Capstone Project

Data Scientist Nanodegree

Paulo Mota

May 2nd, 2023

Definition

Project Overview

The stock market is a dynamic and volatile environment, with prices constantly fluctuating based on a variety of factors such as economic conditions, company performance, and investor sentiment. Accurately predicting stock prices is a challenging task, but it can provide significant benefits to investors and traders. This project aims to develop a stock price predictor forecasting model that utilizes machine learning algorithms to forecast future prices based on historical data.

Problem Statement

The primary objective of this project is to develop an accurate and reliable stock price predictor forecasting model. The model will utilize machine learning algorithms such as regression, time-series analysis, and deep learning to analyze historical data and predict future stock prices. The specific objectives of the project are as follows:

- Gather and preprocess historical stock data from yahoo API
- Develop and train machine learning models for stock price prediction using Prophet
- Evaluate model performance and select the most accurate model
- Deploy a web application to user predict a close price from a stock symbol
- Extra: Create a stock comparator with historical data.

Metrics

The performance of each model will be evaluated using appropriate metrics such as mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE). The model with the best performance will be selected for the deployment model. It measures the average absolute difference between the actual values (y) and the predicted values (yhat).

The formula for calculating MAE is:

$MAE = (1/n) * \Sigma|y - yhat|$

Where:

- n is the number of data points
- Σ represents the sum over all data points
- | | denotes the absolute value

To calculate MAE in Python, we can use the scikit-learn library's mean_absolute_error() function. The function takes two arguments: the actual values (test_data['y']) and the predicted values (predictions['yhat']).

The Root Mean Squared Error (RMSE) is another metric used to evaluate the accuracy of a predictive model. It measures the square root of the average of the squared differences between the actual values (y) and the predicted values (yhat).

The formula for calculating RMSE is:

RMSE = sqrt((1/n) * Σ(y - yhat)^2)

Where:

- n is the number of data points
- Σ represents the sum over all data points
- sqrt denotes the square root

To calculate RMSE in Python, we can use the numpy library's sqrt() function and the scikit-learn library's mean_squared_error() function. The function takes two arguments: the actual values (test_data['y']) and the predicted values (predictions['yhat']).

Analysis

Data Exploration

I used Yahoo Finance API to get the stock data by symbol.

The stock data generally contains information on the performance of a particular stock, and typically includes several columns such as open, high, low, close, volume, and adj.

