UNIVERSIDADE DE SÃO PAULO
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E COMPUTAÇÃO - ICMC

Gustavo Bartholomeu Trad Souza
Nº USP – 11219216

Desafio de Séries Temporais

São Paulo 2023

1. Descrição do problema e análise dos dados

O dataset do problema é composto por dois arquivos:

- O arquivo **DailyDelhiClimateTrain.csv** contém dados meteorológicos coletados na cidade de Delhi, na Índia, no período de 4 anos (de 2013 a 2017) e será usado como dataset para treinamento dos modelos de séries temporais.
- O arquivo **DailyDelhiClimateTest.csv** contém dados meteorológicos coletados na cidade de Delhi, na Índia, no período de 4 meses (do ano de 2017) e será usado como dataset para teste dos modelos de séries temporais.

O objetivo é adaptar algoritmos para prever os valores do dataset de teste através do seu treinamento com os valores presentes no dataset de treino.

1.1. Descrição dos dados

Ambos os arquivos possuem as mesmas colunas, sendo elas:

- **date**: Data de coleta do dado no formato YYYY-MM-DD.
- **meantemp**: Temperatura média calculada a partir de múltiplos intervalos de 3 horas em um dia.
- **humidity**: Valor de umidade para o dia (gramas de vapor de água por metro cúbico de volume de ar - g/m³).
- **wind_speed**: Velocidade média do vento em km/h.
- **meanpressure**: Leitura de pressão do clima (medida em atm)

1.2. Preparação do dataset

1.2.1. Valores faltantes:

Ambos os conjuntos de dados não possuem valores ausentes, portanto não será necessário utilizar técnicas de substituição. No caso de haver valores ausentes, uma abordagem adequada seria calcular a média entre os valores imediatamente anteriores e posteriores ao valor ausente. Dessa forma, a relação temporal entre os dados seria levada em consideração, o que não aconteceria se fosse feita uma substituição pela média global, por exemplo.

1.2.2. Outliers:

Os outliers (valores cujo z-score em relação à coluna é maior que 1.5) serão substituídos pela média entre os valores imediatamente anterior e imediatamente posterior àquele valor

caso ambos os valores existam ou pela média da coluna no dataset de treino caso contrário.

1.2.3. Intervalo entre os dados:

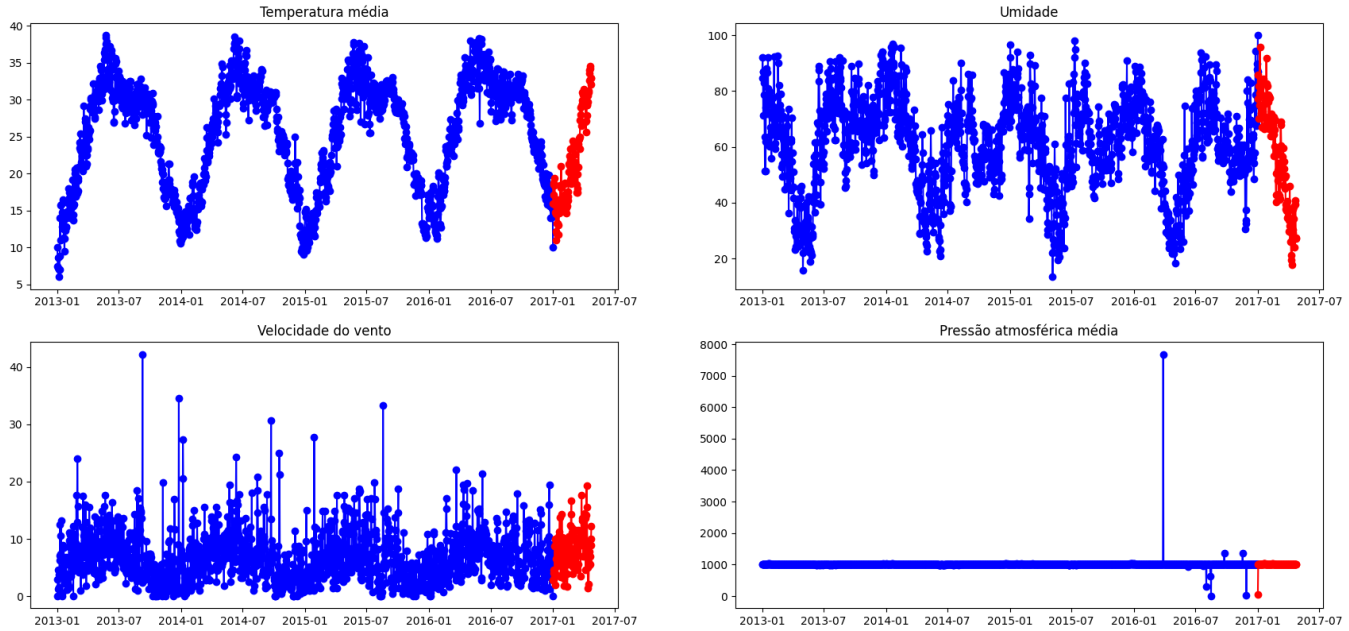
Serão gerados dois conjuntos à partir de cada dataset, um dos conjuntos terá um intervalo de 1 dia entre dados consecutivos (frequência do dataset original) e o outro dataset terá um intervalo de 2 dias (metade dos elementos serão removidos de forma intercalada), a fim de testar se um undersampling pode melhorar no aprendizado uma vez que pode reduzir o ruído e fortalecer tendências e sazonalidades de longo prazo.

