Project Report- Time Series Forecasting Using Prophet Algorithm

April 29th, 2025

1. Introduction

This project utilizes the Prophet Algorithm for time series forecasting.

- **Features Engineering**: by adding additional columns to the dataset for deeper understanding
- **Renaming:** the columns according to the Prophet Model Algorithm
- **Yhat:** indicates the forecasting values

2. List of examples of the use of the method in science or Real-World Applications (with references)

2.1 List of Examples

1. Used for Temperature Forecasting- https://www.researchgate.net/publication/348064035_Time_Series_Prediction_Based_on_Facebook_Prophet_A_Case_Study_Temperature_Forecasting_in_Myintkyina
2. Used for Sales Forecasting- https://www.researchgate.net/publication/351378322_Time_Series_Forecasting_Model_for_Supermarket_Sales_using_FB-Prophet
3. Used for Energy Consumption Forecasting- https://www.mdpi.com/2071-1050/15/22/15860
4. Used for Stock Price Forecasting- https://ieeexplore.ieee.org/document/9776830
5. Used for House Price Forecasting- https://www.semanticscholar.org/paper/Time-Series-Forecasting-of-U.S.-Housing-Price-Index-Nunna-Zhou/5feda1b8a6b6f61a53c959195c057d9f42d7e49a
6. Used for Website Traffic Forecasting- https://www.researchgate.net/publication/349447175_Forecasting_Website_Traffic_Using_Prophet_Time_Series_Model

2.2 References:

1. K. C. Nunna, Z. Zhou and S. R. Shakya, "Time Series Forecasting of U.S. Housing Price Index Using Machine and Deep Learning Techniques," 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Nadi, Fiji, 2023, pp. 1-6, doi: 10.1109/CSDE59766.2023.10487733.

2. Oo, Zar & Phyu, Sabai. (2020). Time Series Prediction Based on Facebook Prophet: A Case Study, Temperature Forecasting in Myintkyina. International Journal of Applied Mathematics Electronics and Computers. 8. 263-267. 10.18100/ijamec.816894.

3. K. Prakhar, S. S, S. E, K. M and S. K. B, "Effective Stock Price Prediction using Time Series Forecasting," *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2022, pp. 1636-1640, doi: 10.1109/ICOEI53556.2022.9776830.

4. Son, N., & Shin, Y. (2023). Short- and Medium-Term Electricity Consumption Forecasting Using Prophet and GRU. *Sustainability*, *15*(22), 15860. https://doi.org/10.3390/su152215860

5. Yildiz, Umut & Korkut Uysal, Sila. (2024). Electricity Consumption Forecasting Using the Prophet Model in Industry: A Case Study. 102-111. 10.1007/978-3-031-53717-2_10.

6. Author links open overlay panelHossein Abbasimehr a *et al.* (2023) *A novel hybrid model to forecast seasonal and Chaotic Time Series*, *Expert Systems with Applications*. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0957417423029639?via%3Dihub (Accessed: 28 March 2025).

3. Libraries and Functions

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from prophet import Prophet
```

