**Stock Market Prediction Tool**

CS 455 - Artificial Intelligence

Final Project

Cameron Stark

Luis Olivier

Dustin Cribbs

**Introduction**

Stock market price prediction is a concept that many developers and analysts have tried to perfect. The difficulty in this is that there are many variables that can shift the price changes on a daily basis. The volatility of the prices makes it very difficult to predict with high accuracy which is why it is an intriguing problem. There are many different ways to tackle this problem and the one that was chosen for this project is a linear regression analysis. This is a linear approach to modeling a relationship between a dependent variable and one or several independent variables. The way to do this certain approach is to use historical data of the stocks and their prices from previous years on specific days. In simple terms, the model could use data from April 20th from the years 2009-2019 to predict the price of stock X on April 20th 2020. The way we approached this issue is by first acquiring a large data set of stocks, each company having its own .txt file with its respective data in columns that consist of; Date, Price at Open, Highest price that day, Lowest price that day, Closing price, and volume. We implemented our solution using Python and Visual Studio as our development environment and some of the libraries used included Pandas, Numpy, and most importantly, the Sklearn library. We were able to work together in real-time using Visual Studio’s live share feature. Instead of loading all of the stock data everytime the code is run, we decided to allow the user to choose which stock(s) and how many days of forecast they would like to perform the regression on, this way there are only one or several files loaded onto memory instead of the whole data set. When the regression is performed, the file is sorted in ascending order by date and the training data set gets prepared to execute. After the model is trained it will run the prediction model and plot the results for the user in a browser tab.

**Approach**

The dataset is provided from the Kaggle website which contains numerous amounts of datasets used for data processing or application development. The data provided is not modified in any way such as pre-processed in any way for ML implications. The data is a series of txt files for every available stock up to the year of 2017/2018, with values for each day the market is open which values such as date, open price, high price, low price, close price and volume for that day. Upon selection of the data by the user, all columns are dropped except date and close price.

The design of the software is in two parts the interface portion runs in its own file, asking for the user input and starting the regression process by sending over the list of stocks, the number of days to forecast out, both provided by the user. Once the main file calls the algorithm file the program remains there by iterating through the stock array/list provided by the user, each iteration the stock is loaded from the txt file and the split in half (to improve final readability of the graph) with taking the bottom half of the data, which goes from some date to the most recent dates, after that the data gets splitted into training and test data, which was then passed into the fit and prediction functions for the Linear regression model. The forecast value provided by the user tells the program how many data points to predict out from the dataset for the possible future values of the stock. After the prediction is acquired it is passed to the Plotly library to plot the graph on a time series plot, containing the data and the predicted data. Upon each iteration's prediction on the stock data, a new local host window is opened in the browser for each stock that is requested.

|  |  |
| --- | --- |
| **File** | **Purpose** |
| main.py | Gets user input and starts process |
| algorithm.py | Takes the inputs to predict and plot stock values |
| Stocks | Folder that contains stock price changes for each stock |

**Results**

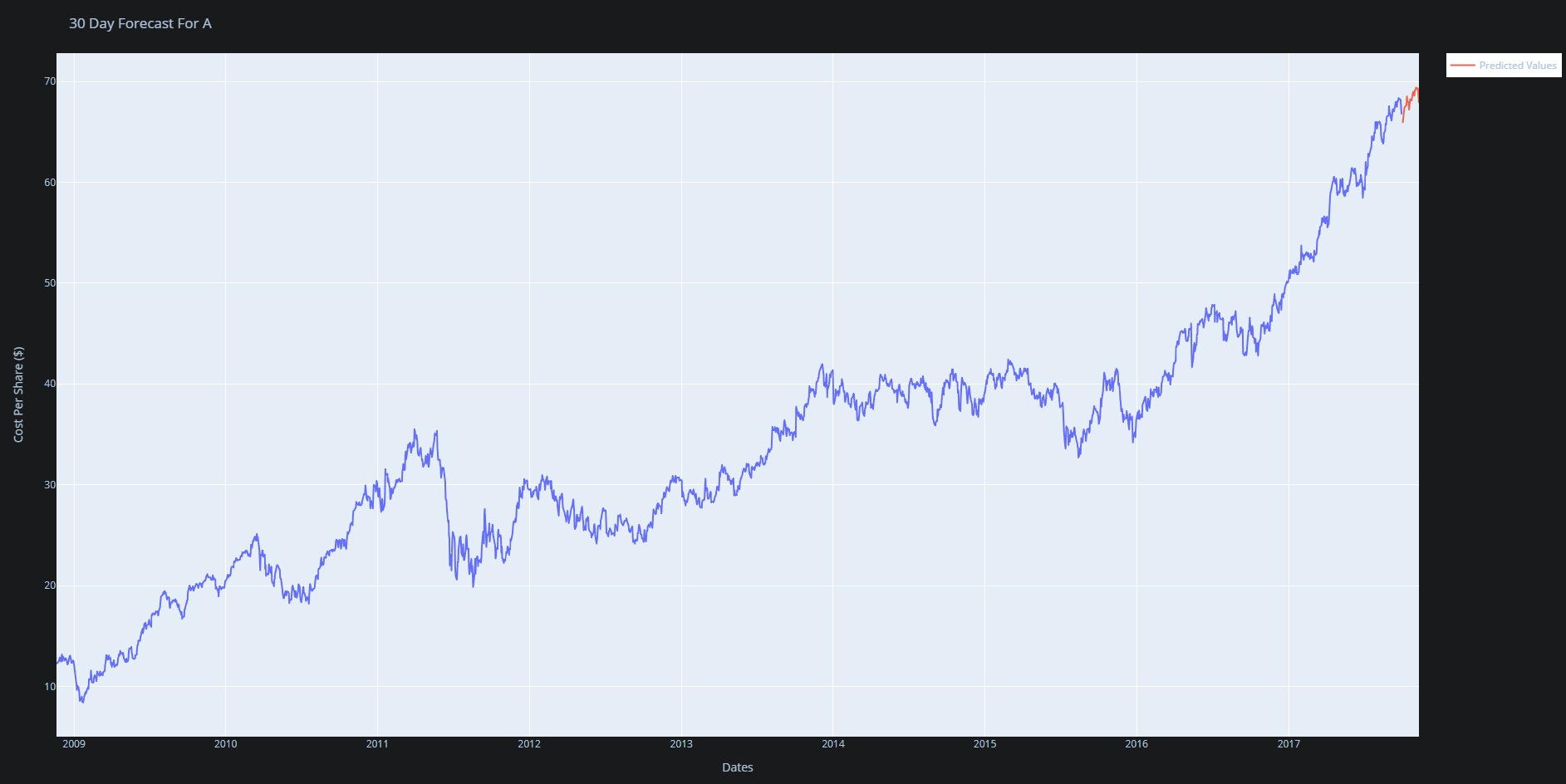
The project with the scope we laid out was successful in taking the user inputted multiple stocks in question and a forecast of days they would like to see future potential prices for the stock. The program takes the user input and converts the comma list to a list it can iterate through and the forecast days is used to create the length of the array of data that is used to predict the future values.

