ISSN: 2755-0214

Journal of Economics & Management Research

Review Article

Open Access

Forecasting Payables in Financial Industry using Prophet Algorithm

Karthika Gopalakrishnan

Data Scientist, USA

ABSTRACT

In the realm of finance, accurately predicting payables is crucial for effective financial management and planning. Traditional forecasting methods may fall short in capturing the complex patterns inherent in financial data. This paper proposes the use of the FbProphet model, a robust time-series forecasting tool developed by Facebook, for predicting payables in the finance industry. The paper provides an introduction of FbProphet, review relevant literature, detail our methodology for implementing the model, present empirical results, and offer conclusions on the effectiveness of FbProphet for payable prediction.

***Corresponding author**

Karthika Gopalakrishnan, Data Scientist, USA.

Received: July 06, 2022; **Accepted:** July 11, 2022; **Published:** July 17, 2022

Keywords: Payables, FbProphet, Finance, Forecasting, Time-series Analysis

Introduction

Predicting payables accurately is essential for businesses to manage their cash flow effectively and make informed decisions about financial obligations. In the finance industry, traditional forecasting methods often struggle to capture the inherent complexities of financial data, such as seasonality, trends, and irregularities. To address these challenges, advanced time-series forecasting models have been developed, among which FbProphet stands out for its simplicity and effectiveness.

Literature Review

FbProphet is a forecasting tool developed by Facebook, designed to handle time-series data with multiple seasonality and outliers. Unlike traditional statistical models, FbProphet incorporates domain knowledge and heuristics to produce more accurate forecasts. The model is built on the foundation of Generalized Additive Models (GAMs) and utilizes a decomposable time-series model consisting of trend, seasonality, and holiday effects.

Research on FbProphet has demonstrated its effectiveness in various domains, including retail sales forecasting, demand forecasting, and financial time-series analysis. Studies have highlighted its ability to handle noisy data, automatic handling of missing data, and scalability to large datasets. Moreover, FbProphet's user-friendly interface and ease of implementation make it accessible to both novice and experienced analysts. Some of the key features of the Prophet model are

• **Additive Decomposition Model:** Prophet decomposes time-series data into three main components: trend, seasonality, and holidays. This decomposition is additive, meaning that the time-series data is modeled as the sum of these components. The trend captures the overall direction of the data, while seasonality accounts for periodic fluctuations, and holidays represent one-off events that impact the time series.

• **Flexibility:** Prophet is highly flexible and customizable, allowing users to specify various parameters such as seasonality, holidays, and trend changepoints. Users can include custom seasonalities, such as weekly, monthly, or yearly patterns, and define holidays that affect the time series.

• **Automatic Changepoint Detection:** Prophet automatically detects changepoints in the data, which are points where the time-series trajectory changes significantly. Changepoints allow the model to capture abrupt changes in the trend, such as shifts in growth rates or sudden fluctuations.

• **Handling Missing Data and Outliers:** Prophet handles missing data and outliers seamlessly, utilizing an imputation strategy to fill in missing values and robust statistical methods to identify and mitigate the impact of outliers on the forecast.

• **Scalability:** Prophet is designed to be scalable, capable of handling large datasets efficiently. It implements a parallelized algorithm that can train models quickly, making it suitable for real-time or batch forecasting applications.

• **User-Friendly Interface:** Prophet provides a user-friendly interface that requires minimal configuration and parameter tuning. It is implemented in Python and includes comprehensive documentation and examples to help users get started with forecasting tasks quickly.

• **Diagnostic Tools:** Prophet includes diagnostic tools for evaluating forecast performance, such as cross-validation and Performance metrics like Mean Absolute Error (MAE) and Mean Squared Error (MSE). These tools help users assess the accuracy and reliability of their forecasts.

Methodology

To predict payables using the Prophet model, we follow a structured methodology

- **Data Preparation:** Collect historical payable data, preprocess, and format it for analysis.
- **Model Configuration:** Configure the Prophet model, specifying parameters such as seasonality, holidays, and trend flexibility.
- **Model Training:** Fit the model to the historical data, allowing Prophet to learn patterns and relationships.
- **Forecasting:** Generate future payable forecasts based on the learned patterns.
- **Evaluation:** Evaluate the model's performance using appropriate metrics such as Mean Absolute Error (MAE) and Mean Squared Error (MSE).