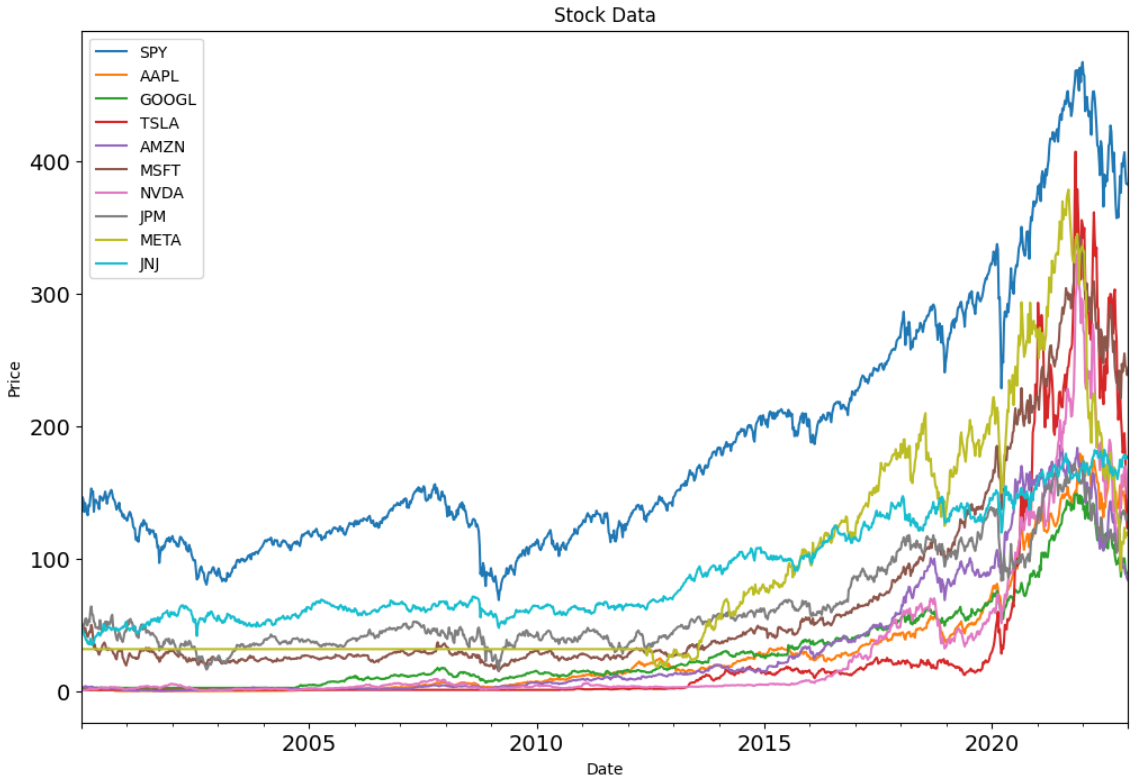
- Open: the price of the stock at the opening of the trading day.
- High: the highest price of the stock during the trading day.
- Low: the lowest price of the stock during the trading day.
- Close: the price of the stock at the end of the trading day.
- Volume: the number of shares of the stock that were traded during the trading day.
- Adj: short for adjusted, this column represents the adjusted stock price that takes into account any changes to the stock such as stock splits or dividends.

These data points are often used by investors, analysts, and traders to analyze the performance of a particular stock, and to make informed decisions about buying or selling that stock. For example, the high and low values can indicate the volatility of the stock, while the volume can provide an indication of how active the market is for that particular stock. The adjusted price is often used to analyze the long-term performance of a stock, as it reflects any changes to the stock that may have occurred over time.

In terms of domain understanding, I realized to make two cleanings:

- backfill and forward fill (because there is no stock data on weekends and which company starts and ends stock activities in different time periods.)

Exploratory Visualization



The following plot displays the closing prices for a selection of stock symbols, after the data processing stage where backfill and forward fill techniques were utilized to fill in any gaps in the stock data that may have occurred within the user's specified date range.