1.2.4. Normalização:

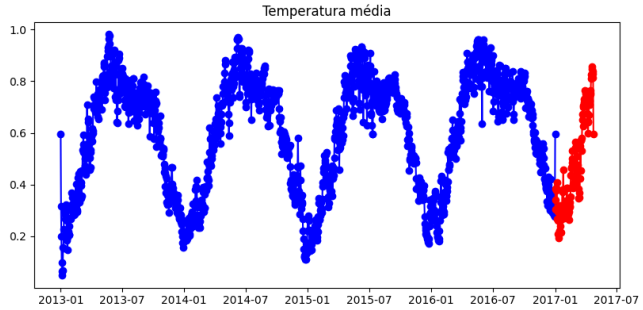
Será realizada uma normalização min-max com os valores máximo e minimo do dataset de treino sendo usado em ambos os datasets, a fim de reduzir a escala dos atributos e facilitar o ajuste dos algoritmos.

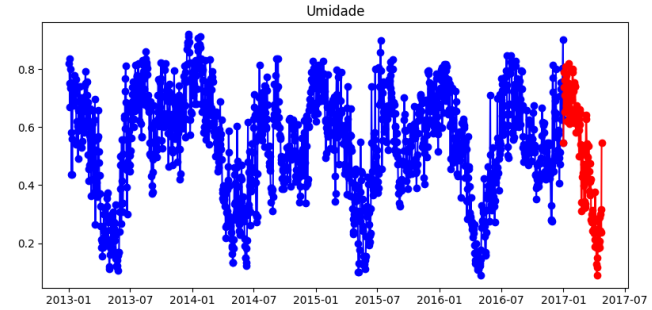
1.3. Datasets

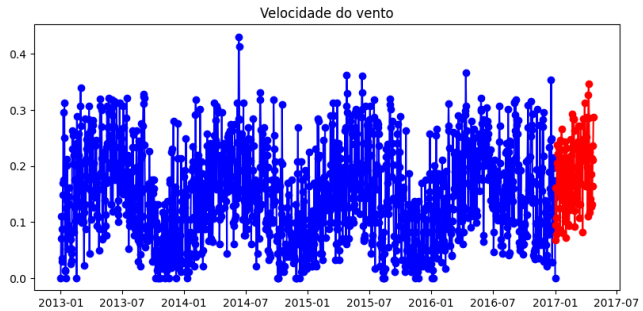
1.3.1 - Dataset original sem tratamento

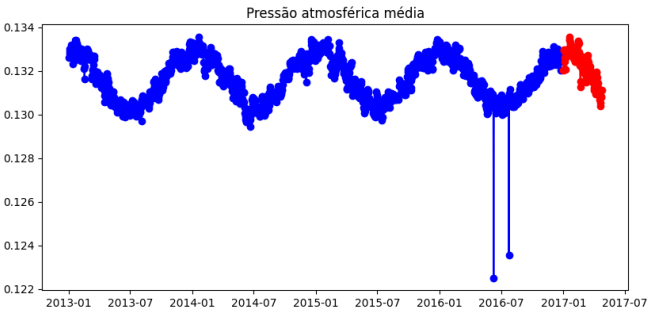


1.3.2 - Dataset tratado com intervalos de 1 dia

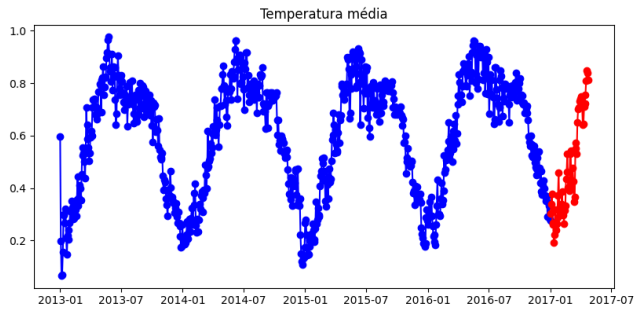


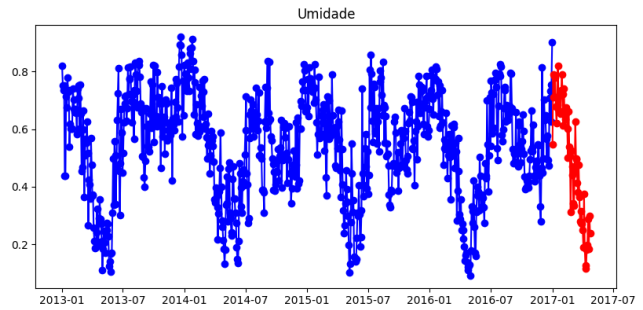


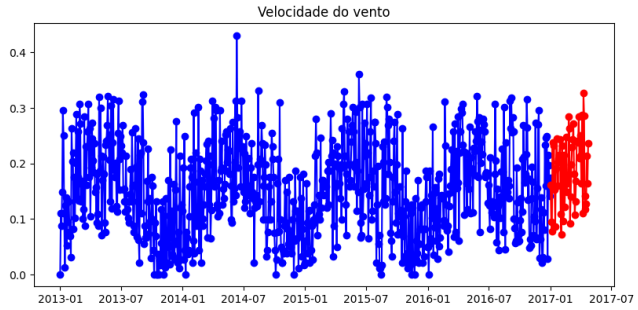


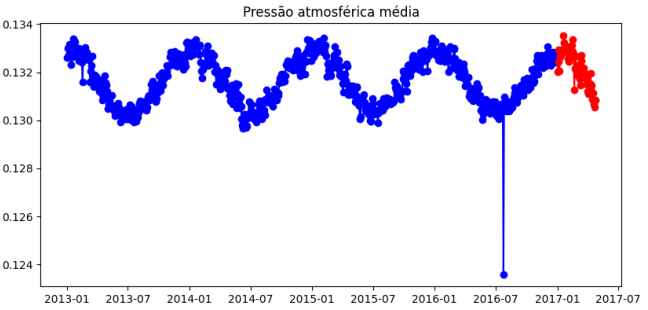


1.3.3. - Dataset tratado com intervalos de 2 dias









2. Descrição das técnicas utilizadas

2.1. ARIMA (Auto-Regressive Integrated Moving Average)

Decompõe a variável observada em componentes de tendência, sazonalidade e irregularidade. Remove a tendência e sazonalidade, permitindo a modelagem da série estacionária usando componentes auto regressivos (AR) e média móvel (MA).