The criteria laid out were the need for user interactivity, predicting of values and plotting for the user to visualize. The interactivity was satisfied by the requesting of a list of Stocks in question and the number of days to forecast out the prediction of stock prices. The predicting of values was satisfied with the using of a linear regression to predict values based on the regression line that it creates. The plotting requirement is satisfied with the use of the time series plots in the Plotly library which makes use of several javascript libraries like D3.

The only main issue with the data is the predicted and last known value don't completely line up in the plot for some instances, so that is something that can be improved upon in future addition and the combining of the different plots for multiple stocks into one plot for comparison reasons.

**Conclusion**

To conclude the project, the results from the Linear Regression to predict the prices of the provided stock day closing values for previous days came out as expected and followed relatively the trend of the regression line to the data.



The technical lessons from this project were capable of changing libraries, models, and in our case data. We were initially doing a aircraft maintenance scheduling programming, but there was no pre made data for that, so we switched to engine degradation data, but ended up switching from that because some advanced math would need to be done to the data before being able to put into the models, so we switched to a stock pricing dataset i found on Kaggle and decided to implement a regression model to the data and provide the ability for the user to select stocks to look at and predict the future stock price

Future additions that could be made would be provide multiple potential predictives per stock plot and provide the ability for stocks next to each other in same window or plot for trending comparison.

**Github link:**

[**https://github.com/starkc1/CS455\_Project**](https://github.com/starkc1/CS455_Project)

**References**

“Essentials of Linear Regression in Python,” *DataCamp Community*. [Online]. Available: https://www.datacamp.com/community/tutorials/essentials-linear-regression-python. [Accessed: 23-Apr-2020].

B. Marjanovic, “Huge Stock Market Dataset,” *Kaggle*, 16-Nov-2017. [Online]. Available: https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs#aaoi.us.txt. [Accessed: 23-Apr-2020].

A. Singh, “Predicting the Stock Market Using Machine Learning and Deep Learning,” *Analytics Vidhya*, 04-Sep-2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/. [Accessed: 23-Apr-2020].

randerson112358, “Predict Stock Prices Using Python & Machine Learning,” *Medium*, 18-Dec-2019. [Online]. Available: https://medium.com/@randerson112358/predict-stock-prices-using-python-machine-learning-53aa024da20a. [Accessed: 23-Apr-2020].

**Appendix**

**Dustin:**

On the project I worked on the project report, finding the algorithm, and preparing the data from the dataset. I learned that finding an algorithm that works is very important. We tried to use a different algorithm before using linear regression, but the data just wouldn't fit properly and it was difficult to work with so I did some more research and found that linear would work best. It showed me the importance of how if you have the wrong algorithm and the fit is off you won't get the results you're trying to achieve. The second thing I worked on was getting the data ready to be used. This posted a bit of a time consuming effort as there is a lot of data. The dataset we used uses every publicly traded stock with data on every stock inside an individual text file, but luckily it was already organized to some degree. This project has shown me how useful machine learning type algorithms can be in practical examples like stock predictions. I think this class should be mandatory for CS students as their future will be shaped by these algorithms and their predictions.

**Cameron:**

On the project I worked on implementing the Linear Regression model, User input, Loading data and the Plotting of the data in the python program. Added these by means of the online documentation for python as a whole is the I/O components, the sklearn documentation for the linear regression components, and the plotly documentation for the plotting api documentation. From working on this I learned the many applications that having machine learning and knowing how to implement machine learning can affect or improve projects that i work personally or at whichever company I end up working at. I also worked on the report specifically the results and conclusion of the report, where i stated the various goals that were successfully met for the report and the how reaching those goals taught various aspects of python and machine learning.

**Luis:**

On this project I worked on setting up the program to load the data to be regressed, I helped out with code wherever needed as well as with the research conducted to find what we will do our project on. I also worked on the report by doing the Introduction section, references and helping with the formatting. From working on this project I learned how to effectively set up environments to develop code, we came across many issues when beginning to set up vscode and git to collaborate. By researching and communicating effectively with my team we were able to come up solutions and changes to get our files to work with each other. I also learned how to choose algorithms by weighing pros and cons and seeing what fits our application best. Working with python in this project also improved my python skills and working with algorithms such as linear regression and KNN classification which our project was going to originally be.