

# Desafio de Séries Temporais

Victor Gomes de Carvalho - 11275168

## Link para o projeto:

[https://colab.research.google.com/drive/1NjVjHihDMwKQpVm\\_DgoC0C\\_65zrriuiB?usp=sharing](https://colab.research.google.com/drive/1NjVjHihDMwKQpVm_DgoC0C_65zrriuiB?usp=sharing)

## 1. Introdução

Séries temporais são muito utilizadas em áreas como previsão de demanda, vendas, mercado financeiro, correção ortográfica, vídeos e áudios. Devido à dependência da análise de dados passados, são altamente complexas.

As principais aplicabilidades são:

- encontrar padrões de comportamento ao longo do tempo;
- prever valores futuros se baseando no passado.

Componentes de uma série temporal:

- tendência: direção dos valores da variável em relação ao tempo (alta/baixa; crescimento/decrescimento, etc);
- sazonalidade: qualquer mudança ou padrão previsível, ou seja, repetição de comportamento (aumento de vendas no Natal, queda de ocupação de hotéis em baixa temporada, etc).

Através desse projeto, o objetivo é extrair o potencial das séries temporais para tentar prever o padrão de temperatura em Delhi pelos próximos meses, através de dados climáticos coletados entre 2013 e 2017.

## 2. Análise exploratória de dados

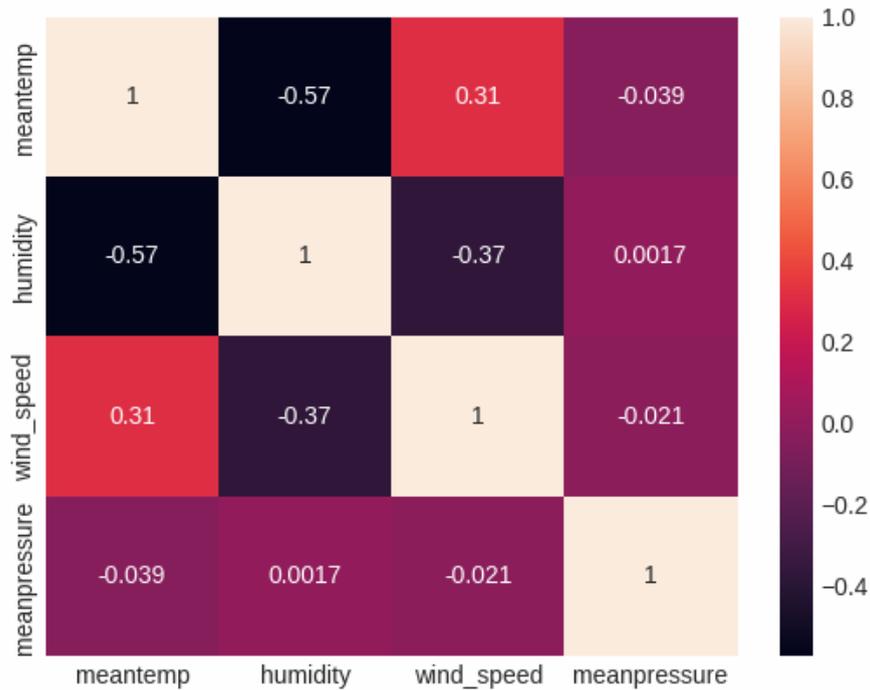
O dataset é composto por 5 atributos, sendo quatro numéricos e apenas um categórico:

- **date** - Data no formato YYYY-MM-DD
- **meantemp** - Temperatura média de uma sequência de medições espaçadas em 3 horas ao longo do dia
- **humidity** - Valor da umidade em gramas de vapor d'água por metro cúbico de ar
- **wind\_speed** - Velocidade do vento em quilômetros por hora
- **meanpressure** - Pressão do clima medida em atm