Auto Arima: Método que otimiza o modelo Arima identificando a melhor configuração de seus hiperparâmetros.

Propriedades:

- Uma série temporal é estacionária se suas propriedades (média e variância, por exemplo) não dependem do tempo da observação. Portanto, séries com tendências ou sazonalidades não são estacionárias.
- Autocorrelação é uma medida do relacionamento linear entre os valores dentro de uma mesma série. Assim, se a série é composta por valores aleatórios, a autocorrelação é praticamente nula.

2.2. Exponential Smoothing

Podemos especificar a tendência (trend), a sazonalidade (seasonal) e sua quantidade (sp)

- tendência: direção dos valores da variável em relação ao tempo
- sazonalidade: qualquer mudança ou padrão previsível, ou seja, repetição de comportamento

2.3. Prophet

Prophet é um procedimento para previsão de dados de séries temporais baseado em um modelo aditivo, no qual tendências não lineares são ajustadas com sazonalidade anual, semanal e diária, além de efeitos de feriados. Ele funciona melhor com séries temporais que possuem fortes efeitos sazonais e vários anos de dados históricos. O Prophet é robusto em relação a dados ausentes e mudanças na tendência, e geralmente lida bem com valores discrepantes.

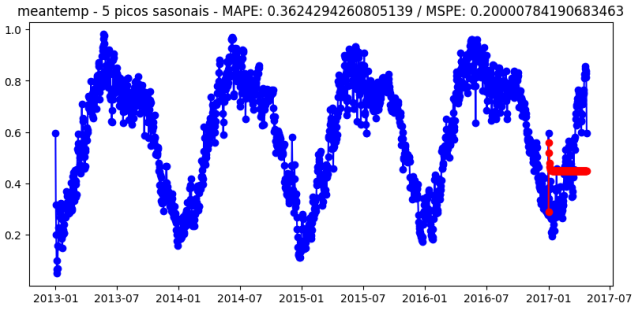
3. Interpretação dos resultados obtidos

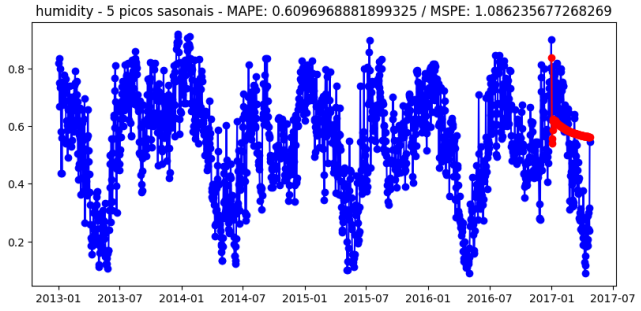
3.1. Resultados

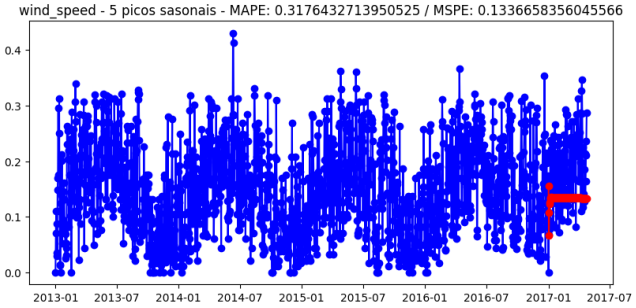
3.1.1. autoARIMA

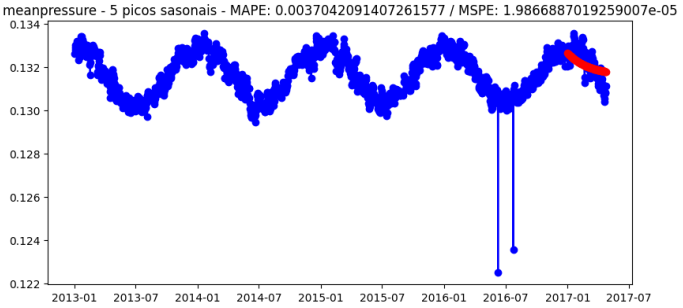
Os resultados obtidos para os três datasets de teste com o algoritmo autoARIMA foram:

- **Frequência: 1 dia:**
  - **meantemp** - MAPE: 0.3624294260805139 / MSPE: 0.20000784190683463
  - **humidity** - MAPE: 0.6096968881899325 / MSPE: 1.086235677268269
  - **wind_speed** - MAPE: 0.3176432713950525 / MSPE: 0.1336658356045566
  - **meanpressure** - MAPE: 0.0037042091407261577 / MSPE: 1.9866887019259007e-05

