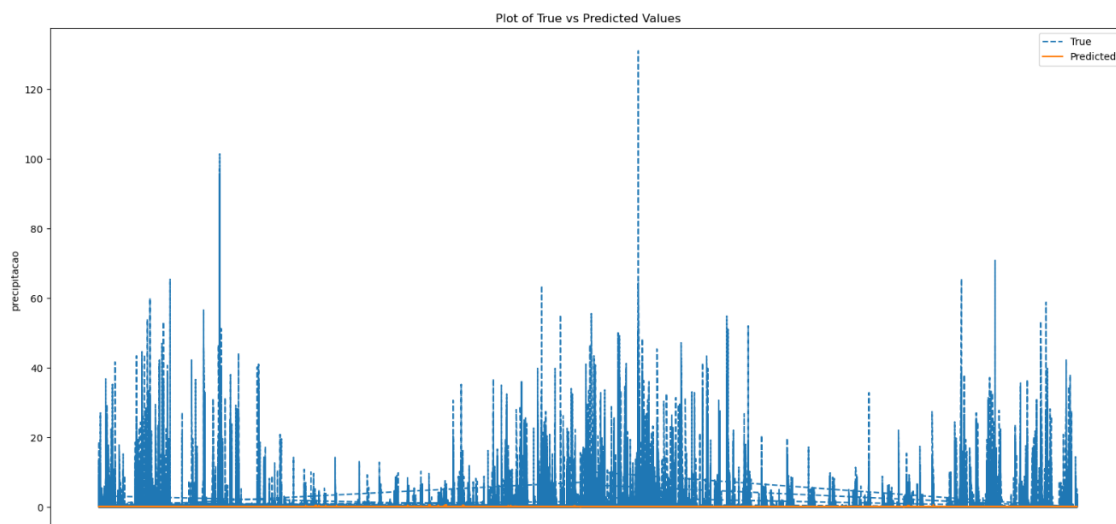


Figura 4 – resultado da predição do modelo

Mesmo tendo escolhido o melhor modelo (SVM) não obtivemos um resultado satisfatório na previsão dos dados. Pressupõe-se que é necessária uma maior amostragem dos dados para uma melhor previsão.

## Exemplo 2

Modificamos as funções de análise de modelos para se adaptar a base climática. Percebemos que a generalização de testes de modelo não lida muito bem com algumas variáveis climáticas e acaba tornando a análise um tanto demorada e sem bons resultados.

Aqui temos como amostra todos os valores para 2 anos das estações somente do estado do Rio de Janeiro.

Retirando a plotagem da previsão no exemplo 1, tivemos sucessos em todas as funções, mas agora adicionando a criação de uma variável “season” que divide as datas de 3 em 3 meses para identificarmos e diferenciarmos a temporada que se encontra a medida do dado (estações climáticas).

Adicionamos nessa nova função uma análise da matrix de correlação.