O dataset é dividido em 2 arquivos csv separados. São 1462 entradas no conjunto de treino e 114 no conjunto de teste. Um ponto positivo é que todos os campos são preenchidos.

```
RangeIndex: 1462 entries, 0 to 1461
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   date            1462 non-null   object
1   meantemp       1462 non-null   float64
2   humidity       1462 non-null   float64
3   wind_speed    1462 non-null   float64
4   meanpressure  1462 non-null   float64
```

Abaixo temos um gráfico que mostra a correlação entre os atributos presentes no dataset. Pode-se observar uma correlação positiva entre a temperatura e a velocidade do vento. Ao mesmo tempo, há uma correlação negativa entre a temperatura e a umidade, o que faz sentido pois remonta ao fenômeno da amplitude térmica. Também há uma correlação negativa entre a umidade e a velocidade do vento. A pressão média praticamente não tem correlação com os outros atributos.



Vamos agora visualizar como os diferentes atributos se comportam ao longo do tempo.



Gráfico da temperatura em função do tempo



Gráfico da umidade em função do tempo

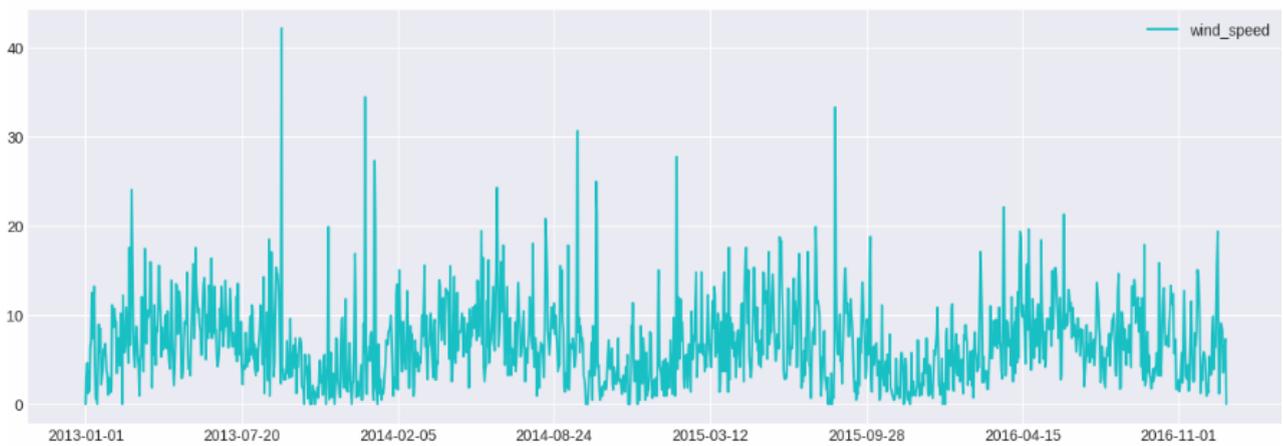


Gráfico da velocidade do vento em função do tempo

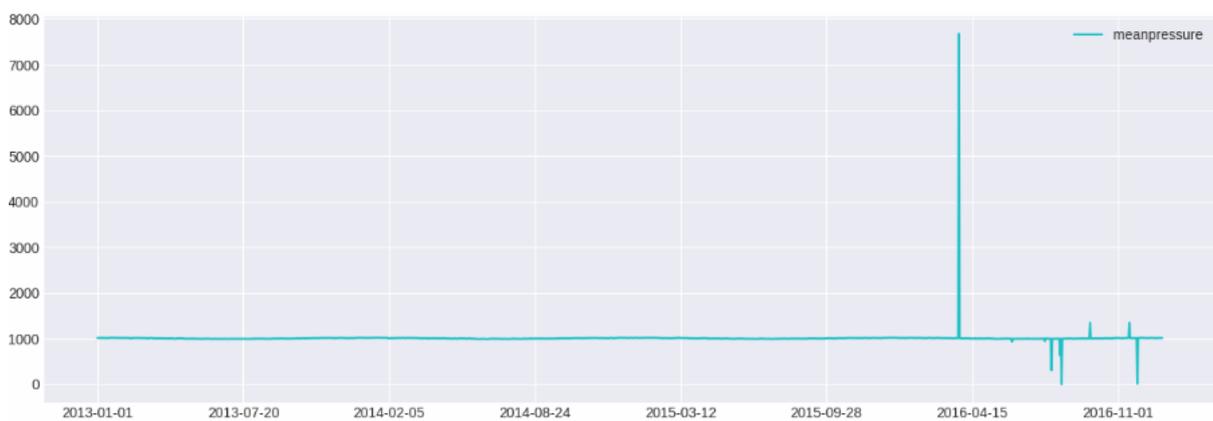


Gráfico da pressão média em função do tempo

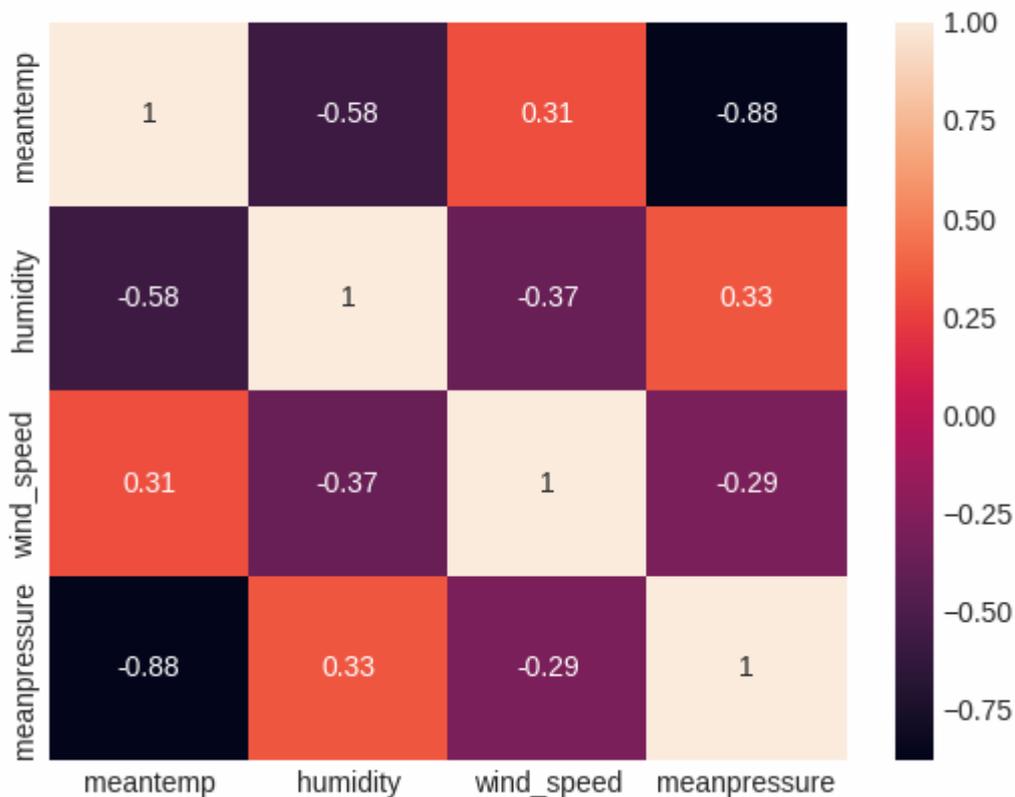
Como série temporal da pressão média tem nitidamente outliers significativos, foi gerada uma nova série com essas instâncias removidas através de amplitude interquartil. Isso resultou no seguinte gráfico:



Gráfico da pressão média em função do tempo (sem outliers)

Abaixo temos a nova matriz de correlação entre os atributos. Percebe-se que a pressão atmosférica está fortemente relacionada negativamente com a temperatura, e agora também possui certo grau de conexão com os outros atributos, que é natural que faça mais sentido do que os números desprezíveis da matriz anterior.