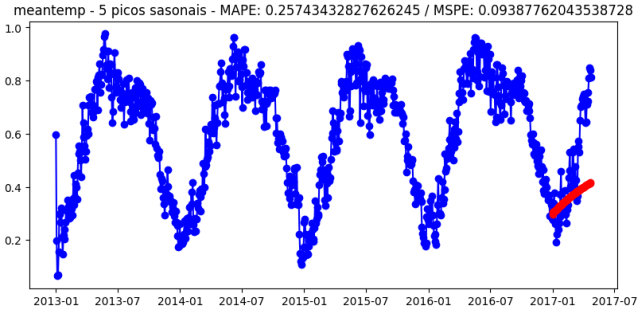


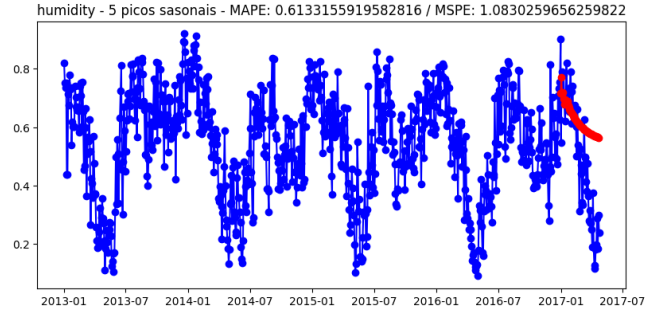


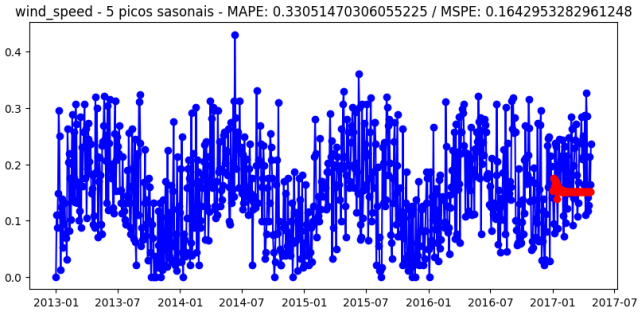


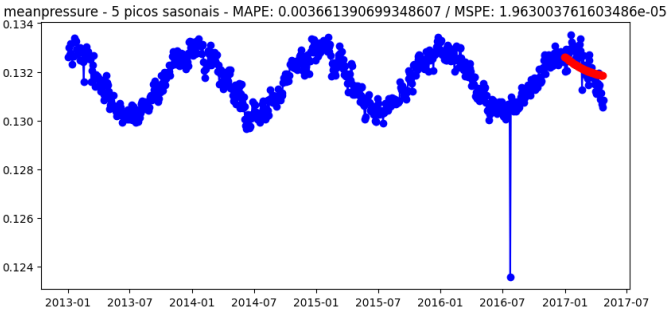


- **Frequência: 2 dias:**
    - **meantemp** - MAPE: 0.25743432827626245 / MSPE: 0.09387762043538728
    - **humidity** - MAPE: 0.6133155919582816 / MSPE: 1.0830259656259822
    - **wind_speed** - MAPE: 0.33051470306055225 / MSPE: 0.1642953282961248
    - **meanpressure** - MAPE: 0.003661390699348607 / MSPE: 1.963003761603486e-05

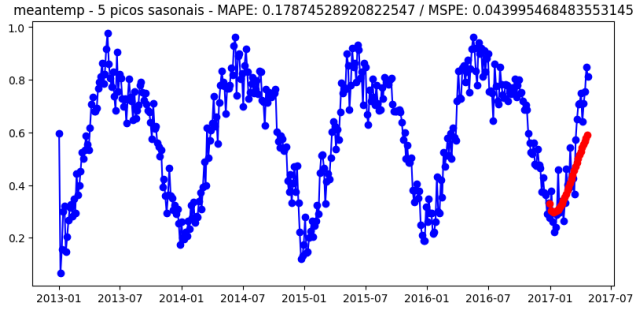


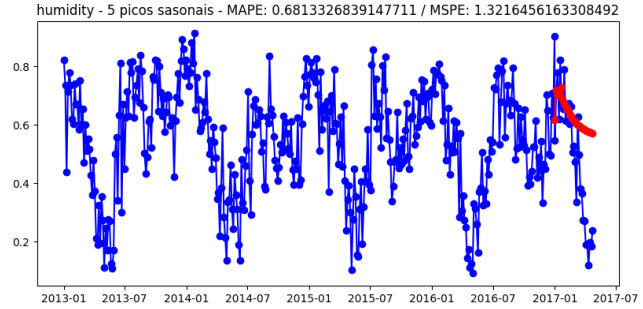


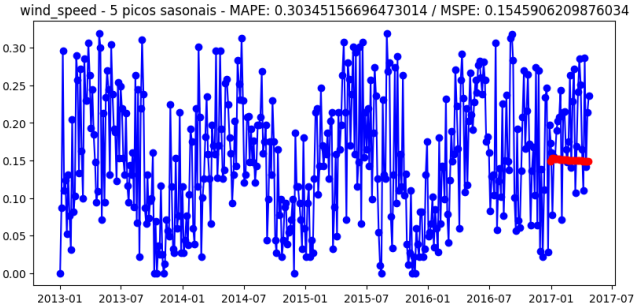


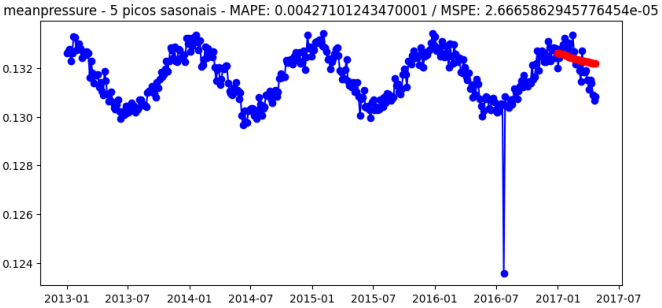


- **Frequência: 4 dias:**
    - **meantemp** - MAPE: 0.17874528920822547 / MSPE: 0.043995468483553145
    - **humidity** - MAPE: 0.6813326839147711 / MSPE: 1.3216456163308492
    - **wind_speed** - MAPE: 0.30345156696473014 / MSPE: 0.1545906209876034
    - **meanpressure** - MAPE: 0.00427101243470001 / MSPE: 2.6665862945776454e-05





