Price Forecasting of S&P500 Using Time Series Analysis

Overview

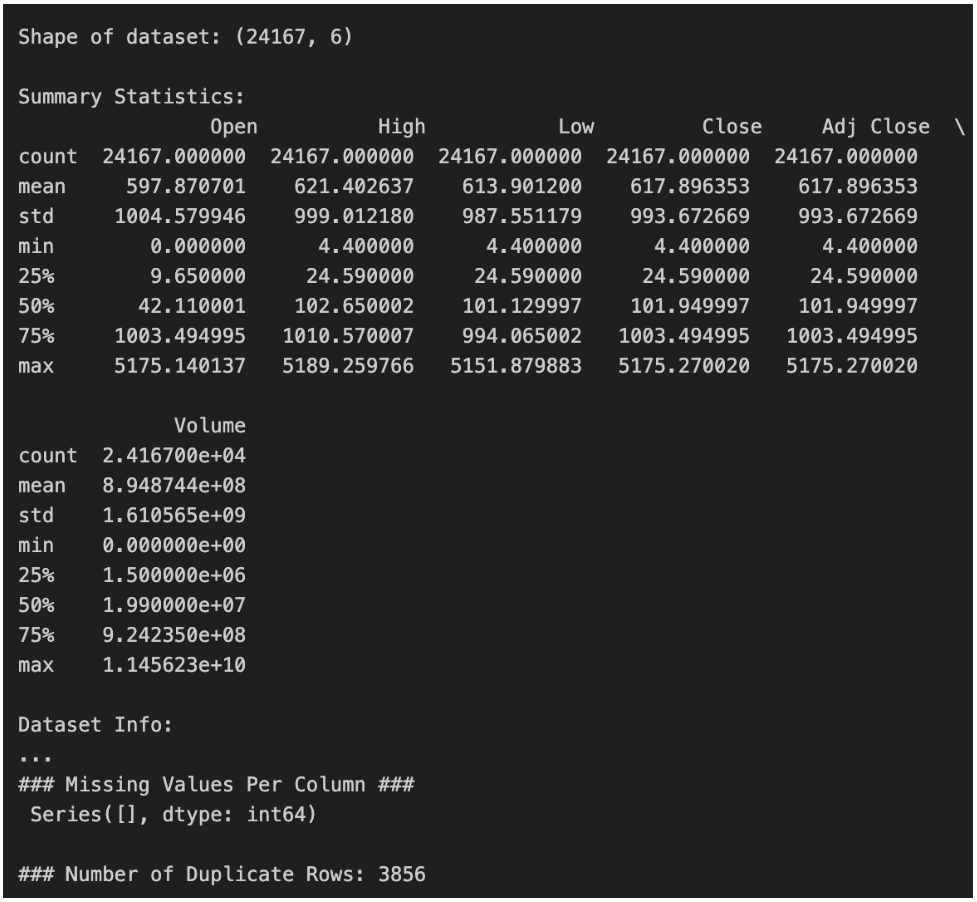
Exchange Traded Funds (ETFs) such as Standard & Poor's 500 (SPY, also SPX) are one of the most traded commodities on the stock market since they offer an easy way to own a relatively diversified portfolio for investors while owning multiple stocks at the same time. S&P500 is the largest ETF in the world and has long been associated with helping predict the direction of the economy. Having a sense of how the price of this ETF could change would allow anyone to be prepared for possible future economic changes and trends. The SPX dataset compiled and maintained by Thomas Misikoff on HuggingFace for the daily closing price from 1928 to 2024 is thus a perfect candidate for time series analysis and predictive modeling of the ETF.

Prior Work

S&P500 has long been studied using statistical methodologies from all aspects such as Chaos Theory, Volatility Analysis, various kinds of Group Analysis, and even causality tests. Almost all possible methods of analyses have been used on the dataset such as Linear and Nonlinear Regression, Deterministic Processes, Frequency Domain Analysis, and even Neural Networks however, the literature body has been historically dominated by the traditional tools of time series analysis i.e. ARIMA, SARIMA and even SARIMA using GARCH methods. The recently developed deep learning techniques, from RNNs and LSTMs to more novel and complicated tools such as Prophet, universally outperform almost all classical forecasting techniques and so have quickly replaced them.

