|  |  |
| --- | --- |
| **Model Card Version:** 0.0\_2024 Prophet **Model Card Authors:** William Earley, Alec Bothwell, Cory LeRoy | Prophet is a forecasting model developed by Facebook's Core Data Science team, designed specifically for time series forecasting tasks. It's directed for use on time series data, and can easily analyze seasonality and regression trends. Prophet is suitable for a wide range of forecasting applications, including business demand forecasting, financial analysis, and trend prediction. We tried prophet because of its simplicity, flexibility, and ability to provide reliable forecasts with minimal effort, making it accessible to both experts and non-experts in time series analysis. |
|  | |

|  |  |  |
| --- | --- | --- |
| Model Snapshot | | |
| Model Overview | | |
| MODEL ARCHITECTURE | INPUT(S) | OUTPUT(S) |
| As per research paper for prophet model:  “A decomposable time series model with three main model components: trend, seasonality, and holidays”  y(t) = g(t) + s(t) + h(t) + ct  “Here g(t) is the trend function which models non-periodic changes in the value of the time series, s(t) represents periodic changes (e.g., weekly and yearly seasonality), and h(t) represents the effects of holidays which occur on potentially irregular schedules over one or more days. The error term ct represents any idiosyncratic changes which are not accommodated by the model; later we will make the parametric assumption that is normally distributed.” | Dataframe of features. All features must be int or float values. Data column is renamed to “ds” while prediction column of sales is renamed to “y” | Float value of sales prediction (held in dataframe[‘yhat’] column) |
| Usage | | |
| APPLICATION | BENEFITS | KNOWN CAVEATS |
| Where has this model been used, or where is it currently used? Include links for readers to learn more. | Why might users choose to use this model, relative to others? Evidence your response with metrics or performance results | Are there any known and preventable failures about this model? |
| Prophet has been used in a wide array of industries, but includes forecast sales, counts, and numbers for different events and areas.  [Quick Start | Prophet (facebook.github.io)](https://facebook.github.io/prophet/docs/quick_start.html#python-api)  [Time Series Forecasting With Prophet in Python - MachineLearningMastery.com](https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/) | Prophet is known for it’s transparency in results and ease of use. It’s as simple as instantiating the model, feeding the dataset, adding holidays and regressors, and then predicting on the test set. Prediction includes y\_hat, y\_hat\_lower, y\_hat\_upper, influencing factors, trend number, etc. | Limitations in performance, only works for univariate time series, and is hyper tuned to holidays but not necessarily other correlated features. |
| Model Creators | | |
| CITATION Taylor, Sean, & Letham, Benjamin (September 27, 2017). Forecasting at Scale.  [Forecasting at scale [PeerJ Preprints]](https://peerj.com/preprints/3190/) | | |
|
|
| System Type | | |
| SYSTEM DESCRIPTION | UPSTREAM DEPENDENCIES | DOWNSTREAM DEPENDENCIES |
| The predictions of Prophet are to be used as a standalone model. | The data fed into Prophet must be numerical. Categorical fields are to be label-encoded. Prophet also must include the field name of “ds” for timestamp and “y” for predicted value.  Model imported using: import prophet  Model can be created and fit using the below code:  Model = Prophet().fit(model\_df) | NA |

|  |  |  |  |
| --- | --- | --- | --- |
| Data Overview | | | |
| TRAINING DATASET SNAPSHOT | | DATASET MAINTENANCE & VERSIONS | EVALUATION DATA |
| Statistics of df\_train.pkl | | Data is preprocessing in the Feature Engineering notebook. Files are df\_train.pkl, df\_val.pkl, and df\_test.pkl | Train 1,935,987  Validation 276,569  Test 553,141  **Above:** Train, Validation, Test split. 70%-10%-20%. |
| **Dataset Size** | 100,180 KB |
| **Number of Instances** | 1,935,987 |
| **Number of Fields** | 29 |
|  | |
|  | | | |

###### 