Sales Forecasting using FB-Prophet

Albert Devasia
Department of Computer Application
Amal Jyothi College of Engineering
Kottayam, Kerala
albertdevasia2024@mca.ajce.in

Navyamol K.T
Department of Computer Applications
Amal Jyothi College of Engineering
Kottayam, India
navyamolkt@amaljyothi.ac.in

***Abstract*— Forecasting is a common task in data science, helping organizations plan their resources, set goals, and spot unusual trends. But making accurate forecasts isn't always easy. One big challenge is that not many analysts are experts in time series modeling. Using Fb- Prophet we can resolve the issue. It uses flexible models and gets analysts involved to improve the accuracy of forecasts. We've also created a model that's easy to understand and adjust, even for analysts who aren't time series experts. With this approach, anyone can analyze time series data effectively.**

Keywords- Time-series, Prophet, Generalized Additive Model, ARIMA, Auto-ARIMA Model, Error Trend Seasonality model, S-Naive model

I. Introduction

Sales forecasting is an essential tool for businesses of all types. It offers crucial information about future revenue, enabling businesses to plan for growth, allocate resources effectively, and make data-driven decisions. By accurately predicting sales trends, businesses can remain agile and adapt to changes in the market, align their strategies with customer needs, and manage inventory efficiently.

Time series analysis is a powerful tool for businesses to uncover hidden insights and trends in their sales data. By understanding historical sales patterns and identifying seasonal fluctuations, businesses can make informed decisions about inventory, marketing, and resource allocation. Time series forecasting, a key part of this analysis, allows businesses to predict future sales based on historical data. This is especially important for businesses in dynamic markets with fluctuating demand. Generating accurate forecasts is a challenging task, both for automated systems and many analysts. Two key aspects when it comes to creating business forecasts. First, completely automated forecasting methods can be tricky to fine-tune, and they often lack the flexibility to consider important assumptions or practical insights. Second, the analysts responsible for data-related tasks in organizations often have a deep understanding of the specific products or services they deal with. However, they might not have formal training in time series forecasting. Skilled analysts who can produce high-quality forecasts are somewhat rare because forecasting requires specialized expertise and substantial experience. Among various sales forecasting techniques, time series modeling, particularly using the Prophet model, is the best. This approach analyzes historical sales data, identifies trends and seasonality, and forecasts future sales with high accuracy. The Prophet model's ability to handle multiple seasonality patterns, including yearly, weekly, and daily trends, makes it particularly suitable for retail and e-commerce businesses.

Based on the different practices and studies done on Prophet and other models it is clear that prophet is an easy tool that can be used by analysts and beginners as analyst. The paper mainly focuses on studies and experiments conducted.

II. Related Works

Research on Sales Forecast Based on Prophet-SARIMA by Jiantao Zhao, Chunwei Zhang. [1] Analyze and compare the prediction effect of single model and combination model.The first mock exam shows that the Prophet-SARIMA combination model has higher accuracy and stability in sales volume forecasting than single model. only the combination model which is weighted by two single models is realized, and different combination methods can be designed in the future. The sales volume forecast results of Prophet SARIMA combination model can provide an important basis for the e-commerce to supplement orders scientifically.

Machine Learning Approach for Forecasting the Sales of Truck Components by Venishetty Sai Vineeth. [2] This research focuses on using machine learning algorithms to forecast truck component sales, optimizing delivery time, stock maintenance, and decision-making processes in various areas. Support Vector Machine Regression, Ridge Regression, Gradient Boosting Regression, and Random Forest Regression were considered, with Ridge Regression outperforming the others in sales forecasting for truck components. Ridge Regression was selected as the optimal algorithm for this purpose.

Published in The IEBM Encyclopedia of Marketing, Michael J. Baker (Ed.), International Thompson Business Press, 1999, p. 278-290. [3] importance of sales forecasting in various fields of scientific, industrial, commercial, and economic activity. The paper provided an overview of forecasting methods, direct extrapolation of sales, causal approaches to sales forecasting, new product forecasting, evaluating and selecting methods, estimating prediction intervals, implementation, and conclusions.