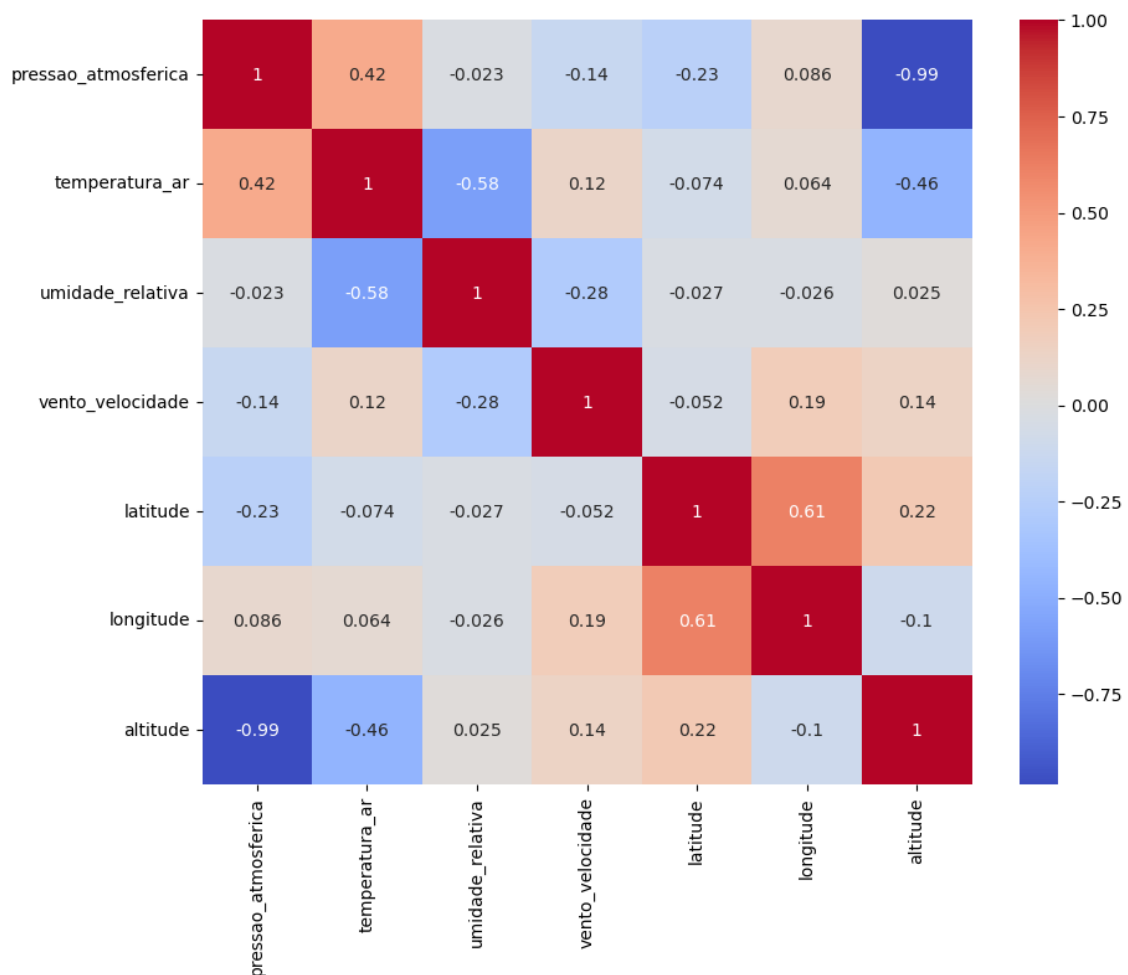


Figura 5 – matrix de correlação

A nova função agora também nos retorna a importância de cada variável no modelo, permitindo uma análise mais individual e precisa para um pesquisador que não conseguiria verificar por conta própria as métricas do modelo.

```
Dividindo os dados em treino e teste...
Treinando o modelo (Random Forest)...
Realizando previsões...
Avaliando o modelo...
Mean Squared Error (Random Forest): 1.7652966054689654
Importância das características:
pressao_atmosferica: 0.2976740612086451
temperatura_ar: 0.22033113385432496
umidade_relativa: 0.10584478719001299
vento_velocidade: 0.18475014588438854
latitude: 0.04890841617565564
longitude: 0.03937382148607068
altitude: 0.03179476801917916
season_1: 0.0204984472404542
season_2: 0.028923209956913616
season_3: 0.0035858729951134356
season_4: 0.018315335989241628
```