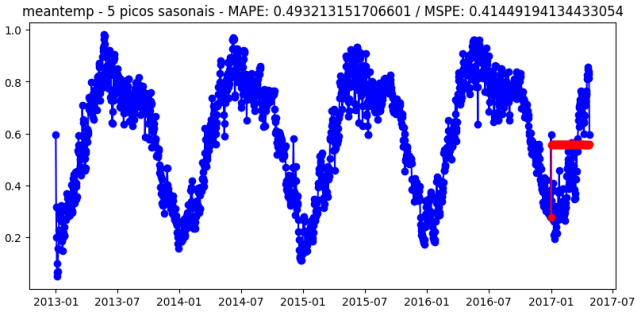


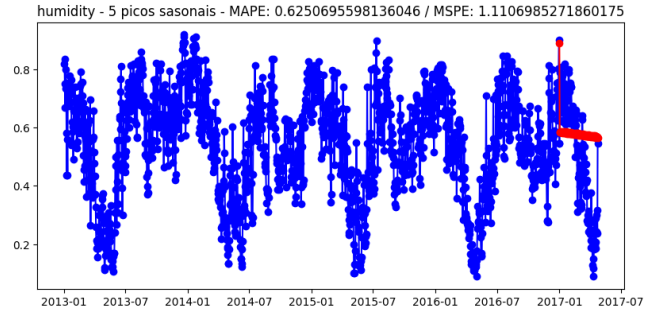


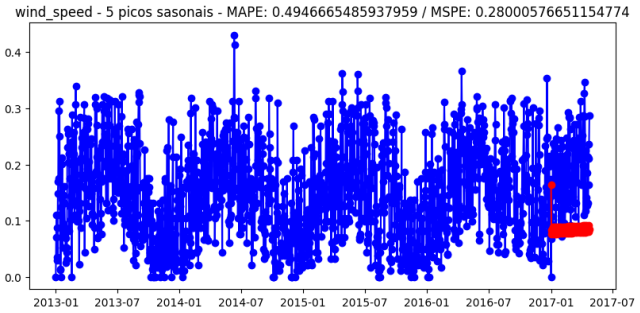
3.1.2. Exponential Smoothing

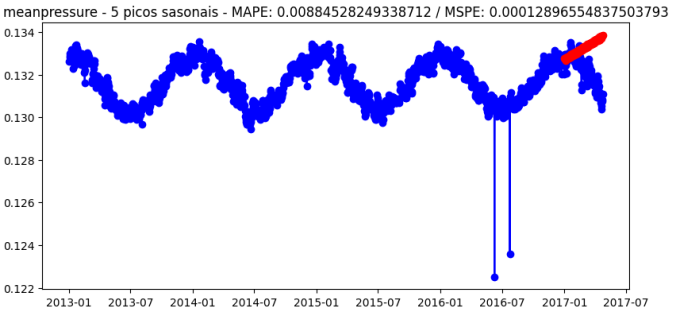
Os resultados obtidos para os três datasets de teste com o algoritmo Exponential Smoothing foram:

- **Frequência: 1 dia:**
    - **meantemp** - MAPE: 0.493213151706601 / MSPE: 0.41449194134433054
    - **humidity** - MAPE: 0.6250695598136046 / MSPE: 1.1106985271860175
    - **wind_speed** - MAPE: 0.4946665485937959 / MSPE: 0.28000576651154774
    - **meanpressure** - MAPE: 0.00884528249338712 / MSPE: 0.00012896554837503793