Dataset

The present study leverages real-time Account Payable transactions sourced from the publicly available City of Oxnard Finance Department covering the fiscal year 2021 along with the preceding four fiscal years. The dataset encompasses daily transactions of payables, including details such as check number, vendor name, description, and invoice number. A comprehensive overview of the dataset is illustrated in Table 1. Utilizing Python, the time series representation of the account payables is depicted in Figure 1.

Table 1: Dataset

index	Date	CheckNum	VendorName	Amount	Description	AccountNumber	InvoiceNumber	AcctPerYr	AcctPerMo
0	08/25/2014	9999998	MISCELLANEOUS VENDOR CX	1373.43	LID-560 POCKET READER 4 A	101-2106-802.81-36	544	2015	2
1	07/09/2020	373810	THE WHARF	134.8	SAFETY BOOTS	101-5701-805.81-05	86037	2019	7
2	07/22/2020	374297	E RECYCLING OF CALIFORNIA	-529.54	E WASTE	631-6301-842.82-09	87656	2020	8
3	06/25/2020	373524	MOTOR VEHICLE NETWORK	600.0	BEVERAGE CONT RECYCLING	631-6828-823.82-09	20-36202	2020	9
4	06/18/2020	373208	HARRIS WATER CONDITIONING	40.0	ENGINEERING WATER	101-3201-803.81-02	47638	2020	9

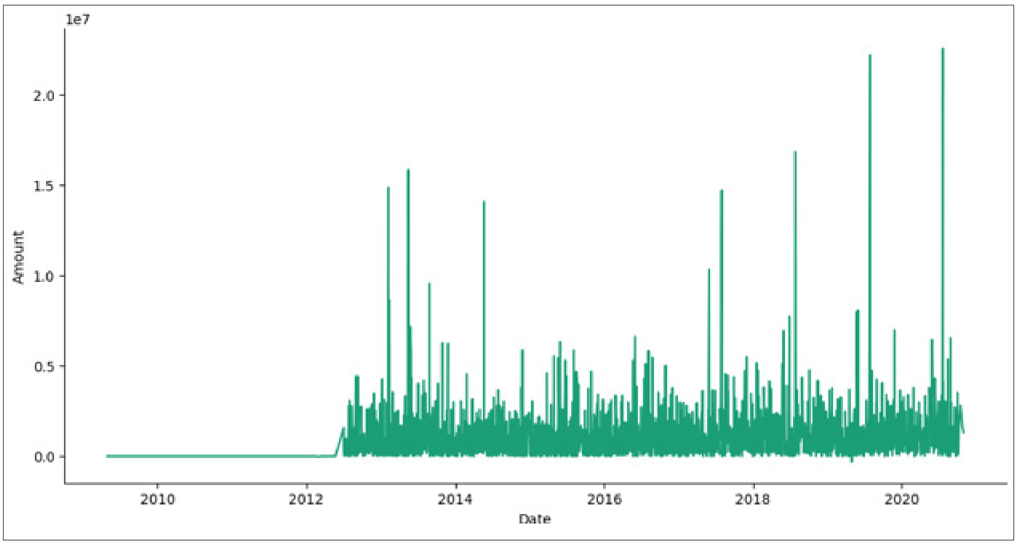

Figure 1: Time Series Representation of Data

The payables are aggregated by date, disregarding vendor names and invoice numbers, to facilitate time series modeling. Table 2 presents an overview of the cumulative transactions per day.

	Date	Amount
0	2009-04-28	-175.00
1	2009-06-02	-39.93
2	2010-01-19	-10.00
3	2010-03-16	-44.58
4	2010-05-18	-10.00

Table 2: Aggregated Data Overview

The time series signal underwent stationarity testing using the Augmented Dickey-Fuller Test (ADF) via the "adfuller" command in the Python kernel. Given that the p-value was less than 0.05, the signal was deemed stationary. Consequently, no further adjustments were necessary to convert the signal into a stationary form for ARIMA modeling. Additionally, the time series signal was regarded as additive, and its decomposition was visualized in Figure 2.