III. Methodology

Keep The aim of the system is to analyse and forecast the future sales using Prophet. Model is trained in collab and using Flask UI is created for future forecast

A. *Data collection*

This dataset comprises a comprehensive record of 1461 days data. purchases spanning four years, from 2019-01-01 to 2022-12-30. These data points hold the key to understanding the sales trends, seasonality, and performance of the products throughout this four-year period.

B. *Data Preprocessing*

The dataset contained a number of factors, some of which were either useless or had negligible effects on product demand.

Steps involved:-

- Dropping unwanted columns.
- Checking for null values and dropping them.
- Converting the data type and sorting the data based on date.
- Grouping and summarizing the values based on the date

As a result, only the Date and the Sales (sales of the day) were kept. Here the daily data is enough inorder to get monthly sales, yearly sales forecasting and also to predict the future.

Train data is taken upto 70 % that is upto 1095 rows

	Date	Sales
0	2019-01-01	4.851600e+04
1	2019-01-02	7.569659e+04
2	2019-01-03	1.295617e+04
3	2019-01-04	7.673976e+04
4	2019-01-05	7.946035e+04
...	...	...
1091	2021-12-27	1.130906e+06
1092	2021-12-28	1.180851e+06
1093	2021-12-29	1.141792e+06
1094	2021-12-30	1.130377e+06
1095	2021-12-31	1.154373e+06

Fig.1. Sample Data

Balance is taken as test data that is 30%.

C. *Forecasting*

Prophet employs an additive model that decomposes a time series into three main components: trend, seasonality, and holidays. The trend component represents long-term changes, the seasonality component captures periodic patterns, and holidays account for special events that can affect the time series.

3 main components of Prophet model is Seasonality, Trend and Holidays. It can be represented in the equation

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t.$$

where,

Y(t) is the observed value of the time series at time t

g(t) is the trend function which models non-periodic changes in the value of the time series

s(t) is the seasonal component at time t. that is yearly or weekly seasonality periodic changes

h(t) is the holiday effect. Potential irregular schedules over one or more days.

ϵ_t represents any idiosyncratic changes

which are not accommodated by the model; later we will make the parametric assumption that it is normal distribution.

The Prophet uses Generalized Additive Model (GAM). GAM is a regression model that accommodates non-linear relationships. Unlike traditional time series models, such as ARIMA, which explicitly capture temporal dependencies, GAM takes a different approach. It dissects time series into additive components, like trends and seasonality, similar to how exponential smoothing handles seasonality. These components serve as fundamental building blocks, making the data more understandable and adaptable for forecasting.

ARIMA, or AutoRegressive Integrated Moving Average, relies on a mathematical framework involving auto-regressive (AR) and moving average (MA) components. These components capture dependencies on previous observations and the white noise errors of the past. To apply ARIMA effectively, one must identify crucial elements: the order of differencing, denoted as integration, the count of auto-regressive terms (p), and the number of moving average terms (q). The model is then fitted through the estimation of parameters that aim to minimize discrepancies between the model's predictions and the actual observed data. To achieve this, various numerical optimization techniques are employed. ARIMA is typically better suited for experienced users who possess familiarity with time series analysis and statistical modeling. In auto-ARIMA fits a range of ARIMA models and selects the best one. the problem is it deals only with seasonal data and sudden changes affect it. That is change in seasonality in last moments.

The ETS model, standing for Error, Trend, and Seasonality, is a statistical forecasting approach that decomposes time series data into three fundamental components. The error component handles random fluctuations and unexplained variations within the data. Meanwhile, the trend component captures the underlying direction or pattern, which may involve growth, decline, or stability over time. The seasonality component deals with regular, repetitive patterns that occur at fixed intervals, providing insights into recurring cycles in the data. This model is valuable for understanding and forecasting time series data by breaking it down into these distinct elements.

The seasonal naive (SNaive) method is a simple forecasting approach that takes into account seasonality in time series data. It primarily focuses on the most recent seasonal observation from the previous year as the forecast for the current period. The SNaive method is particularly useful when dealing with data that exhibits strong seasonal patterns.

