Time Series Analysis using Facebook Prophet

Last Updated : 21 Nov, 2024

Time series analysis is one of the important methodologies that helps us to understand the hidden patterns in a dataset that is too related to the time at which it is being recorded. The article aims to explore the fundamentals of time series analysis and demonstrates the **analysis using Facebook Prophet**.

Table of Content

What is Time Series Analysis?

Time series analysis is a statistical approach that entails gathering data at consistent intervals to recognize patterns and trends. This methodology is employed for making well-informed decisions and precise forecasts by leveraging insights derived from historical data.

And the process of predicting the future values of the data by analyzing the previous trends and patterns hidden in the data is known as time series forecasting. Time series forecasting can be done using various forecasting techniques like ARIMA, SARIMA, Prophet, Theta and other statistical method. Time series data is a sequential data hence, deep learning-based methods like RNN, LSTM, BLSTM and GRU are also used for time series forecasting.

Key components of Time Series analysis

Grasping these dependencies is essential for accurate predictions and gaining valuable insights.

2. Trend analysis involves recognizing the fundamental direction or long-term movement in a time series, whether it is upward, downward, or stable. This understanding provides insights into the overall trajectory of the data.
3. Seasonal patterns, observed in numerous time series, depict recurring cycles at fixed intervals, such as daily, monthly, quarterly, or yearly. Analyzing these patterns through seasonal analysis helps identify and comprehend their periodic nature.
4. Time series data often incorporates random fluctuations or noise that lacks a specific trend or pattern. It is crucial to eliminate this noise to unveil meaningful information within the data.

Facebook Prophet Library

Prophet is an open-source tool from Facebook used for forecasting time series data which helps businesses understand and possibly predict the market. It is based on a decomposable additive model where non-linear trends fit with seasonality, it also takes into account the effects of holidays. Before we head right into coding, let's learn certain terms that are required to understand this.

- **Trend:** The trend shows the tendency of the data to increase or decrease over a long period of time and it filters out the seasonal variations.
- **Seasonality:** Seasonality is the variations that occur over a short period of time and is not prominent enough to be called a "trend".

Understanding the Prophet Model

The general idea of the model is similar to a generalized additive model. The "Prophet Equation" fits, as mentioned above, trends, seasonality, and holidays. This is given by,

$$y(t) = g(t) + s(t) + h(t) + e(t)$$

- ***h(t)*** *refers to effects of holidays to the forecast*
- ***e(t)*** *refers to the unconditional changes that is specific to a business or a person or a circumstance. It is also called the error term.*
- ***y(t)*** *is the forecast.*

Need of Facebook Prophet

We need it because, although the basic decomposable additive model looks simply, the calculation of the terms within is hugely mathematical. If you do not know what you are doing, it may lead to making wrong forecasts, which might have severe repercussions in the real world. So, to automate this process, we are going to use Prophet. However, to understand the math behind this process and how Prophet actually works, let's see how it forecasts the data.

Prophet provides us with two models (however, newer models can be written or extended according to specific requirements).

1. **Logistic Growth Model**
2. **Piece-Wise Linear Model**

By default, Prophet uses a piece-wise linear model, but it can be changed by specifying the model. Choosing a model is delicate as it is dependent on a variety of factors such as company size, growth rate, business model, etc., If the data to be forecasted, has saturating and non-linear data (grows non-linearly and after reaching the saturation point, shows little to no growth or shrink and only exhibits some seasonal changes), then logistic growth model is the best option. Nevertheless, if the data shows linear properties and had growth or shrink trends in the past then, the piece-wise linear model is a better choice. The logistic growth model is fit using the following statistical equation,

$$g(t) = \frac{C}{1+e^{-k(t-m)}}$$

where,

The piece-wise linear model is fit using the following statistical equations,

$$y = \begin{cases} \beta_0 + \beta_1 x, & \text{if } x \leq c, \\ \beta_0 - \beta_2 c + (\beta_1 + \beta_2) x, & \text{if } x > c. \end{cases}$$

where c is the trend change point (it defines the change in the trend).? is the trend parameter and can be tuned as per requirement for forecasting.

Implementation – Analyzing Time Series Data using Prophet

Now let's try and build a model that is going to forecast the number of passengers for the next five years using time series analysis.

Installing Prophet and Other Dependencies

Install pandas for data manipulation and for the dataframe data structure.

```
!pip install pandas
```

Install Prophet for time series analysis and forecasting.

```
!pip install prophet
```

Importing Required Libraries

```
import pandas as pd
from prophet import Prophet
from prophet.plot import add_changepoints_to_plot
```

Loading Air Passenger Dataset

Now let's load the csv file in the pandas data frame. The dataset contains the number of air passengers in the USA from January 1949 to December 1960.

```
        "/master/airpassengers.csv")
data = pd.read_csv(url)
data.head()
```

Output:

```
     Month  #Passengers
0  1949-01          112
1  1949-02          118
2  1949-03          132
3  1949-04          129
4  1949-05          121
```

