Forecasting Adjusted Closing Price Utilizing ARIMA

Jalin Roberts | Dr. Robert Kelley

Bellarmine University

ABSTRACT

The Predictive Analytics Model outlines a project focused on forecasting stock prices using the ARIMA model, specifically targeting the Adjusted Closing Price. It begins with data preparation, leveraging Python libraries like Pandas, SciKit-Learn, and Statsmodels, and emphasizes using only adjusted closing prices for precision. The project, challenging for the author, involves predicting stock prices based on historical data from 2018-2022 to forecast for 2023. Although the ARIMA model is central to the analysis, offering visual and textual predictions, the author candidly discusses the complexities encountered and the model's limitations, such as not indicating time series stationarity. This document represents a comprehensive effort to apply ARIMA modeling for stock price forecasting, reflecting both the analytical process and the learning journey involved.

Objective: Automated Model Calibration for Optimal Predictive Accuracy

•**Accomplishment**: Implemented a grid search algorithm to identify the best (p, d, q) parameters for each time series variable, ensuring optimal model calibration.

Objective: Data Segmentation into Training and Testing Sets

•**Accomplishment**: The dataset was systematically divided into training and testing segments, with the training set used for model calibration and the testing set (for validating forecasts.

Objective: Performance Metrics and Visualization

•**Accomplishment**: Calculated key performance metrics such as RMSE, MAE, and APE. Additionally, implemented visualizations like time series plots, ACF/PACF plots, and forecast plots with confidence intervals.

Objective: Out-of-Sample Predictions with Confidence Intervals

•**Accomplishment**: Extended the forecasting ability of the models to predict out-of-sample data and included confidence intervals for these predictions.

Objective: Residual Analysis and Record Keeping

MATERIALS AND METHODS

Materials:

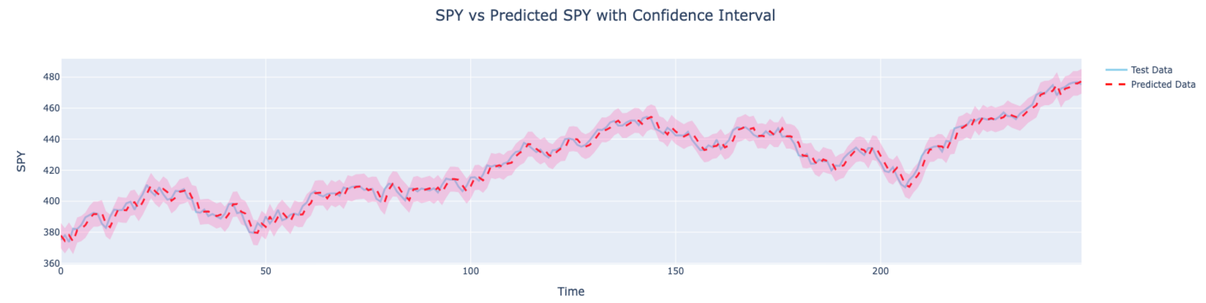
- **Data Source**: The primary datasets contained in this project were acquired from Yahoo Finance.
- **Software and Libraries**: The analysis is conducted entirely in Python, with libraries to aid in data manipulation, statistical modeling, visualization, and performance measurement. These include: Pandas, NumPy, Statsmodels, SciKit-Learn, Matplotlib, Seaborn, Plotly.
- **Data Preparation**: The initial step involves loading the dataset from the .csv file and parsing the 'Date' column to ensure it's in the correct datetime format. The dataset is then set with 'Date' as the index, focusing on the Adjusted Closing Price.
- **Model Building**: The ARIMA model, specifically chosen for its suitability in forecasting time series data, is the core analytical method. The project details the process of fitting this model to the historical stock price data.
- **Model Evaluation**: The notebook mentions the use of specific model performance metrics, likely mean squared error (MSE) or mean absolute error (MAE), though specific details on the evaluation process are not provided in the abstract. The evaluation aims to assess the accuracy of the stock price forecasts generated by the ARIMA model.
- **Visualization**: Throughout the project, various plots and visualizations are created to illustrate the data, model forecasts, and possibly diagnostics related to model fit, such as autocorrelation plots. These visual aids are crucial for interpreting the model's performance and understanding the time series data.