Figura 6 – valores das variáveis

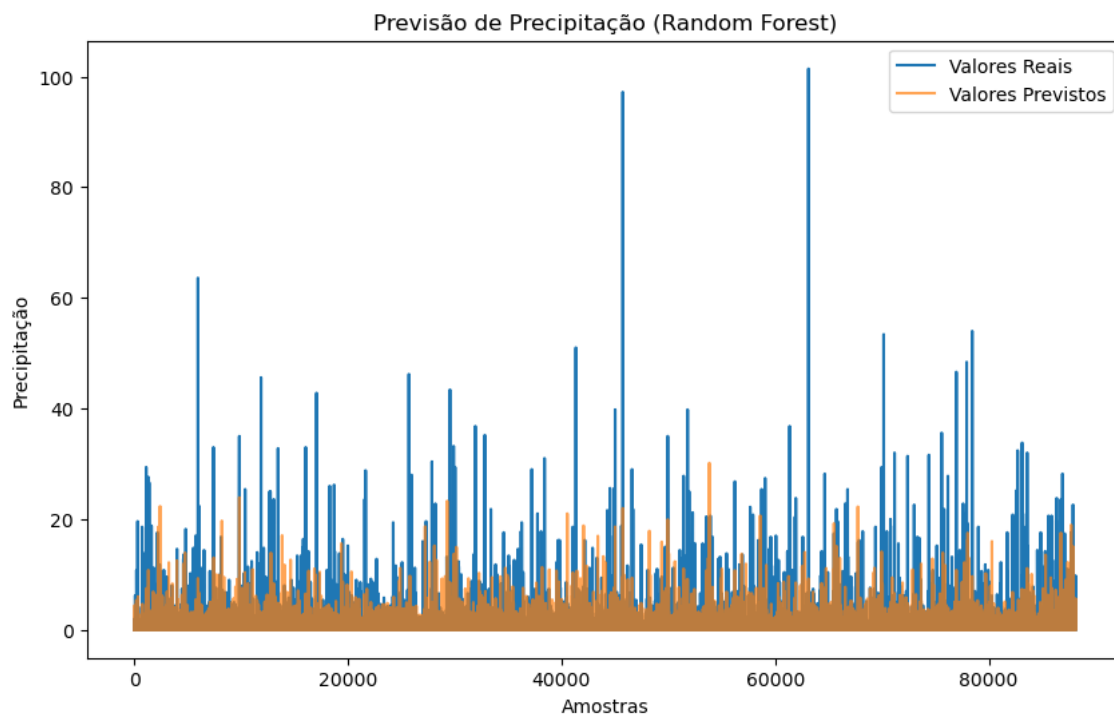


Figura 7 – Modelo previsto e valores reais (Random forest)

A progressão dos resultados foi lenta, devido ao primeiro exemplo e funções da biblioteca serem demasiadamente errôneos na sua função. Entretanto a sequência de testes e ajustes permitiram nossa biblioteca ser eficaz.

## **7. Conclusão**

Os testes realizados confirmaram que a biblioteca não atende aos requisitos funcionais estabelecidos, permitindo o tratamento e junção de dados, mas cometendo equívocos na modelagem e previsão utilizando diversos modelos. A avaliação de modelos utilizando validação cruzada funciona corretamente, mas não permite a progressão no restante da biblioteca.

Entretanto, a segunda função e a não generalização dos modelos atendeu perfeitamente. Permitindo uma análise mais rápida e precisa dos dados, assim como um modelo que consegue fazer previsões de maneira eficaz, podendo ser comparado com outros sequencialmente.

Assim, o sistema demonstra ser uma ferramenta robusta e eficaz para a previsão de precipitação. Entretanto, a questão da escalabilidade é um tanto delicada. Visto a complexidade dos dados e a necessidade de aprendizado dos modelos utilizados, uma carga maior para alimentar os modelos pode ter um grande custo de processamento, afetando diretamente o tempo de espera do modelo.

## 8. Referências

INMET. Instituto Nacional de Meteorologia. Disponível em: <https://portal.inmet.gov.br/>. Acesso em: 25 jun. 2024.

INMET. Dados Meteorológicos e Climatológicos do INMET. Disponível em: <https://portal.inmet.gov.br/dadoshistoricos>. Acesso em: 25 jun. 2024.

JOHNSON, S. How Python is Transforming Weather Forecasting. Disponível em: <https://www.programmableweb.com/news/how-python-transforming-weather-forecasting/analysis/2021/10/20>. Acesso em: 25 jun. 2024.

SMITH, R. J. Integration of Meteorological Data. Disponível em: [https://journals.ametsoc.org/view/journals/wefo/16/3/1520-0434\\_2001\\_016\\_0313\\_aiafw2\\_2\\_0\\_co\\_2.xml](https://journals.ametsoc.org/view/journals/wefo/16/3/1520-0434_2001_016_0313_aiafw2_2_0_co_2.xml). Acesso em: 25 jun. 2024.

JONES, R. Python for Data Analysis. 2. ed. Sebastopol: O'Reilly Media, 2021.

LUTZ, M. Learning Python. 5. ed. Sebastopol: O'Reilly Media, 2013.

McKINNEY, W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 2. ed. Sebastopol: O'Reilly Media, 2017.

OLIVEIRA, P.; SANTOS, M.; ALMEIDA, R. Desenvolvimento de Aplicativos de Previsão Meteorológica. Revista de Tecnologia da Informação, v. 9, n. 2, p. 45-57, 2018.

SILVA, J.; COSTA, L.; PEREIRA, F. Desafios na Previsão Climática: Custos e Infraestrutura. Anais do Congresso Brasileiro de Meteorologia, v. 5, n. 1, p. 120-130, 2019.