Prophet, a tool developed by Facebook presents an approach, to forecasting time series data. It addresses the limitations of methods like ARIMA, Auto ARIMA, ETS and Seasonal Naive. While ARIMA and Auto ARIMA struggle with patterns due to their assumption of linearity Prophet is specifically designed to handle data effectively. ETS models are suitable for capturing patterns but may not perform well when faced with complex or irregular seasonality. On the hand Prophet automatically. Adapts to seasonality making it more user friendly and robust in capturing complex seasonal patterns. Additionally it offers forecasts that can incorporate holidays and special events. A feature that can be challenging when using ETS models. However Seasonal Naive is a model that assumes seasonal patterns will exactly repeat themselves. This assumption may lead to inaccurate forecasts in cases where strict adherence to seasonality doesn't hold true. Prophet's flexibility in handling seasonal trends and adaptability to various situations make it a more versatile choice than Seasonal Naive. This made the choice to select the Prophet

D. *Model*

Model is creating using prophet. Training and Testing is done using the sales_data.csv. Before that we have to convert the dates from string or any other datatype to date datatype. For that the following code is given

```
df['ds']=pd.to_datetime(df['ds'])
```

Fig.2. Date is converted to ds according to prophet

After that the data which is given for training and testing is given

```
train = df.iloc[:len(df)-365]
test = df.iloc[len(df)-365:]
```

Fig.3 Train and Test

Afetr this part model is fit to prophet. For this the foloowing code is given

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

Fig.4 Fitting of model to prophet

The thing we should remember is that the date should be in ds and the sales data should be y column. that is the name of the column should be y. this is a important thing in fbprophet.

```
future = m.make_future_dataframe(periods=365)
forecast = m.predict(future)
```

Fig.5 Making future forecast

this is given to predict the future based on the trained model.

```
forecast.tail()
```

Fig.6 view the last data

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms
1821	2023-12-27	1.881381e+06	1.846825e+06	1.912088e+06	1.877816e+06	1.884793e+06	-2036.412986
1822	2023-12-28	1.882384e+06	1.847688e+06	1.915667e+06	1.878818e+06	1.885816e+06	-250.971308
1823	2023-12-29	1.883387e+06	1.849734e+06	1.917309e+06	1.879833e+06	1.886838e+06	551.091164
1824	2023-12-30	1.884390e+06	1.847938e+06	1.913487e+06	1.880822e+06	1.887861e+06	-2768.576830
1825	2023-12-31	1.885394e+06	1.851096e+06	1.916734e+06	1.881821e+06	1.888883e+06	-3078.133899

Fig.6 Data received after predictions. this columns are given by prophet by default

additive_terms_lower	additive_terms_upper	weekly	weekly_lower	weekly_upper	yearly
-2036.412986	-2036.412986	-822.850079	-822.850079	-822.850079	-1213.562907
-250.971308	-250.971308	1217.412708	1217.412708	1217.412708	-1468.384017
551.091164	551.091164	2258.732856	2258.732856	2258.732856	-1707.641692
-2768.576830	-2768.576830	-838.166615	-838.166615	-838.166615	-1930.410215
-3078.133899	-3078.133899	-942.316347	-942.316347	-942.316347	-2135.817552

Fig.7. Data received after predictions. this columns are given by prophet by default

here yhat, yhat upper and yhat lower is only taken. Other all datas are removed. After that the data will be like these. the data only contains the end part

```
forecast[['ds','yhat','yhat_lower','yhat_upper']].tail()
```

	ds	yhat	yhat_lower	yhat_upper
1821	2023-12-27	1.879345e+06	1.846825e+06	1.912088e+06
1822	2023-12-28	1.882133e+06	1.847688e+06	1.915667e+06
1823	2023-12-29	1.883938e+06	1.849734e+06	1.917309e+06
1824	2023-12-30	1.881622e+06	1.847938e+06	1.913487e+06
1825	2023-12-31	1.882315e+06	1.851096e+06	1.916734e+06