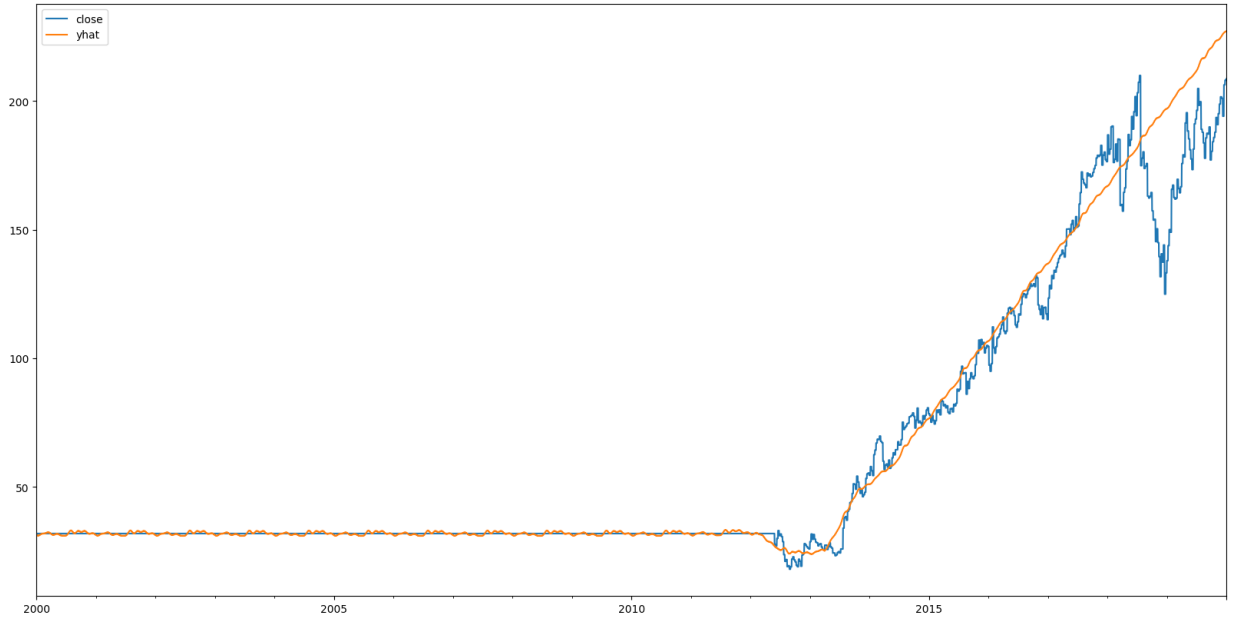
Algorithms and Techniques

The model is a time series forecasting model developed by Facebook called Prophet. It is designed to be easy to use and to provide accurate forecasts for a variety of time series data in our case Stock Price data.

For training the model I needed to prepare my data in a specific format. The data should be in a pandas DataFrame with two columns: 'ds' and 'y', where 'ds' is a datetime column containing the dates and 'y' is the column containing the values you want to forecast.

For testing my predictions I used “Backtesting” , which involves applying a trading strategy to historical stock price data to see how well it would have performed if it had been used in the past.

The figure below shows, the result of predictions about **META**(Facebook stock symbol)



Methodology

Data Preprocessing

The preprocessing done, consists of the following steps:

1. Connection with Yahoo Finance API
2. Get JSON data into Pandas Dataframe
3. Fill stock price data using backfill and forward fill.
4. Create dataframe for train and test

Implementation

The implementation process can be split into two main stages:

1. Prophet training stage
2. The application development stage

During the first stage, the classifier was trained on the preprocessed training data.

1. Implement main functions:
    a. get_stock_data_by_symbol(...): Get pandas dataframe with stock data from a symbol and specific start date and end date.
    b. get_data(...): Get pandas dataframe with stock data from several stock symbols to be plot easier in the line chart.