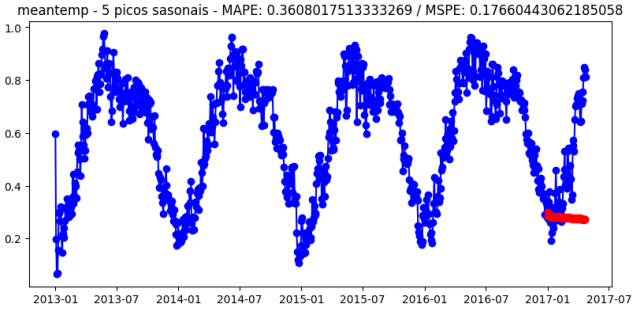


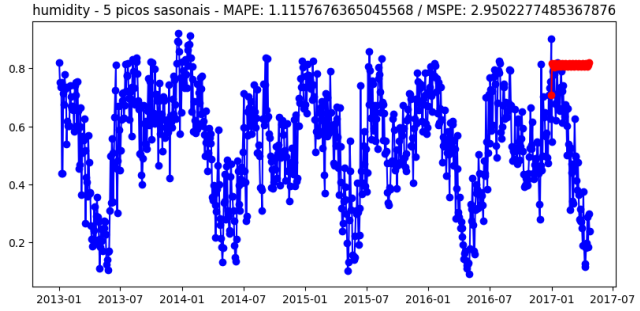


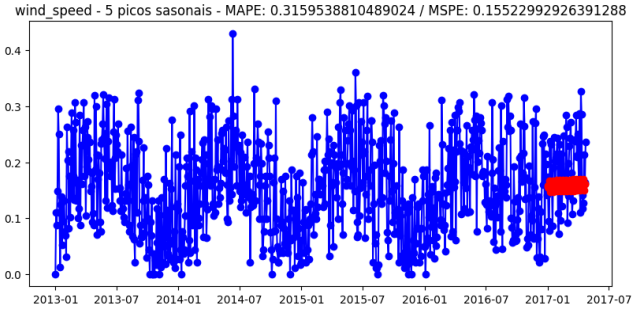


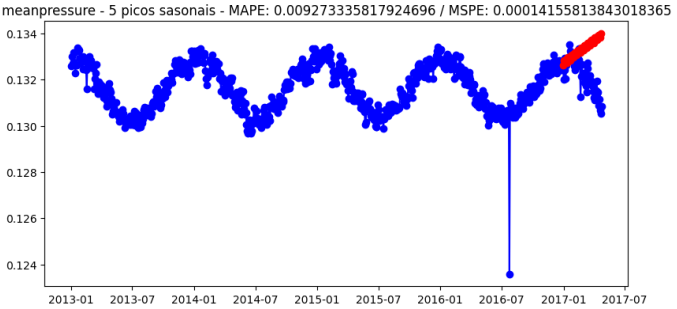


- **Frequência: 2 dias:**
    - **meantemp** - MAPE: 0.3608017513333269 / MSPE: 0.17660443062185058
    - **humidity** - MAPE: 1.1157676365045568 / MSPE: 2.9502277485367876
    - **wind_speed** - MAPE: 0.3159538810489024 / MSPE: 0.15522992926391288
    - **meanpressure** - MAPE: 0.009273335817924696 / MSPE: 0.00014155813843018365

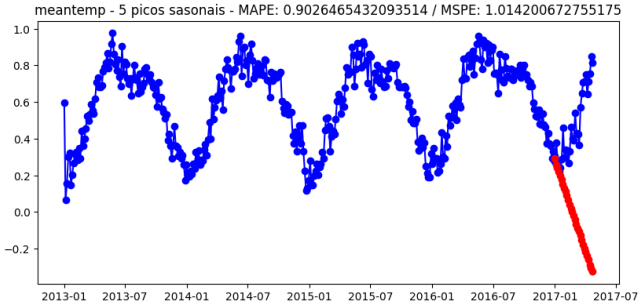


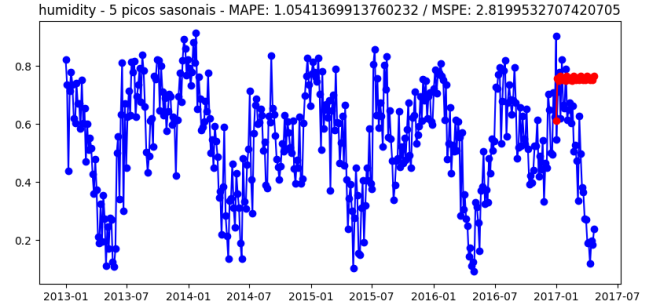


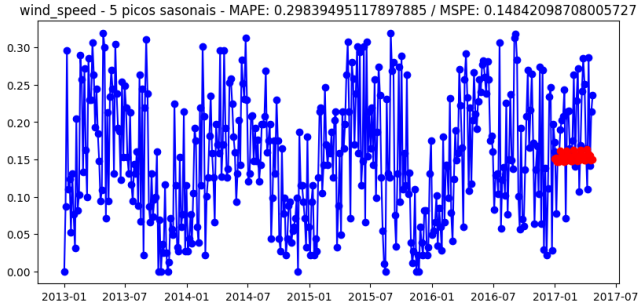


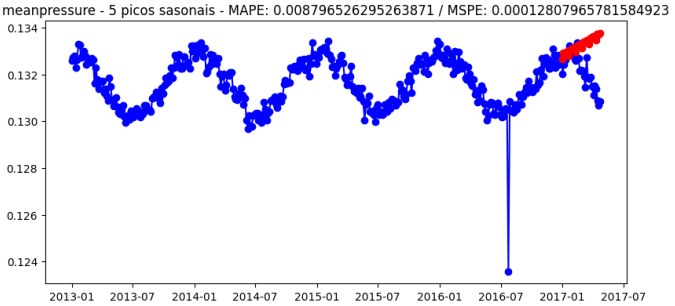


- **Frequência: 4 dias:**
    - **meantemp** - MAPE: 0.9026465432093514 / MSPE: 1.014200672755175
    - **humidity** - MAPE: 1.0541369913760232 / MSPE: 2.8199532707420705
    - **wind_speed** - MAPE: 0.29839495117897885 / MSPE: 0.14842098708005727
    - **meanpressure** - MAPE: 0.008796526295263871 / MSPE: 0.00012807965781584923





