

Stock Price Forecasting with Prophet –Venu Yerramsetti

This project forecasts Apple (AAPL) stock prices using **Prophet**, a powerful open-source time series forecasting library. Historical data is fetched from Yahoo Finance via `yfinance`, and forecasts are visualized using `matplotlib`.

Technologies Used

- Python
- `yfinance`
- Prophet
- Matplotlib
- Pandas

Project Structure

File Name	Description
<code>stock_forecast.py</code>	Python script for downloading data, training model, forecasting
<code>forecast_plot.png</code>	Image file showing the forecasted stock price
<code>requirements.txt</code>	List of Python dependencies
<code>README.md</code>	Project overview and instructions

Output

The forecast plot will be saved as `forecast_plot.png`.
It will also be displayed interactively using `matplotlib`.

About Prophet

Prophet is a forecasting tool built by Meta (Facebook) for time series data. It supports modeling:

- Seasonality (yearly, weekly, daily)
- Holiday effects
- Non-linear trends
- Missing data and outliers

🌐 How GitHub Fits In

This GitHub repository serves as a central hub for this project:

- **Hosts the source code:** All the project files are stored here.
- **Tracks project history:** You can see all the changes made to the code over time.
- **Shares the project with others:** Makes it easy for collaborators and others to access and contribute.

Important Note: GitHub does not execute your code directly. To run this project, you'll need to **clone the repository** and execute the code locally on your machine or within a cloud notebook environment.

🚀 Optional: Run in Google Colab or Binder

For a quick and easy way to run this notebook online without any local setup, you can use one of these free platforms:



Alternatively, you can upload the notebook directly to [Google Colab](https://colab.research.google.com/) and run all cells there.

▶ How to Run the Project Locally

```
# 1. Clone the Repository
git clone https://github.com/VenuYerramsetti/stock-price-forecasting-prophet
cd stock-price-forecasting-prophet

# 2. Set Up the Environment
pip install -r requirements.txt
```

```
# If needed, install manually:
pip install yfinance prophet matplotlib pandas

# If Prophet fails to install:
pip install pystan==2.19.1.1
pip install prophet

# 3. Run the Code
# Run the script:
python stock_forecast.py

# Or run the Jupyter Notebook:
jupyter notebook
# Then open `stock_forecast.ipynb` and run all cells
```

Requirements

- yfinance==0.2.61
- prophet==1.1.6
- matplotlib==3.4.3
- pandas==1.3.3

Code blocks Explanation

```
import yfinance as yf
from prophet import Prophet
import pandas as pd
import matplotlib.pyplot as plt
```

Imports the necessary libraries:

- yfinance to fetch stock data from Yahoo Finance.
- Prophet for time series forecasting.
- pandas for data manipulation.
- matplotlib.pyplot for plotting.

1. Download stock data

```
data = yf.download('AAPL', start='2018-01-01', end='2024-12-31')
```

- Downloads historical stock data for Apple (ticker symbol AAPL) from January 1, 2018 to December 31, 2024.
- The result, data, is a pandas DataFrame containing Open, High, Low, Close, Volume, and Adj Close prices for each trading day in that period.

2. Prepare data for Prophet

```
df = pd.DataFrame() df['ds'] = data.index df['y'] = data['Close'].values
```

- Creates a new DataFrame df formatted specifically for Prophet.
- ds stand for datestamp — it must be a column of datetime objects representing the dates of observations. Here it is set from the index of the original data DataFrame (which contains the trading dates).
- y is the target variable to forecast — here, the closing price of the stock on each date. Prophet expects the target variable column to be named y.
- This formatting is required because Prophet only accepts DataFrames with these two columns named exactly ds and y.

Check types (optional debug print)

```
print(type(df['y']), df['y'].shape)
```

- Prints the type and shape of the y column. This is a quick sanity check to ensure the target data is in the expected format (a pandas Series with the correct number of elements).

3. Initialize and fit model

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

- Initializes a new Prophet forecasting model with default parameters.
- Fits the model to the historical stock data you prepared in df. Prophet uses this data to learn patterns, trends, and seasonality.

4. Make future dataframe & forecast

```
future = model.make_future_dataframe(periods=180)
```

```
forecast = model.predict(future)
```

- `make_future_dataframe(periods=180)` generates a DataFrame that extends the original dates (`ds`) by 180 future days. This DataFrame will be used to generate predictions.
- `model.predict(future)` computes the forecast for both historical and future dates, producing predicted values (`yhat`) along with confidence intervals and components (like trend and seasonality).

5. Plot forecast

```
fig1 = model.plot(forecast)
```

```
fig1.savefig("forecast_plot.png")
```

```
plt.title("AAPL Stock Price Forecast")
```

```
plt.show()
```

- Creates a plot of the forecasted data including historical and predicted values using Prophet's built-in plotting method.
- Saves the plot as `forecast_plot.png` in your working directory.
- Adds a title to the plot with matplotlib.
- Displays the plot interactively on your screen.
- **Summary of `ds` and `y` in Prophet:** `ds` (datestamp): A column of dates (datetime objects) representing the time component of your data. Prophet requires this exact column name and type.
- `y`: The numeric value you want to forecast (in your case, the daily closing price of AAPL). Prophet expects this to be a numeric column named