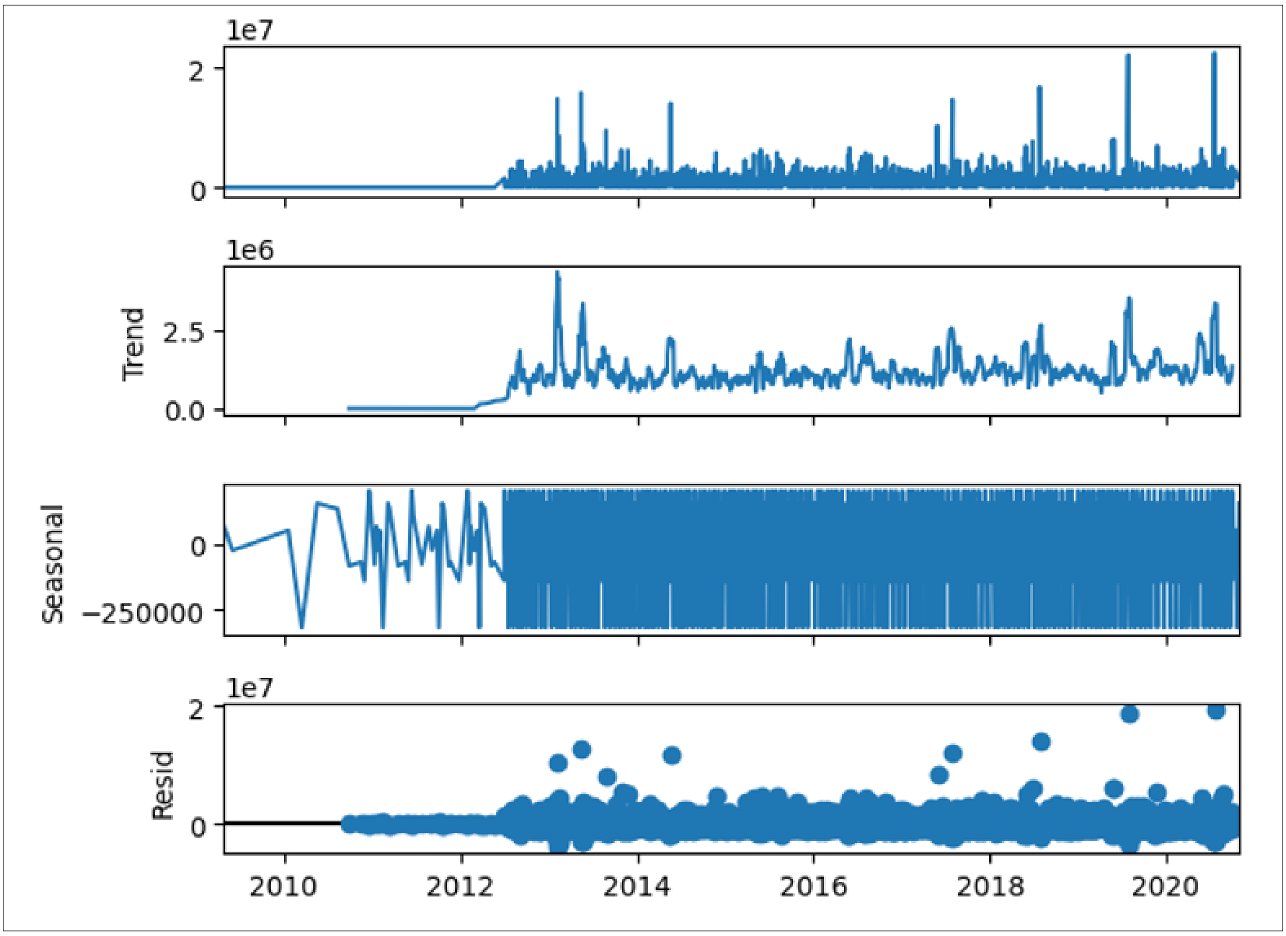

Figure 2: Time Series Signal Decomposition

Model Configuration and Evaluation

In this study, the basic Prophet model was employed without any alterations to the trend and seasonality components. The dataset underwent division into training and test sets. Subsequently, the model underwent training on the training dataset, followed by validation against the test set using the trained model. The forecasted outcomes were depicted in Figure 3, while Figure 4 illustrates the plot contrasting actual values with forecasted values for the test set.