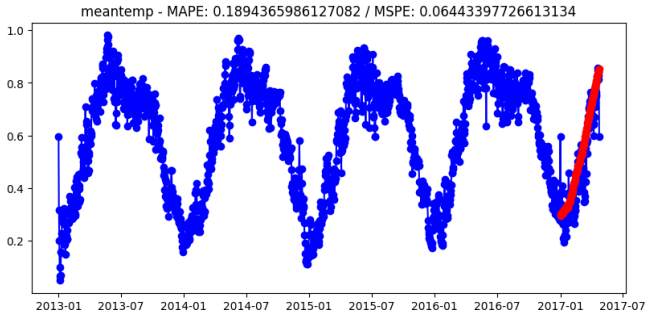


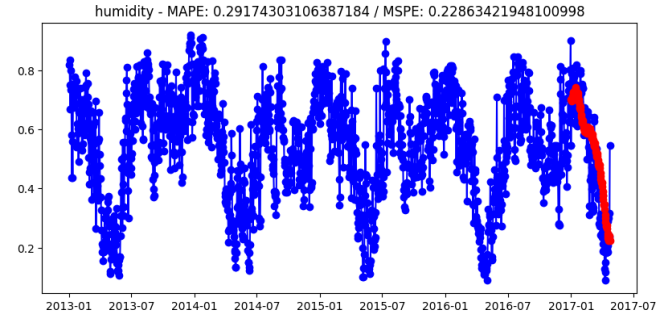


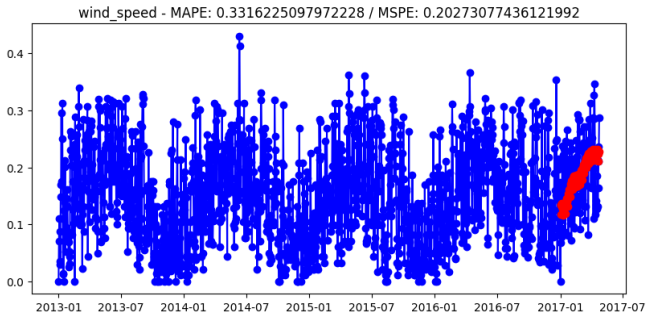
3.1.3. Prophet

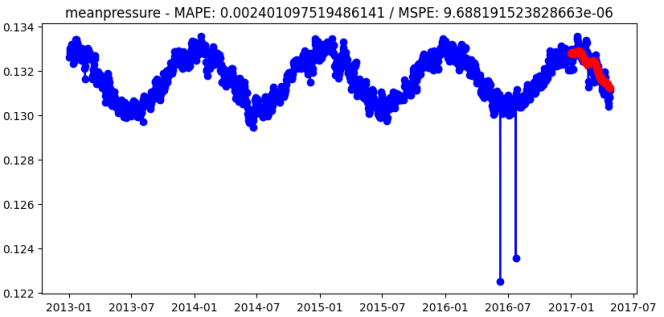
- **Frequência: 1 dia:**

    - **meantemp** - MAPE: 0.1894365986127082 / MSPE: 0.06443397726613134
    - **humidity** - MAPE: 0.29174303106387184 / MSPE: 0.22863421948100998
    - **wind_speed** - MAPE: 0.3316225097972228 / MSPE: 0.20273077436121992
    - **meanpressure** - MAPE: 0.002401097519486141 / MSPE: 9.688191523828663e-06

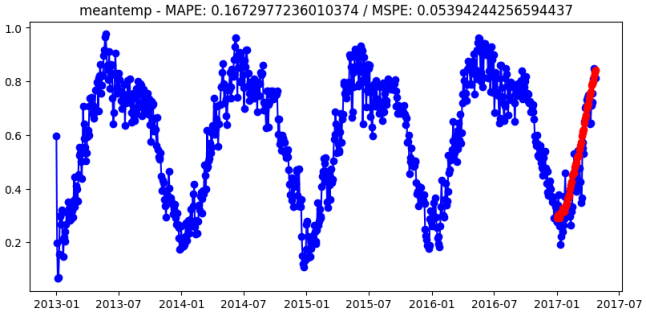


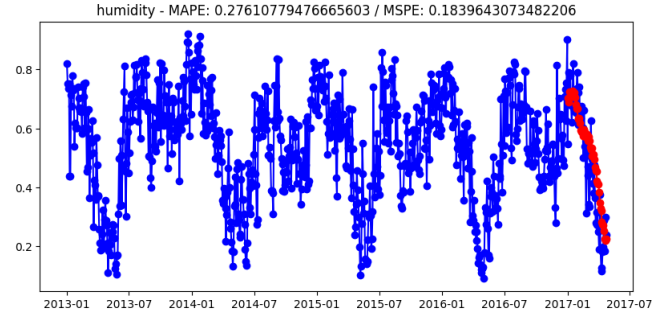


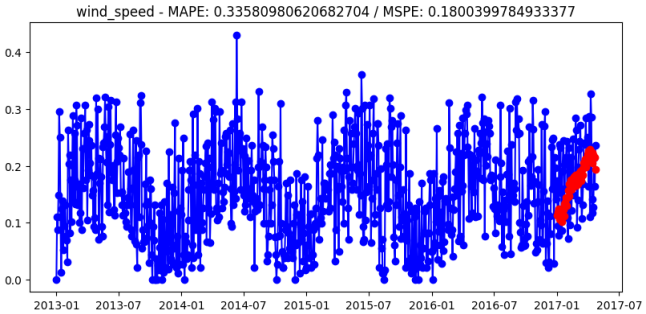


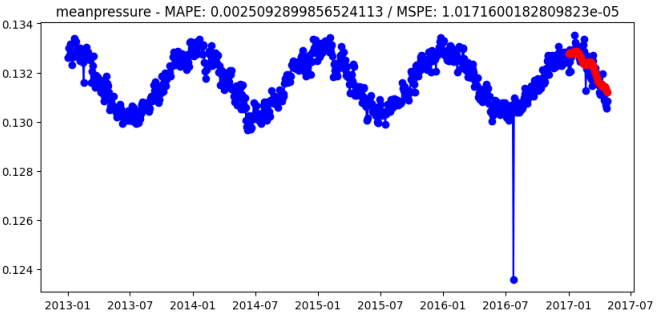


- **Frequência: 2 dias:**
    - **meantemp** - MAPE: 0.1672977236010374 / MSPE: 0.05394244256594437
    - **humidity** - MAPE: 0.27610779476665603 / MSPE: 0.1839643073482206
    - **wind_speed** - MAPE: 0.33580980620682704 / MSPE: 0.1800399784933377
    - **meanpressure** - MAPE: 0.0025092899856524113 / MSPE: 1.0171600182809823e-05