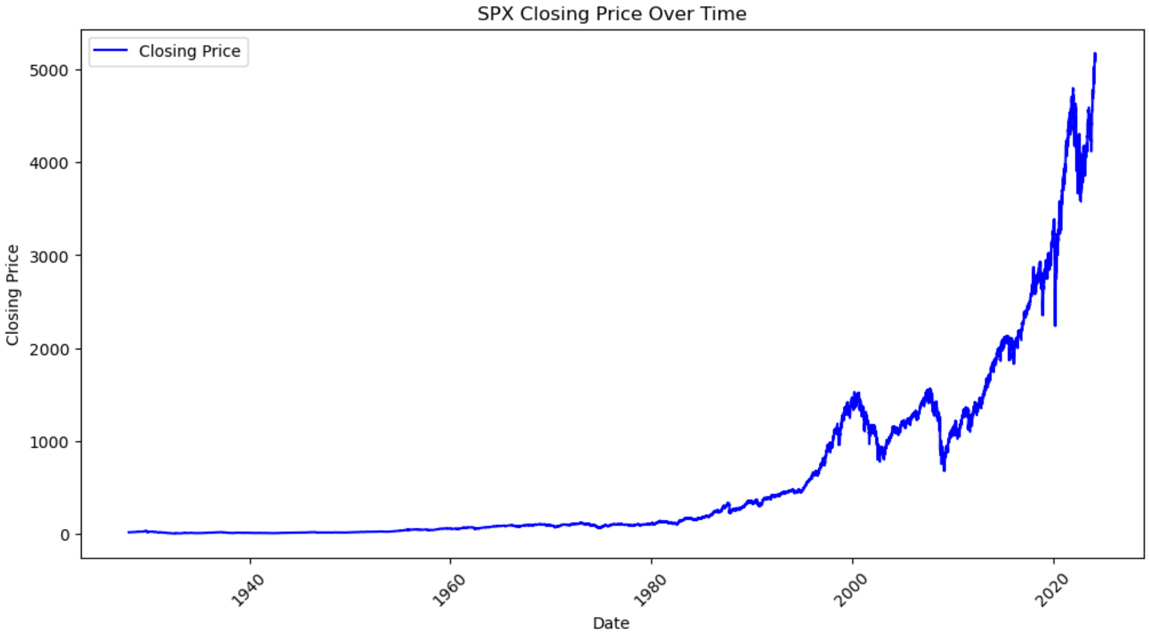
Preliminary Analysis

The SPX dataset from Thomas Misikoff consists of 24167 rows of daily Open, Close, High, and Low prices along with Volume from Dec 30, 1927 to Mar 15, 2024. There are no missing values in the dataset and the distribution of the data is as follows:

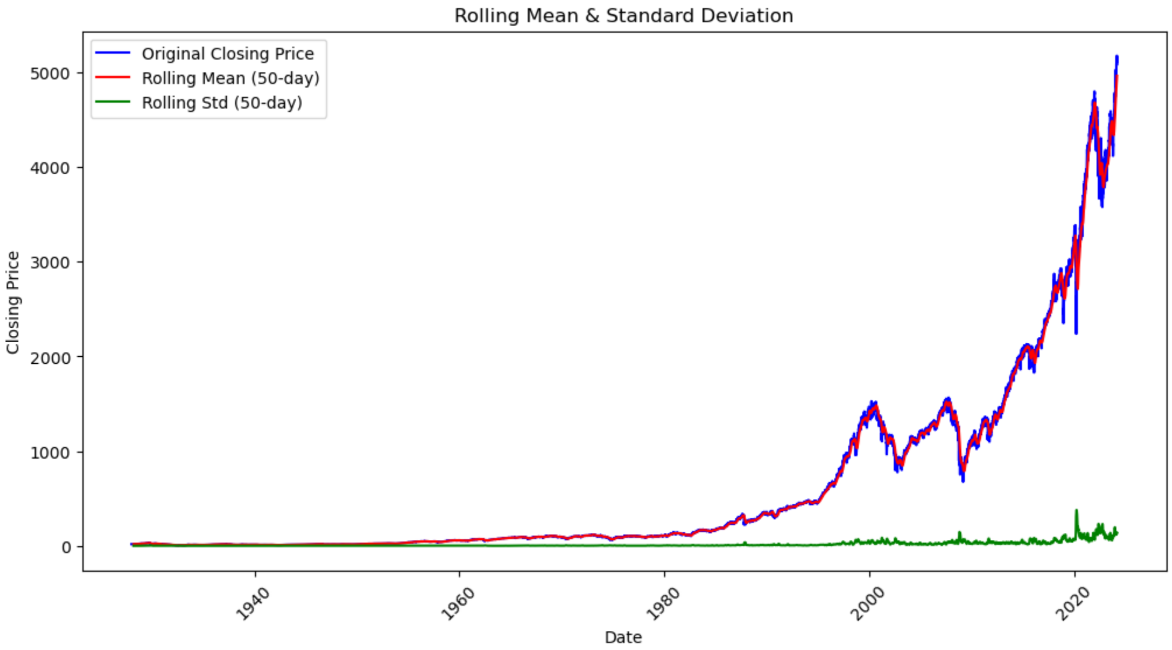


```
Shape of dataset: (24167, 6)

Summary Statistics:
                Open          High           Low         Close     Adj Close  \
count  24167.000000  24167.000000  24167.000000  24167.000000  24167.000000
mean     597.870701    621.402637    613.901200    617.896353    617.896353
std     1004.579946    999.012180    987.551179    993.672669    993.672669
min        0.000000      4.400000      4.400000      4.400000      4.400000
25%        9.650000     24.590000     24.590000     24.590000     24.590000
50%       42.110001    102.650002    101.129997    101.949997    101.949997
75%     1003.494995   1010.570007    994.065002   1003.494995   1003.494995
max     5175.140137   5189.259766   5151.879883   5175.270020   5175.270020

             Volume
count  2.416700e+04
mean   8.948744e+08
std    1.610565e+09
min    0.000000e+00
25%    1.500000e+06
50%    1.990000e+07
75%    9.242350e+08
max    1.145623e+10

Dataset Info:
...
### Missing Values Per Column ###
 Series([], dtype: int64)

### Number of Duplicate Rows: 3856
```

We can see that the dataset has a decent number of duplicate rows so one way to deal with this would be to delete those rows since there are 24K data points so we would still have a decent amount of data.



When we plot the Closing Price (Close) of the index, we clearly see that the price has risen almost exponentially as we approach the late 1990s. The above trend seems to suggest that Prophet might be our best tool to model such kind of long-term erratic behavior. However, the price is still consistent in short term and that is evident by the rolling 50-day window plot for the mean.



The basic model used will be the Long-Short Term Memory model to get a baseline estimate. We will then use Recurrent Neural Networks (RNNs) to better learn the explosive rate of growth and then finally use Prophet to see if it is able to predict the trends with a good degree of accuracy. These hypotheses will be tested by using a train-test split of 80% training points and remaining testing points.

Various tools from class will be used. Matplotlib and Seaborn will be used for visualizations. The entire implementation will be in Python. Pandas and NumPy are two of several possible libraries that will be used. Deep learning models and libraries will be used, including TensorFlow and Keras.

Project Deliverables

A successful project will produce a comparative analysis of the three models for the financial analysis of SPX. High-quality visualizations, including time series plot, model performance metrics as well as future estimates will be presented.

The process for this includes creating a preprocessing pipeline for handling and possibly normalizing the dataset, training and evaluating the models, analyzing the models using metrics such as RMSE or MAE and then developing visualizations to communicate trends, patterns, and model performance effectively.

Timeline

Week 1-2: Conduct exploratory data analysis to identify trends and patterns. Preprocess the data, including normalization and possible feature engineering.

Week 3-4: Train initial models. Fine-tune the models and conduct hyperparameter tuning. Compare performance metrics across all models.

Week 5: Create visualizations. Prepare the final report and GitHub repository, ensuring all results and visualizations are well-documented.