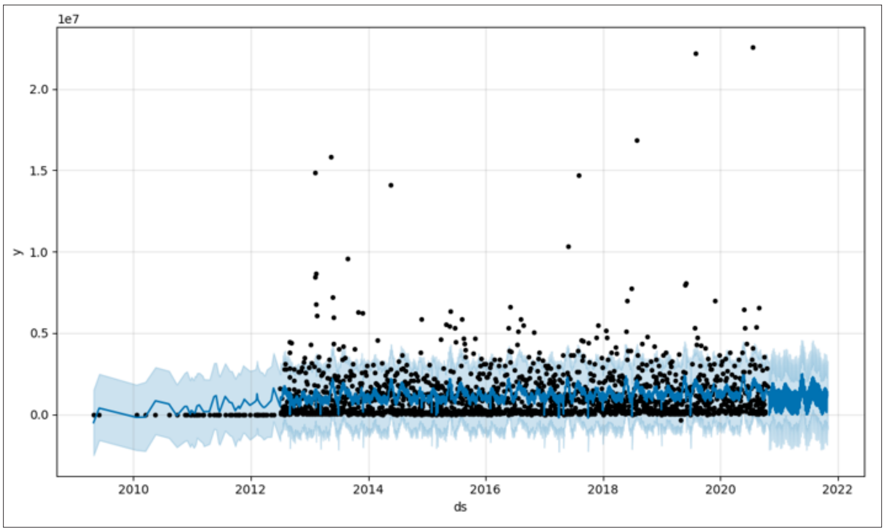

Figure 3: Forecasted values using Prophet

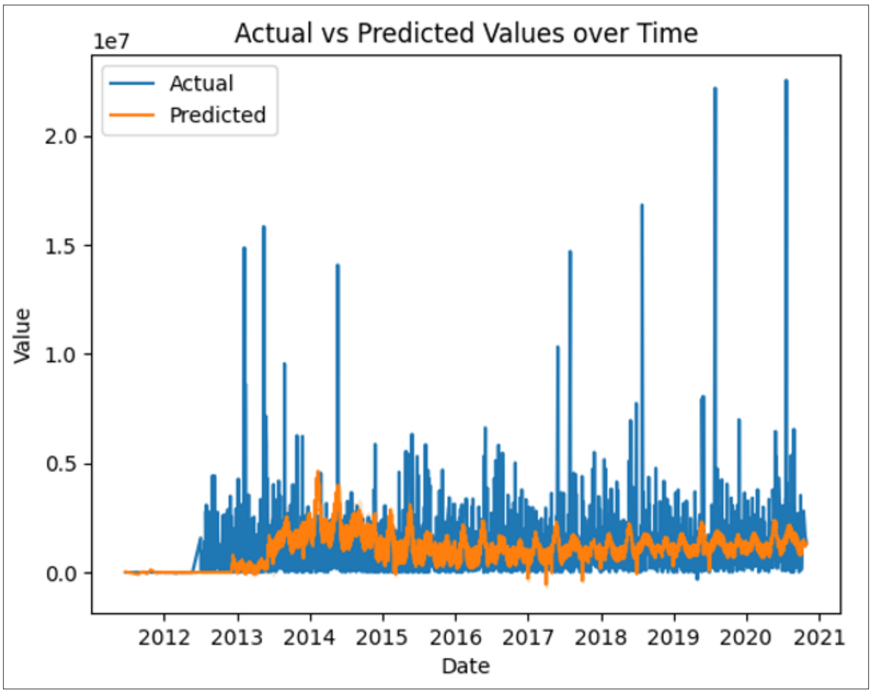

Figure 4: Actual vs Forecasted Values

Prophet incorporates time series cross-validation functionality to evaluate forecast error using historical data. This process involves selecting cutoff points within the historical data and fitting the model using information up to each respective cutoff point. Subsequently, we can assess the forecasted values against the actual values. Cross-validation was done to assess prediction performance on a horizon of 365 days, starting with 730 days of training data in the first cutoff and then making predictions every 180 days and the results were shown in figure 5.

index	ds	yhat	yhat_lower	yhat_upper	y	cutoff
0	2011-06-21 00:00:00	18613.811930851774	18557.741231327425	18672.396492720258	-600.0	2011-06-14 00:00:00
1	2011-07-26 00:00:00	-30165.838116077626	-30225.914257487726	-30110.1285755818	-600.0	2011-06-14 00:00:00
2	2011-08-23 00:00:00	-99438.1295467815	-99496.97743776509	-99381.5365281349	-677.9	2011-06-14 00:00:00
3	2011-09-06 00:00:00	14273.631548597765	14215.057515043987	14336.424085574783	-294.22	2011-06-14 00:00:00
4	2011-09-27 00:00:00	16056.352482844595	15998.991958946208	16116.437753116697	-600.0	2011-06-14 00:00:00