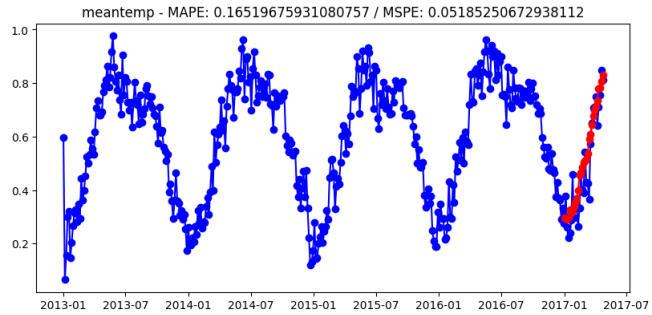


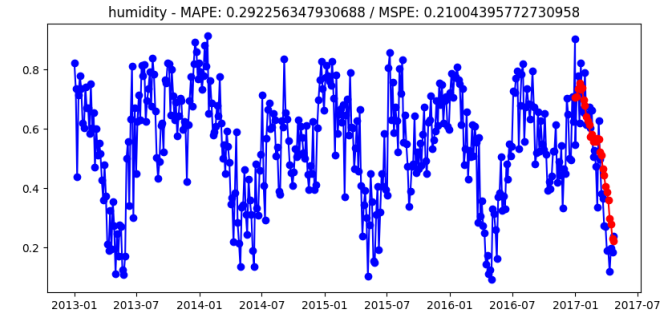


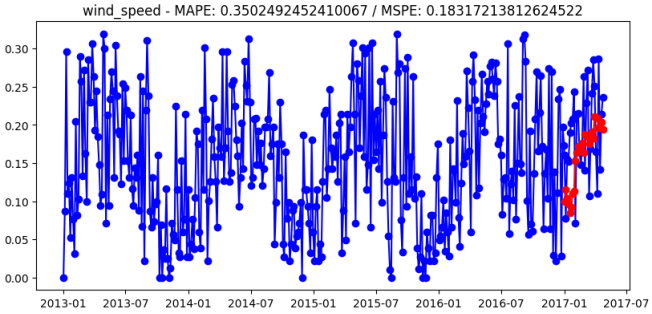


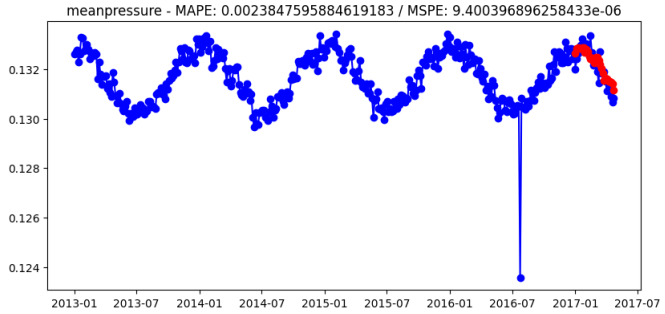


- **Frequência: 4 dias:**
    - **meantemp** - MAPE: 0.16519675931080757 / MSPE: 0.05185250672938112
    - **humidity** - MAPE: 0.292256347930688 / MSPE: 0.21004395772730958
    - **wind_speed** - MAPE: 0.3502492452410067 / MSPE: 0.18317213812624522
    - **meanpressure** - MAPE: 0.0023847595884619183 / MSPE: 9.400396896258433e-06









3.1. Frequência dos dados

O aumento do intervalo entre dados consecutivos gerou uma diminuição dos erros proporcionais em todos os algoritmos, isso pode se dar ao fato de que os dados contém informações climáticas, que medem grandezas com variações à longo prazo, portanto a redução da frequência pode reduzir o ruído de curto prazo e fortalecer as sazonalidades de períodos mais longos.

3.2. Desempenho dos algoritmos

Os algoritmos de autoARIMA e Exponential Smoothing apresentaram resultados bastante inferiores ao algoritmo Prophet. Isso pode ser devido ao número de picos sazonais escolhido que foi inferior ao valor ótimo. O algoritmo autoARIMA obteve melhores resultados porém o seu tempo de execução também é bastante superior, de forma que para valores elevados do número de picos sazonais o algoritmo autoARIMA toma um tempo bastante significativo e consome uma quantidade

excessiva de recursos computacionais, o que pode inviabilizar a sua utilização, fazendo o Exponential Smoothing preferível.

O algoritmo Prophet apresentou o melhor desempenho em questão da sua capacidade de previsão e em relação ao consumo de recursos computacionais, sendo um algoritmo extremamente rápido e preciso.

4. Conclusão

A partir dos dados é seguro concluir que o algoritmo Prophet é o mais recomendado para essa aplicação, pois além de obter resultados excelentes, ele o faz com um baixo custo computacional associado. Os algoritmos de autoARIMA e Exponential Smoothing apresentaram resultados interessantes e, caso fosse aumentado o número de picos sazonais, também poderiam ter apresentado resultados satisfatórios porém o Exponential Smoothing se sobressair em relação ao custo computacional necessário.

5. Referências

Notebook Jupyter - https://colab.research.google.com/drive/1za1Mk-qrq_yTDpI59fpdqzRXpWpxY6qb?usp=sharing