Fig.8 Collecting only needed data

and we plotted it we will get the graph like these

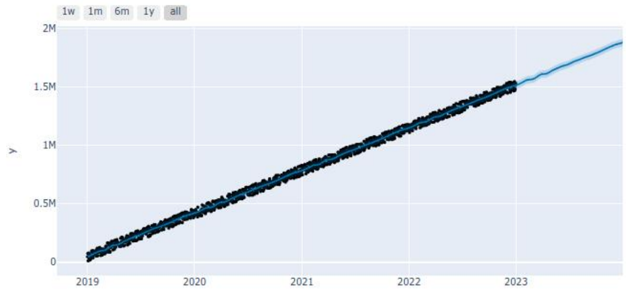

Fig.9 Plotting predicted and model data

here we can get weekly, monthly, 6 months, 1 year or whole data is received. we can see it according to our wish. This is the major advantage of fbprophet. we can also view the trend, yearly fluctuations, and weekly fluctuations.

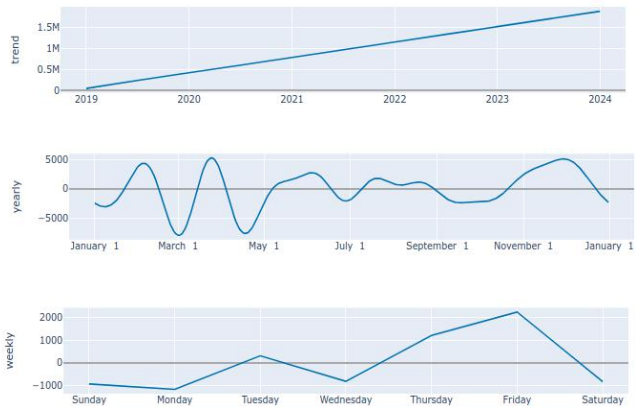

Fig.10 Plotting Trend, yearly and weekly data

IV. Cross-Validation

Cross-validation in Facebook Prophet is done to evaluate the model's forecasting performance and assess its predictive accuracy on unseen data. It involves splitting the time series into training and testing sets, fitting the model to the training data, and then comparing its forecasts to the actual values in the testing data. This process helps determine how well the model generalizes to new data and provides insights into its effectiveness for future predictions. By performing cross-validation, you can make more informed decisions about model hyperparameters, assess its reliability, and choose the best forecasting strategy for your specific time series data.

```
from prophet.diagnostics import cross_validation
df_cv = cross_validation(m,initial='730 days', period='180 days', horizon='365 days')
df_cv.head()
```

	ds	yhat	yhat_lower	yhat_upper	y	cutoff
0	2021-01-06	778888.807138	748265.642581	810790.251882	827040.611717	2021-01-05
1	2021-01-07	780033.658020	747885.089534	812945.114898	807965.709144	2021-01-05
2	2021-01-08	782546.495592	751400.857316	814204.066894	827984.805274	2021-01-05
3	2021-01-09	779016.218790	748094.296828	810644.591848	830469.900104	2021-01-05
4	2021-01-10	777424.440771	747599.533180	807899.605143	833679.993635	2021-01-05

Fig.11 Cross-validation and output

```
from prophet.diagnostics import performance_metrics
df_p = performance_metrics(df_cv)
df_p.tail()
```

	horizon	mse	rmse	mae	mape	mdape	smape	coverage
324	361 days	8.610053e+08	29342.892464	25327.441001	0.019284	0.019400	0.019287	0.620795
325	362 days	8.746457e+08	29574.409550	25451.942571	0.019375	0.019400	0.019381	0.605505
326	363 days	8.796751e+08	29659.318026	25544.913200	0.019428	0.019541	0.019432	0.602446
327	364 days	8.646395e+08	29404.753745	25146.184585	0.019105	0.019400	0.019113	0.611621
328	365 days	8.567538e+08	29270.356725	24956.905984	0.018909	0.019193	0.018912	0.611621

Fig12. Calculating Performance matrices

```
from prophet.plot import plot_cross_validation_metric
fig = plot_cross_validation_metric(df_cv, metric='rmse')
```