Figure 5: Cross Validation Results

The performance_metrics utility was used to compute some useful statistics of the prediction performance (yhat, yhat_lower, and yhat_upper compared to y), as a function of the distance from the cutoff (how far into the future the prediction was). The statistics computed were mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean absolute percent error (MAPE), median absolute percent error (MDAPE) and coverage of the yhat_lower and yhat_upper estimates. These were computed on a rolling window of the predictions in dataset after sorting by horizon (ds minus cutoff). By default, 10% of the predictions were included in each window and the metrics were represented in figure 6 and comparison of RMSE and MAE values were shown in figure 7.

index	horizon	mse	rmse	mae	mdape	smape	coverage
0	40 days 00:00:00	2066247485951.5068	1437444.7766615269	1026726.4340515832	0.901091751402455	1.0081584691457095	0.8600583090379009
1	41 days 00:00:00	2060010436265.0981	1435273.6450813476	1019854.2843350164	0.901091751402455	1.0131342232785598	0.8537414965986394
2	42 days 00:00:00	2050008355585.698	1431785.0242217572	1013610.4694810866	0.9330958057519506	1.0275439644470516	0.8513119533527697
3	43 days 00:00:00	2066373720387.1553	1437488.6853075246	1019922.6212386193	0.9568723651477473	1.0372505360373236	0.8503401360544218
4	44 days 00:00:00	2040091084806.6892	1428317.5714128455	1014398.1808466838	0.9605855317985234	1.0394226519363097	0.8496112730806609

Figure 6: Performance Metrics

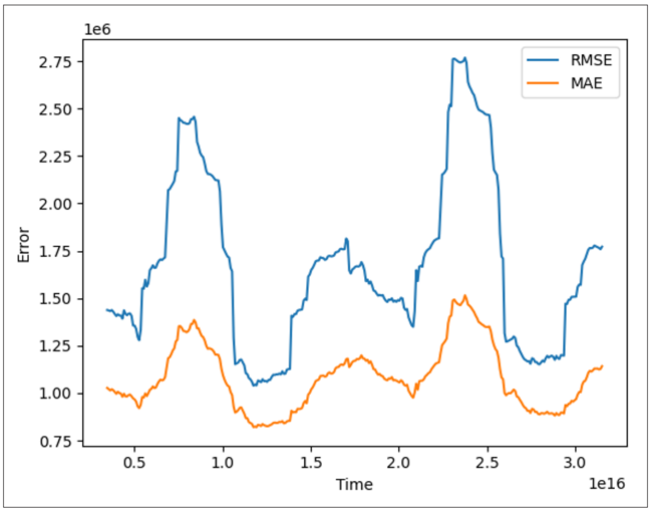

Figure 7: Comparison of RMSE and MAE

Conclusion

In conclusion, this study showcased the application of the Prophet model for predicting payables within the finance industry. The basic iteration of the model was utilized for payable prediction. However, there exists potential to enhance the model by incorporating holidays and conducting deeper analysis on the observed outliers to improve the accuracy of payable forecasts. The author aims to implement more effective techniques for outlier identification and fine-tuning of the model to optimize payable forecasting [1-5].

References

1. Sean Taylor J, Benjamin Letham (2017) Forecasting at Scale, The American Statistician 72: 37-45.
2. Hyndman RJ, Athanasopoulos G (20{}8) Forecasting: principles and practice 2nd ed. https://otexts.com/fpp2/.
3. Li S, Zhao Y, Wang K, Cai R (2020) A hybrid forecasting model for the short-term electricity load based on deep learning and FBProphet. IEEE Access 8: 59802-59811.
4. Herrera G, Rosales-Perez J (2020) Time series forecasting using a hybrid ARIMA-Facebook Prophet model for air quality monitoring in the City of Córdoba, Argentina. In 2020 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC) IEEE 1-6.
5. Mukherjee S, Kar S (2020) Short-term Load Forecasting of Electric Power System using Facebook Prophet. In 2020 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) IEEE 1-6.