Matrix de correlação entre os atributos

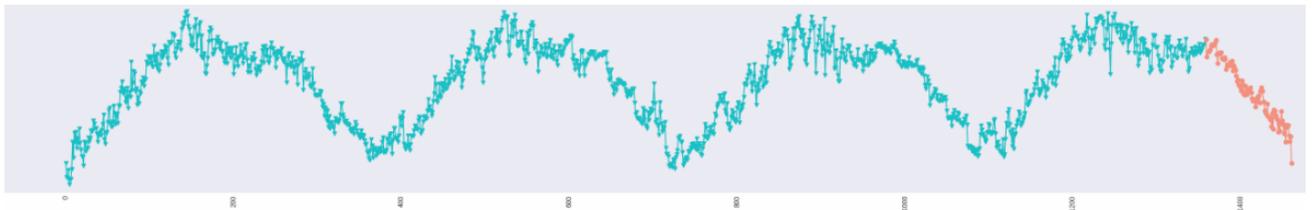


Pode-se concluir a partir dos gráficos acima que todos os atributos possuem certo grau de sazonalidade, ou seja, repetição de comportamento - consequentemente, são séries não estacionárias. Também apresentam pouca ou praticamente nenhuma tendência, oscilando

entre uma faixa de valores durante o ano inteiro. A temperatura média aumenta até o pico em meados de Maio e então decai com um vale em Setembro/Outubro. A umidade desce abruptamente por volta de Fevereiro/Março e aumenta, embora com alguma turbulência, entre Junho e Julho. A velocidade do vento em Deli não varia muito, com um pequeno pico no meio do ano, e decaimentos ao longo do resto. A pressão atmosférica tem uma variação relativamente regular, sempre aumentando a partir de Julho/Agosto e, em sequência, começa a diminuir em Março/Abril.

### 3. Modelagem

Como descrito inicialmente, o objetivo do desafio é prever a temperatura média em Delhi nos próximos meses. O dataset de treino foi dividido em treino e validação, com 100 instâncias reservadas para essa. Em todos os modelos, as features utilizadas foram meanpressure, wind\_speed e humidity. As métricas utilizadas foram MAPE, R2 e RMSE.