Facebook Prophet predicts data only when it is in a certain format. The dataframe with the data should have a column saved as *ds* for time series data and *y* for the data to be forecasted. Here, the time series is the column *Month* and the data to be forecasted is the column *#Passengers*. So, let's make a new DataFrame with new column names and the same data. Also, *ds* should be in a DateTime format.

```
df = pd.DataFrame()
df['ds'] = pd.to_datetime(data['Month'])
df['y'] = data['#Passengers']
df.head()
```

Output:

```
           ds    y
0 1949-01-01  112
1 1949-02-01  118
2 1949-03-01  132
3 1949-04-01  129
4 1949-05-01  121
```

Initializing a Prophet Model

```
m = Prophet()
m.fit(df)
```

We want our model to predict the next 5 years, that is, till 1965. The frequency of our data is 1 month and thus for 5 years, it is 12 * 5 = 60 months. So, we need to add 60 to more rows of monthly data to a dataframe.

```
future = m.make_future_dataframe(periods=12 * 5,
                                 freq='M')
```

Now in the *future* dataframe we have just *ds* values, and we should predict the *y* values.

```
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower',
          'yhat_upper', 'trend',
          'trend_lower', 'trend_upper']].tail()
```

Output:

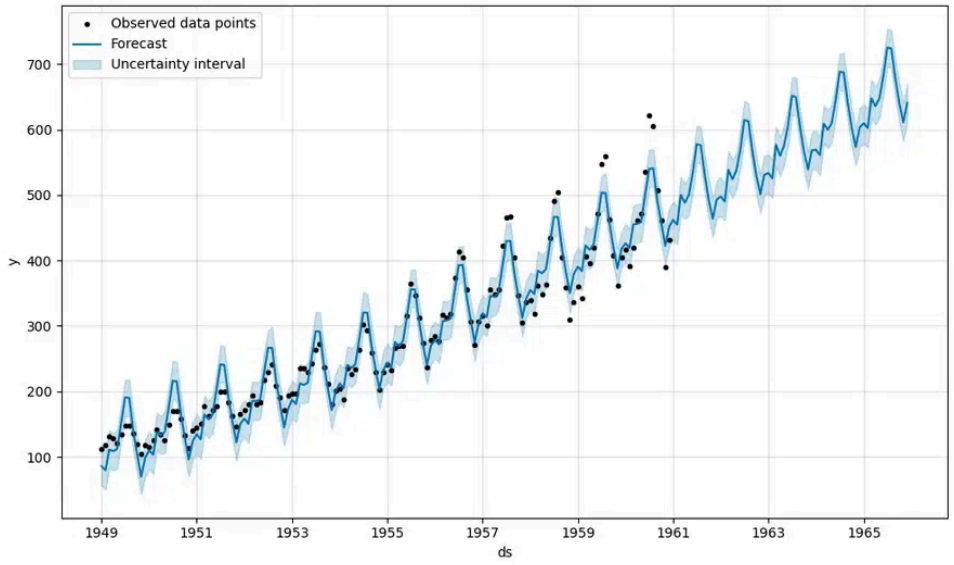
```
            ds        yhat  yhat_lower  yhat_upper       trend
trend_lower  \
199 1965-07-31  723.847886  695.131427  753.432671  656.874802
649.871409
200 1965-08-31  677.972773  649.074148  707.203918  660.006451
652.869703
201 1965-09-30  640.723643  612.025440  670.377970  663.037079
655.722042
202 1965-10-31  610.965273  580.772823  641.770085  666.168728
658.710202
203 1965-11-30  640.594175  611.016447  669.582208  669.199356
661.513728

     trend_upper
199   663.255178
```

Plotting the Forecast Data

Table ds, as we know, is the time series data. *yhat* is the prediction, *yhat_lower,* and *yhat_upper* are the uncertainty levels (it basically means the prediction and actual values can vary within the bounds of the uncertainty levels). Next up we have a *trend that* shows the long-term growth, shrink, or stagnancy of the data, *trend_lower,* and *trend_upper* is the uncertainty levels.

```
fig1 = m.plot(forecast, include_legend=True)
```

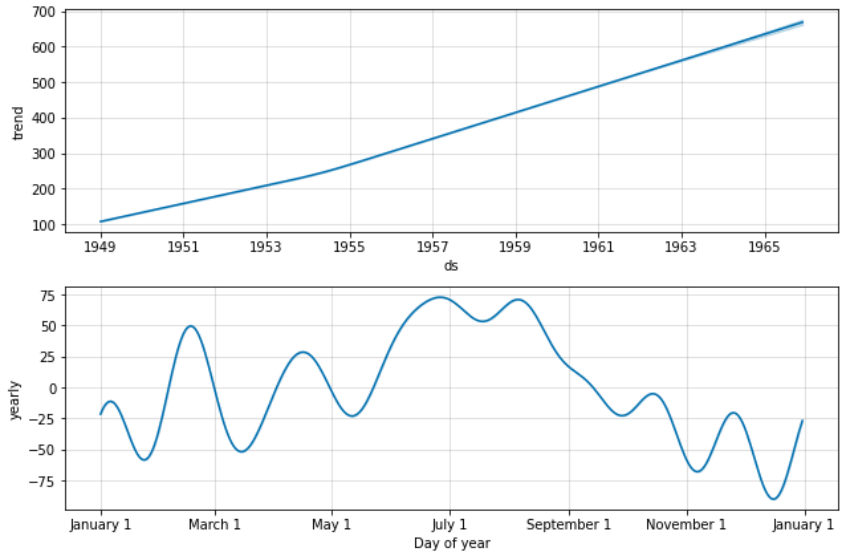
Output:



Time Series Forecasting

The below image shows the basic prediction. The light blue is the uncertainty level(*yhat_upper* and *yhat_lower*), the dark blue is the prediction(*yhat*) and the black dots are the original data. We can see that the predicted data is very close to the actual data. In the last five years, there is no “actual” data, but looking at the performance of our model in years where data is available it is

Output:

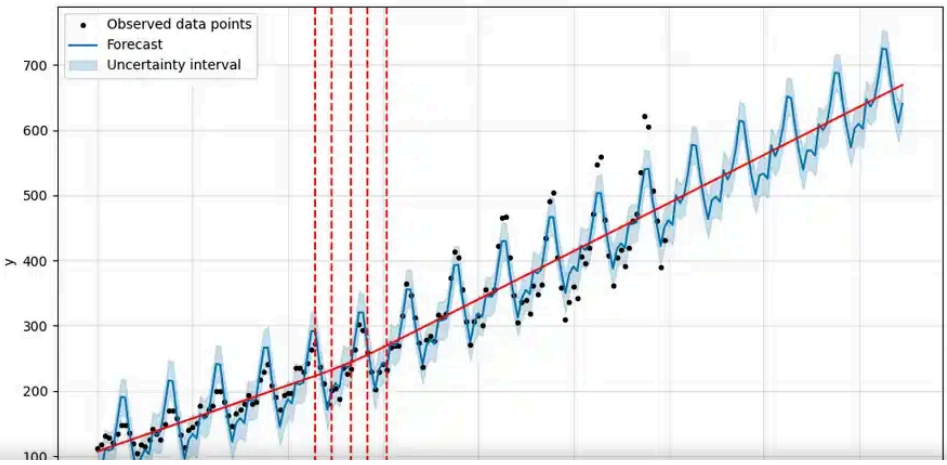


Trends and Seasonality in Time Series Data

The below images show the trends and seasonality (in a year) of the time series data. We can see there is an increasing trend, meaning the number of air passengers has increased over time. If we look at the seasonality graph, we can see that June and July is the time with the most passengers in a given year.

```
fig = m.plot(forecast)
a = add_changepoints_to_plot(fig.gca(),
                             m, forecast)
```

Output:



Add changepoints to indicate the time in rapid trend growths. The dotted red lines show the time when there was a rapid change in the trend of the passengers. Thus, we have seen how we can design a prediction model using Facebook Prophet with only a few lines of code which would have been very difficult to implement using traditional machine learning algorithms and mathematical and statistical concepts alone.

Next Article

Improving Business Decision-Making using Time Series

Similar Reads

Time Series Analysis using Facebook Prophet in R Programming

Time Series Analysis is a way of analysing and learning the behaviour of datasets over a period. Moreover, it helps in learning the behavior of the dataset by plotting the time series object on the graph. In R...

3 min read

Time Series Analysis and Forecasting

Time series analysis and forecasting are crucial for predicting future trends, behaviors, and behaviours based on historical data. It helps businesses make informed decisions, optimize resources, and mitigate risks by...

15+ min read

Improving Business Decision-Making using Time Series

Time series analysis is mainly used to explain, describe, and predict changes via the time of chosen variables. Many companies use time series forecasting, and analysis to develop business strategies. These techniques...

6 min read

6 min read

Active Product Sales Analysis using Matplotlib in Python

Every modern company that engages in online sales or maintains a specialized e-commerce website now aims to maximize its throughput in order to determine what precisely their clients needÂ in order to increase their...

3 min read

Time Series Datasets

Time series datasets are a crucial component of data science and analytics, especially in fields where understanding trends, patterns, and temporal dynamics is essential. A time series is a sequence of data point...

2 min read

Step by Step Predictive Analysis - Machine Learning

Predictive analytics involves certain manipulations on data from existing data sets with the goal of identifying some new trends and patterns. These trends and patterns are then used to predict future outcomes and...

3 min read

Real -Time Analytics in big data

Real-time analytics in Big Data provides the ability to extract useful insights quickly from massive datasets. Real-time analytics stands at the forefront of this transformation, enabling organizations to analyze data...

10 min read

Stock Price Analysis With Python

Python is a great language for making data-based analyses and visualizations. It also has a wide range of open-source libraries that can be used off the shelf for some great functionalities. Python Dash is a library th...

3 min read

Model Planning for Data Analytics

In this article, we are going to discuss model planning for data analytics in which we will cover all procedural steps one by one. Model planning is phase 3 of lifecycle phases of data analytics, where team determines...

3 min read

Registered Address:
K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305