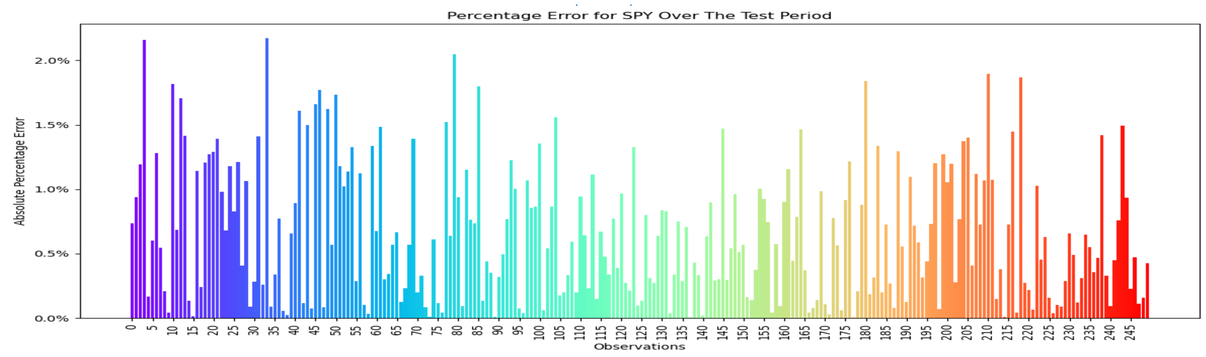
Methods:

- **Model Fitting**: The script performs a grid search over possible combinations of (p, d, q) orders, fitting an ARIMA model for each combination.
- **Stationarity Test**: The script uses the Augmented Dickey-Fuller (ADF) test to check if the residuals of the fitted model are stationary.
- **Selection Criterion**: The Akaike Information Criterion (AIC) is used to select the best model. A lower AIC value indicates a better model fit. This is used as the criteria focus on the forecasting and predictive power of the model
- **Autocorrelation Test**: For the best model, the script generates plots of the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF). These plots help in understanding the correlation structure of the time series.
- **Error Metrics**: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are calculated to evaluate the prediction accuracy. Absolute Percentage Error (APE) is also computed for each forecasted value.
- **Forecasting**: The script uses future values using the best fitted ARIMA model and compares these forecasts with the actual values.

```
Prediction vs Observed Values:
predicted=66.773451, expected=66.259476
predicted=66.325125, expected=66.484879
predicted=66.493346, expected=66.308487
predicted=66.329701, expected=68.111732
predicted=67.961536, expected=66.994499
predicted=67.072933, expected=67.141510
predicted=67.234234, expected=67.631516
predicted=67.509913, expected=68.121521
predicted=68.166053, expected=68.660538
predicted=68.538465, expected=67.670723
...
```

➢ The above values are a small sample used for training and testing.





RESULTS

This ARIMA model provided accurate forecasting and results to ten (10) individual stocks. The training data for this project ranged from 2018 to 2022, with the goal of predicting 2023 Adjusted Closing Price as precisely as possible. Below are the results and explanations for one stock from this model(SPY). Note: SPY is also referred to as the SPDR S&P 500 ETF Trust.

ADF Statistic: -71.06596155287077

p-value: 0.0

- A very negative ADF statistic rejects the null hypothesis (non-stationarity), and a p-value of 0 indicates that the test results are highly significant.

Ljung-Box Test: 0.8452943584039303

p-value: 0.9999208345871031

- Ljung Box Test checks for autocorrelation at multiple lag levels. This value indicates all of the information in the data has been captured by the model.
- Extremely high p-value (close to 1) indicates no significant evidence of autocorrelation.

Best ARIMA Structure for SPY is: (2, 1, 2)

- This project dynamically fits each stock to the best ARIMA structure.

Best AIC found is: 8494.006732657283

- For this ARIMA Structure (2, 1, 2) this AIC is the lowest value attainable. The lower the AIC value, the better the model is.

The RMSE of the predictions is: 3.472202538646774

- On average, the model's predictions are about 3.472 units away from the actual data points, this is very accurate. The goal for this value is to be as low as possible.

Mean Absolute Percentage Error of the prediction is: 0.66%

- This value indicates that the model's predictions are on average less than 1% away from the actual values in percentage terms. This is extremely accurate.

CONCLUSIONS AND FUTURE WORK

Conclusion:

The ARIMA model project successfully achieved its key objectives, establishing a robust predictive model tailored to the nuances of time series data. The project's emphasis on automated calibration, rigorous evaluation, and comprehensive reporting ensures that the final models are both accurate and user-friendly. These models are well-equipped to provide insightful forecasts, aiding in informed decision-making and strategic planning.

Future Work:

- This model serves as a preliminary step to a larger goal of autonomous investing.

REFERENCES

- https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp#:~:text=An%20autoregressive%20integrated%20moving%20average%2C%20or%20ARIMA%2C%20is%20a%20statistical,values%20based%20on%20past%20values.
- https://towardsdatascience.com/what-is-an-arima-model-9e200f06f9eb
- https://coolstatsblog.com/2013/08/14/using-aic-to-test-arima-models-2/
- https://www.projectpro.io/article/how-to-build-arima-model-in-python/544
- https://stackoverflow.com/questions/33661762/python-statsmodels-arima-forecast?rq=3
- https://machinelearningmastery.com/grid-search-arima-hyperparameters-with-python/

CONTACT

Email: jroberts12@bellarmine.edu