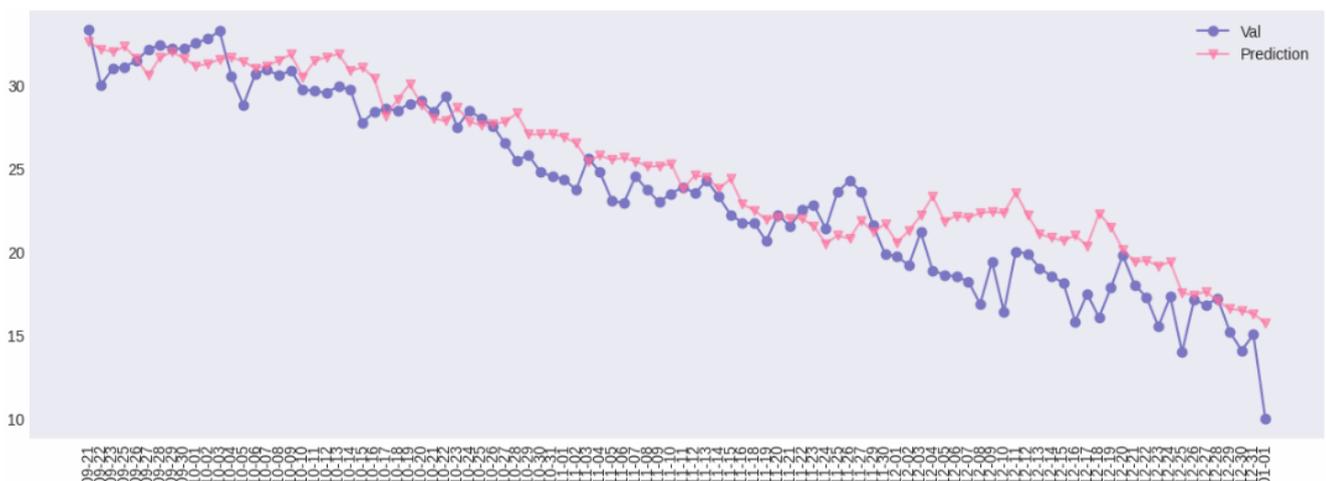


Divisão entre treino e validação

#### I. Arima

Para fazer a primeira previsão, foi utilizado o modelo ARIMA, que significa AutoRegressive Integrated Moving Average. É um modelo de previsão de séries temporais que usa os valores passados para prever os valores futuros se baseando na autocorrelação presentes nos dados ao invés de focar totalmente na sazonalidade da série. Para fazer a modelagem, foi utilizado um método chamado AutoArima, que otimiza o modelo Arima identificando a melhor configuração de seus hiperparâmetros.

A melhor configuração gerou essa série:



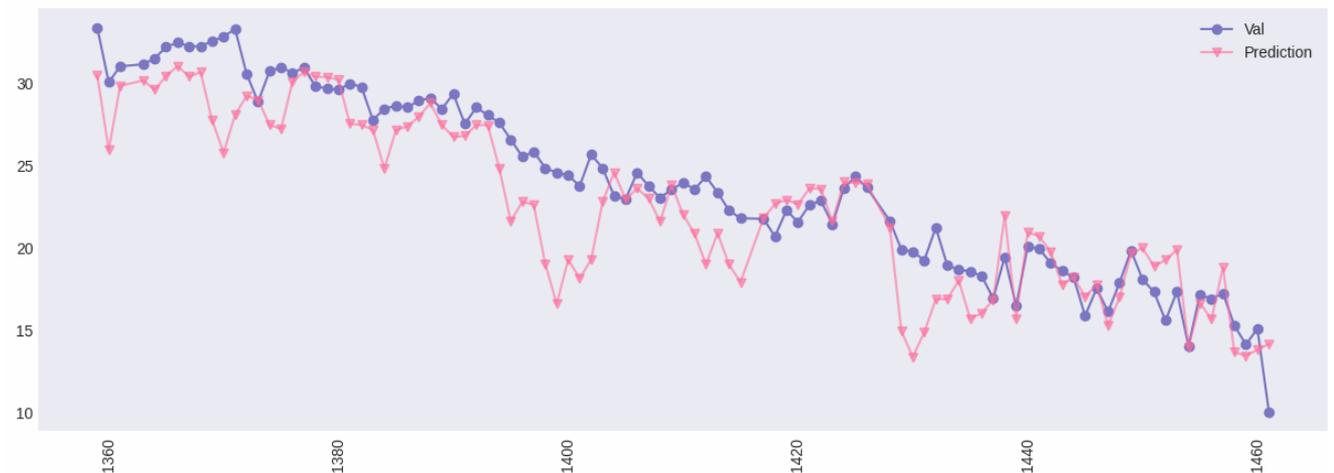
Os valores das métricas foram:

RMSE	2.24
R2	0.83
MAPE	0.08

## II. XGB

O nome XGBoost vem de eXtreme Gradient Boosting, e representa uma categoria de algoritmo baseada em Decision Trees (árvores de decisão) com Gradient Boosting (aumento de gradiente). É extremamente flexível – uma vez que possui um grande número de hiperparâmetros passíveis de aperfeiçoamento - você consegue ajustar adequadamente o XGBoost para o cenário do seu problema, seja ele qual for.

O modelo gerou a seguinte série temporal:



Os valores das métricas foram:

RMSE	2.70
R2	0.76
MAPE	0.09

### III. LSTM

A LSTM é uma arquitetura de rede neural recorrente (RNN) que lembra valores em intervalos arbitrários. A LSTM é bem adequada para classificar, processar e prever séries temporais com intervalos de tempo de duração desconhecida. Ao contrário das redes neurais feedforward padrão, o LSTM possui conexões de feedback.