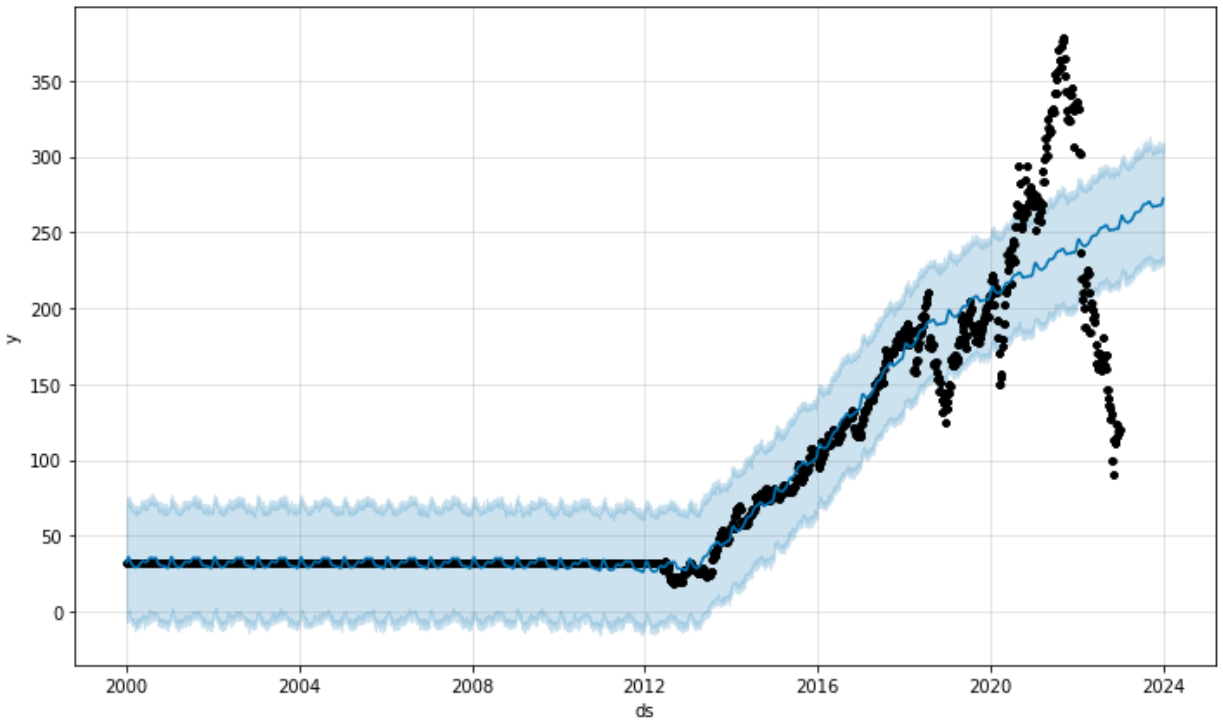
    c. get_data_for_training(...). Get train dataset and test dataset to be processed in the Prophet model.
2. Train the model with stock data
3. Test the predictions with error metrics mentioned before.
4. Use Streamlit library to receive the user input to be processed dynamically.
5. Deploy the application using Streamlit cloud.

Refinement

Refining a Prophet model involves making adjustments to the model's settings, such as its hyperparameters, to improve its accuracy and performance.

If the initial results suggest that the model could benefit from additional information, such as the impact of external factors like holidays or economic indicators, I can add external regressors to the model. This involves including additional data in the form of covariates, which can help the model better capture the relationship between the predictor variables and the target variable.

The plot below displays the real and predicted close values for **META** stock price. Divergence between the two lines indicates periods where the model's predictions were not accurate, such as during the COVID-19 pandemic. To improve the model's accuracy during these specific periods, it is recommended to incorporate this information into the model by providing additional data or adjusting the model's settings. This will help the model better understand the impact of these events on stock prices and improve the accuracy of its predictions.(y = close value and ds = date)