```
from sklearn.metrics import mean_squared_error, mean_absolute_error
```

- Fbprophet- **Prophet** is an open-source forecasting tool developed by **Facebook** for **time series prediction**. It simplifies forecasting by automatically handling trends, seasonality, and holidays with just a few lines of code. Ideal for **business metrics, sales forecasts, and seasonal data** without requiring advanced statistical knowledge.

- **Numpy** (Numerical Python)- is a fundamental Python library for numerical computing. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on them efficiently. It is widely used in data science, machine learning, and scientific computing due to its speed and versatility.

- **Pandas**- is a powerful Python library for **data manipulation and analysis**. It provides **DataFrames** (table-like structures) to efficiently load, clean, filter, and transform structured data (e.g., CSV, Excel). Key features include handling missing values, merging datasets, and performing aggregations.

- **Seaborn**- is a **Python data visualization library** built on Matplotlib, designed for creating **statistical graphics** with minimal code. It simplifies complex plots (like heatmaps, regression lines, and distribution charts) and integrates well with Pandas DataFrames. Ideal for **exploratory data analysis (EDA)** and making visually appealing plots.

- **Matplotlib**.pyplot- is a core visualization library used to create line plots, bar charts, pie charts, and other static visualizations.In this project, it's used to show sentiment distribution and the word cloud.

- **sklearn**.metrics- sklearn.metrics is a model evaluation module in Scikit-Learn that is used to calculate performance metrics for classification, regression, clustering, and ranking tasks like functions accuracy_score(), precision_score(), recall_score(), f1_score() for Classification Metrics, and like functions mean_squared_error() MSE), mean_absolute_error() MAE, r2_score() R^2 for Regression Metrics.

4. Dataset Description

4.1 General Information

- Source: Pennsylvania-New Jersey-Maryland Interconnection East (PJME) Energy Consumption in Mega Watt (MW) Dataset
- Description: Hourly power consumption data from 2002-2018 for the entire east region.
- Fields:
    - Data Type: Structured data
    - Domain: Energy/Electricity Consumption (the data represents hourly power consumption in megawatts)
    - Associated Task: Time Series Forecasting (predicting future power consumption based on historical hourly data)

- Feature Type: **Numerical** (the PJME_MW column contains continuous numerical values representing power consumption in megawatts). **Datetime** (the Datetime column represents timestamps)
- Number of Instances: 145366
- Number of Features: **2 features**:
  - Datetime (timestamp)
  - PJME_MW (power consumption in MW)
- Missing Values: None

4.2 Feature Description

1. **Dataset Overview**: The dataset consists of hourly power consumption data from PJM East, spanning from 2002 to 2018. The target variable is PJME_MW, which represents the hourly power consumption in megawatts (MW).

2. **Time-Related Features**:
   - **Hour (hour)**: The hour of the day in 24-hour format, ranging from 0 (midnight) to 23 (11 PM).
   - **Day of Week (dayofweek)**: Represents the day of the week, where 0 = Monday, 1 = Tuesday, ..., 6 = Sunday.
   - **Weekday (weekday)**: A categorical variable with ordered categories: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.
   - **Month (month)**: The month of the year, ranging from 1 (January) to 12 (December).
   - **Quarter (quarter)**: The quarter of the year, ranging from 1 to 4.
   - **Year (year)**: The year of data collection, ranging from 2002 to 2018.
   - **Day of Year (dayofyear)**: The day of the year, ranging from 1 to 366.
   - **Day of Month (dayofmonth)**: The day of the month, ranging from 1 to 31.
   - **Week of Year (weekofyear)**: The week of the year, calculated using isocalendar().week.

3. **Seasonal Feature**:
   - **Season (season)**: A categorical variable representing the four seasons: Spring, Summer, Fall, and Winter. This is derived from a date offset calculation.

4. **Holiday Feature**:
   - **Holidays (holiday)**: A binary indicator showing whether the day is a holiday (1 = Yes, 0 = No). The dataset includes US federal holidays such as Independence Day, Labor Day, and Christmas Day.

5. **Target Variable**:
   - **Power Consumption (PJME_MW)**: The target variable, representing the hourly power consumption in megawatts (MW). The values range from approximately 20,000 to 60,000 MW.

6. **Model Performance Metrics**:
   - **Root Mean Squared Error (RMSE)**: 6616.97 (without holidays), 6639.59 (with holidays).
   - **Mean Absolute Error (MAE)**: 5181.91 (without holidays), 5201.46 (with holidays).