O modelo gerou a seguinte série temporal:



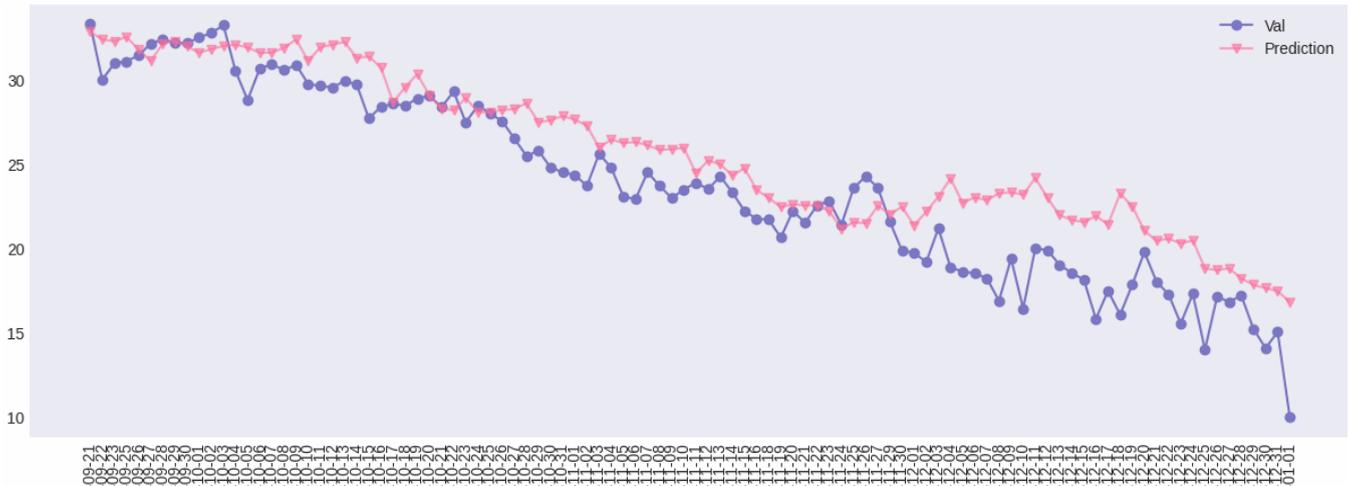
Os valores das métricas foram:

RMSE	5.33
R2	0.06
MAPE	0.19

### IV. Triple Exponential Smoothing

A TES é usada para lidar com os dados de séries temporais que contêm um componente sazonal. Este método é baseado em três equações de suavização: componente estacionária, tendência e sazonal. Tanto a sazonalidade quanto a tendência podem ser aditivas ou multiplicativas. Comumente aplicada na suavização de dados no processamento de sinal, atuando como filtros passa-baixo para remover o ruído de alta frequência.

O modelo gerou a seguinte série temporal:



Os valores das métricas foram:

RMSE	2.76
R2	0.74
MAPE	0.11

#### 4. Interpretação dos resultados

Construindo uma tabela com todos os modelos e suas respectivas métricas, nós temos:

	AutoArima	XGB	LSTM	TES
RMSE	2.24	2.70	5.33	2.76
R2	0.83	0.76	0.06	0.74
MAPE	0.08	0.09	0.19	0.11

O AutoArima foi o modelo de melhor desempenho em todas as métricas, o que de certa forma era esperado, já que captura várias estruturas temporais importantes. Além disso, como o AutoArima é uma versão do método que busca automaticamente pelos melhores parâmetros, pode-se dizer que o modelo já começou na vantagem durante o processo de modelagem. Apesar disso, foi o modelo cujo treinamento levou mais tempo

O XGB teve um resultado próximo, porém se saiu pior, provavelmente pela escolha de parâmetros não ser ótima. Já que é um método Ensemble, pode chegar a um overfitting muito facilmente, perdendo assim a capacidade de generalizar os padrões nos dados.