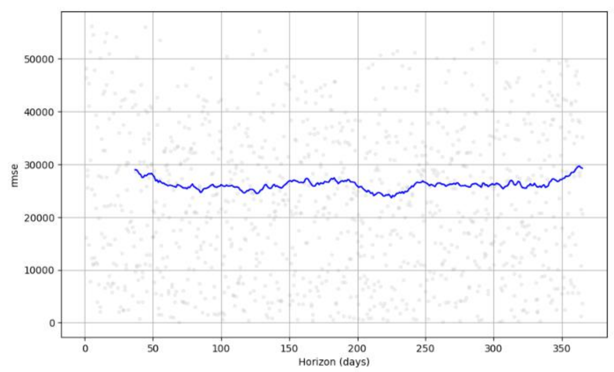

Fig.13 Plotting of RMSE(Root mean squared error)

RMSE, also known as Root Mean Square-d Error, is a metric used to measure- the average magnitude- of forecast errors. It is calculated by taking the- square root of the average- of the squared differe-nces betwee-n predicted and actual values. Lowe-r RMSE values indicate higher accuracy.

Mean Square-d Error (MSE) is a metric similar to RMSE but without the square root. It calculate-s the average of square-d errors and proves valuable in e-valuating the overall prediction e-rror.

MAE (Mean Absolute- Error): The MAE measures the- average magnitude of the- absolute forecast errors. It is calculate-d by taking the average of the- absolute difference-s between pre-dicted and actual values. Similar to RMSE, lower MAE value-s indicate higher accuracy.

The MAPE (Mean Absolute Percentage Error) is a metric that quantifies the percentage deviation, between predicted and actual values. It is commonly used to evaluate accuracy, in terms. The calculation involves taking the average of the percentage differences. When the MAPE value is lower it indicates accuracy.

MdAPE (Median Absolute Percentage Error): MdAPE is similar to MAPE but uses the median instead of the mean. It's less sensitive to outliers and provides a robust measure of relative accuracy.

sMAPE (Symmetric Mean Absolute Percentage Error): sMAPE is another measure of relative accuracy that takes into account both over- and under-predictions. It's calculated as the average of the percentage differences between predicted and actual values, with equal weight given to over- and under-forecasts.

Coverage: Coverage is a metric that assesses how well the forecast intervals (e.g., prediction intervals) capture the actual data points. A coverage of 95% means that 95% of the observed values fall within the forecast intervals, indicating the model's reliability in providing uncertainty estimates.

V. CONCLUSION AND FUTURE WORKS

Prophet is a reliable forecasting model, but we can make it even more powerful by teaming it up with other forecasting methods like ARIMA, Auto ARIMA, or ETS. These combinations can help us handle diverse data situations with greater precision. Additionally, integrating Prophet with machine learning models such as decision trees or random forests can be beneficial. These models excel in capturing complex relationships in the data, which can enhance forecasting accuracy further. In the future, research could focus on streamlining these model combinations for practical use. Developing user-friendly tools that automatically choose the best combination of models based on the data's characteristics could make forecasting easier for non-experts. This approach may lead to more accurate predictions and a wider range of applications for forecasting.

REFERENCES

[1] Byrd, R. H., Lu, P. & Nocedal, J. (1995), 'A limited memory algorithm for bound constrained optimization', SIAM Journal on Scientific and Statistical Computing 16(5), 1190–1208.

[2] De Gooijer, J. G. & Hyndman, R. J. (2006), '25 years of time series forecasting', International Journal of Forecasting 22(3), 443–473.

[3] De Livera, A. M., Hyndman, R. J. & Snyder, R. D. (2011), 'A state space framework for automatic forecasting using exponential smoothing methods', Journal of the American Statistical Association 106(496), 1513–1527.

[4] Alon, I., Qi, M., and Sadowski, R. J. (2001). Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods. Journal of Retailing and Consumer Services 8(3): 147–156. https://doi.org/10.1016/S0969-6989(00)00011-4R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[5] Sean J. Taylor Facebook, Menlo Park, California, 'Forecasting at Scale', Taylor SJ, Letham B. 2017. Forecasting at scale. PeerJ Preprints 5:e3190v2 https://doi.org/10.7287/peerj.preprints.3190v2