   - **Mean Absolute Percentage Error (MAPE)**: 16.51% (without holidays), 16.56% (with holidays).

7. **Forecasting**:
   The Prophet algorithm was used for time series forecasting, incorporating trend, seasonality, and holiday effects. The model was trained on data up to January 1, 2015, and tested on data from January 1, 2015, onwards. Future predictions were also made for the next 365 days at an hourly frequency.

8. **Visualizations**:
    - Time series plots showing actual vs. predicted power consumption.
    - Component plots displaying trend, weekly seasonality, yearly seasonality, and holiday effects.
    - Box plots illustrating power consumption by day of the week and season.

5. <u>Empirical Analysis</u>

5.1 Goal

Objective:
- Analyze the impact of temporal and seasonal factors on hourly power consumption (PJME_MW).
- Build a forecasting model to predict power demand using historical time series data.

Key Questions:
- Which time-based features (hour, day of week, season, holidays) most influence power consumption?
- How accurately can Prophet model future power demand?

5.2 Assumptions

- Temporal patterns (hourly, daily, weekly, yearly) significantly affect power consumption.
- Holidays reduce industrial/commercial power usage, impacting overall demand.
- Seasonality (e.g., higher demand in summer/winter due to HVAC usage) is captured by Prophet's additive model.
- Data is complete (no major missing values) and representative of PJM East's grid behavior.

5.3 Results

Prophet decomposes time series into:
1. **Trend**: Long-term growth/decline in demand.
2. **Seasonality**:
    - **Daily**: Peak demand during daytime (e.g., 12 PM–6 PM).
    - **Weekly**: Lower consumption on weekends.
    - **Yearly**: Higher demand in summer (cooling) and winter (heating).
3. **Holidays**: Reduced demand on federal holidays (e.g., July 4, Christmas).

Model Performance
- **Root Mean Squared Error (RMSE)**:
    - **Without holidays**: **6,616.97 MW**
    - **With holidays**: **6,639.59 MW** (minor improvement).
- **Mean Absolute Percentage Error (MAPE)**: **~16.5%** for both models.
- **Visual Diagnostics**:
    - Prophet's forecast closely tracks actuals but underestimates extreme peaks (e.g., heatwaves).
    - Holiday effects are visible (e.g., July 4 shows a dip).

5.4 Interpretation & Recommendations

1. **Hourly Patterns Dominate**:
    - Power demand peaks at **3 PM** (daily high) and drops overnight.
    - **Action**: Grid operators should allocate reserves for peak hours.
2. **Seasonal Trends Matter**:
    - **Summer/Winter** show higher demand than spring/fall.
    - **Action**: Plan maintenance in low-demand seasons (spring/fall).
3. **Holidays Have Minor Impact**:
    - Demand drops ~5% on holidays (e.g., Independence Day).
    - **Action**: Adjust generation schedules for holiday periods.
4. **Prophet's Limitations**:
    - Struggles with **unusual events** (e.g., storms) not in training data.
    - **Recommendation**: Combine with anomaly detection for robustness.

5.5 Prophet Method Strengths and Weaknesses

S.No.	Strengths	Weaknesses

1	**Designed for Time Series**
	• Handles missing data & outliers gracefully (unlike ARIMA, which requires strict stationarity).
	• Automatic trend detection: Captures long-term growth, cycles, and seasonality without manual tuning.
		Limited to Additive Patterns
		• Assumes linear trends + additive seasonality, which may fail for:
		  ○ Multiplicative trends (e.g., exponential demand growth).
		  ○ Complex interactions (e.g., temperature + weekday effects).

2	**Interpretability**
	• Decomposes forecasts into components:
	  ○ Trend (e.g., increasing demand over years).
	  ○ Seasonality (daily, weekly, yearly patterns).
	  ○ Holiday effects (e.g., lower usage on July 4).
	• Visual diagnostics (e.g., plot_components()) make it easy to explain to stakeholders.
		Struggles with Short-Term Volatility
		• Poor at capturing sudden spikes/drops (e.g., power outages, extreme weather).
		• No built-in anomaly detection (unlike LSTM autoencoders).

3	**Flexibility** • Supports custom seasonality (e.g., adding monthly cycles). • Works with irregular time intervals (unlike ARIMA). • No need for feature engineering (unlike machine learning models like XGBoost).	**Not Ideal for High-Frequency Data** • Hourly data works, but sub-hourly (e.g., minute-level) may require downsampling. • LSTMs/Transformers may outperform for ultra-high-frequency forecasting.
4	**Fast Training** • Optimized for large datasets (faster than LSTM/Deep Learning for most use cases). • Parallelizable for scalability.	**Static Model Assumptions** • Assumes fixed seasonality (e.g., same yearly pattern every year). • Cannot adapt to regime changes (e.g., post-pandemic energy usage shifts).