O TES teve um desempenho próximo ao XGB, mas assim como nos métodos anteriores, pode ter apresentado uma solução sub-ótima devido à escolha dos valores das três componentes. A suavização das curvas inerente à natureza do modelo também é um fator determinante, já que pode facilmente distorcer a interpretação dos dados.

O LSTM foi de longe o pior modelo. Vários motivos podem ter gerado esse cenário. Como é uma rede neural mais complexa do que redes recorrentes tradicionais, o número limitado de instâncias para o treinamento pode ter mitigado o potencial do modelo. A arquitetura das camadas também pode ter sido sub-ótima.

## 5. Resumo dos seminários

### Felipe Cadavez:

Apresentou o Transformador Separável (SepTr), que consiste em dividir a imagem de um espectrograma em tokens de altura igual a intervalos constantes da frequência e largura igual a intervalos constantes de tempo. Dessa forma os tokens juntamente com incorporação de dados de posição passam a criar uma imagem de dimensões token x token, ao invés de frequência x tempo.

O Espectrograma é dividido em tokens. Esses tokens são separados em colunas de mesmo tempo e incorporados com dados de posição (transformador vertical). Após isso é realizado um agrupamento médio dos tokens e eles são separados em intervalos de mesma frequência e incorporados mais uma vez com dados de posição. Por fim temos um dataset de tokens, onde cada dataframe é um intervalo de tempo e frequência. Utilizou 3 datasets: ESC-50, Speech Commands V2 e CREMA-D.

### Débora Buzon:

Apresentou o prophet, que é um modelo de previsão que foi desenvolvido pelo time de Ciências de Dados do Facebook.

Objetivos do modelo:

- Facilitar a produção de previsões em escala
- Grande número de previsões
- Grande variedade de séries temporais diferentes
- Avaliação das previsões geradas (utilizando feedback dos analistas)

Abordagem:

- Modelar as séries temporais usando uma especificação flexível com uma interpretação simples dos parâmetros.
- Produzir previsões para esse modelo e para um conjunto de baselines para avaliar a performance das previsões.
- Quando tem uma performance ruim ou algum aspecto da previsão precisa ser analisada, sinalizar esses problemas para um analista humanos de forma ordenada, para que o modelo possa ser ajustado conforme necessário.
- Abordagem analyst-in-the-loop

or diferencial do modelo Prophet é ser simples e modular, que geralmente funciona bem com parâmetros padrão, permitindo que os analistas selecionem os componentes relevantes para seu problema de previsão e façam ajustes conforme necessário. Além disso, o sistema para medir e rastrear a precisão das previsões, sinalizando aquelas que devem ser verificadas manualmente, é essencial para que os analistas identifiquem quando é necessário ajustar o modelo ou quando um modelo totalmente diferente pode ser mais adequado.

Gustavo Bartolomeu:

Utilizar modelos de aprendizado de máquina baseados em Transformers para prever dados de séries temporais.

Tratou das epidemias sazonais de influenza, que resultam em 291.000 a 646.000 mortes anuais em todo o mundo. Os Centros de Controle e Prevenção de Doenças (CDC) publicam relatórios semanais de ILI, porém, geralmente há um atraso de pelo menos uma semana nos relatórios devido à coleta e agregação de dados. Portanto, a previsão da atividade de ILI é fundamental para o monitoramento de doenças em tempo real e para que as agências de saúde pública aloquem recursos para planejar e se preparar para pandemias potenciais.

Coletou dados históricos de ILI em nível nacional e estadual coletados nos EUA de 2010 a 2018 e fornecidos pelo CDC. Usou LSTM, Seq2Seq e SSM. Foram feitos 3 experimentos, de onde concluiu-se que o método se destaca pelo uso de mecanismos de atenção, permitindo capturar dependências complexas nos dados. Essa abordagem serve como um framework versátil para modelar sistemas dinâmicos não lineares, conforme demonstrado no caso da previsão de ILI. Ela é capaz de modelar efetivamente tanto os dados observados de séries temporais quanto o espaço de fase das variáveis de estado usando TDEs.