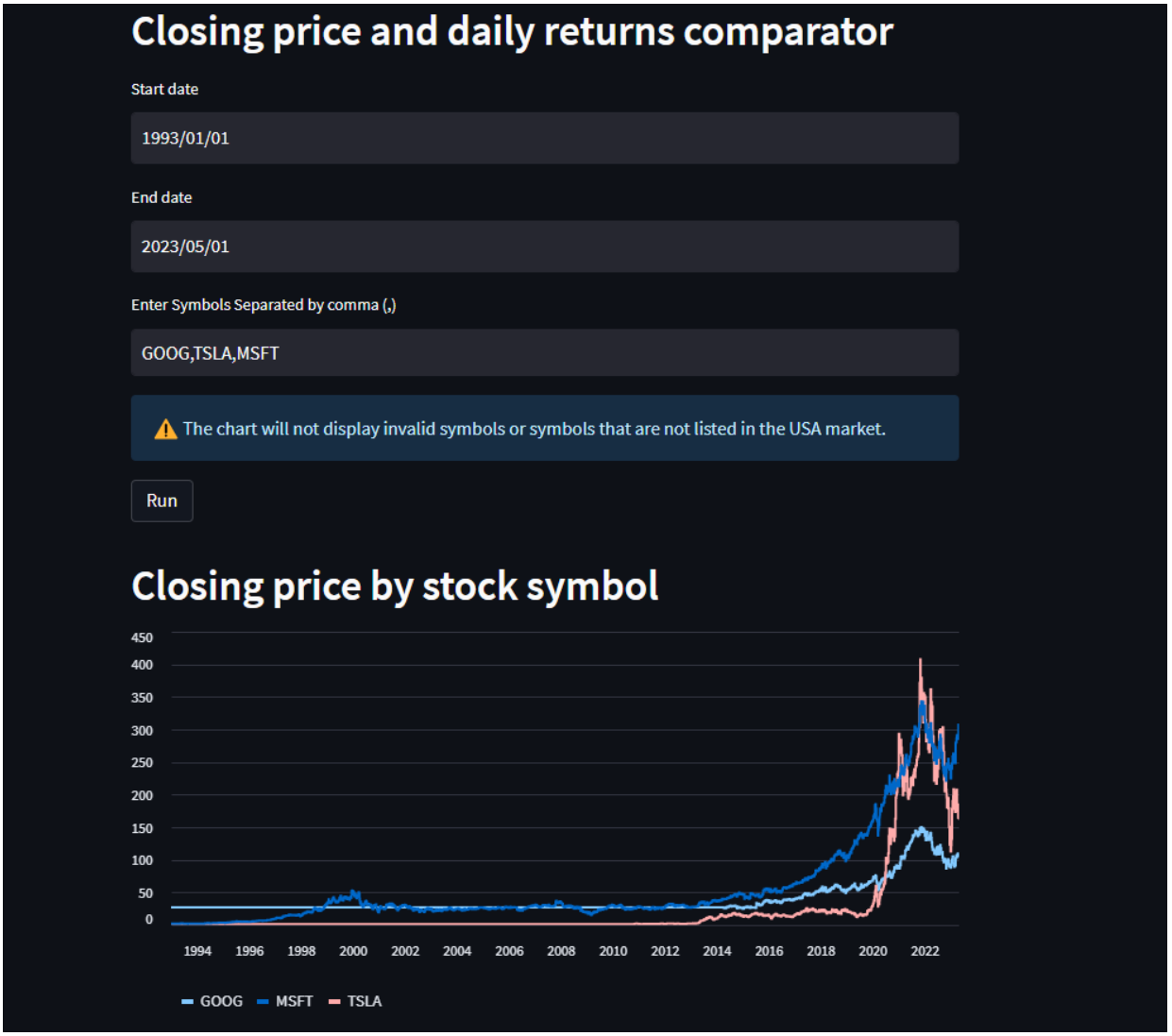
Results

Model Evaluation and Validation

During development, a validation set was used to evaluate the model.

The error metrics MAE and RMSE will depend on the stock symbol data that was used to train the model. Because some companies have more data than others and it affects the performance.

I successfully deployed the web application that allows the user play with the stock symbols and get the predictions. try it out on the link https://paulowiz-udacity-stock-predictor-main-3gk95u.streamlit.app/\



Closing price predictor with Prophet model

More info about Prophet: https://facebook.github.io/prophet/

Why should you predict stock value?

Stock value prediction is to forecast the future price or value of a stock, which is influenced by various factors such as company performance, market conditions, and global economic trends. By accurately predicting the future value of a stock, investors can make informed decisions about whether to buy, sell, or hold a particular stock, potentially resulting in significant financial gains.

Stock Symbol

TSLA

Predict Stock

Closing price forecast - TSLA

Conclusion

Using the Prophet model to predict stock price data can be a powerful tool for investors and traders alike. By incorporating historical data, seasonal trends, and external factors, the model can produce accurate forecasts that can inform investment decisions. However, it's important to note that the Prophet model is not a guarantee of accuracy and should be used in conjunction with other market analysis techniques. Additionally, the refinement process is critical to improving the model's accuracy, and it's essential to carefully evaluate and adjust the model's hyperparameters and external regressors to achieve the best results. Overall, with proper refinement and evaluation, the Prophet model can be an effective tool for predicting stock prices and informing investment decisions.