Abstract

The goal of the project is to create an advanced model that formulates stock predictions based on past economic data. Models such as these have become commonly used for trades by large firms because of the quick and mathematical thinking the algorithm can use to predict small, short term changes in stock prices. Sixty to seventy five percent of trading is done through algorithms using artificial intelligence which accounts for hundreds of billions of dollars of transactions every single day. The strength of the algorithm is only as good as the people who build it, so understanding how the market works is important for judging whether the algorithm works how it should. Using different machine learning algorithms, technical information about stocks is fed in to try to produce low amounts of error in stock price predictions. Based on top technical metrics used for trading determined by Bloomberg(an investing software), evaluation of the market as whole, and buyer sentiment analysis through different news sources, the algorithm will try to formulate a pattern that best predicts the outcome for the next day. The best results achieved by the different algorithms were .8% error, or average difference between predicted and actual stock price, and 1.1% error.

The most advanced stock prediction models take into account fundamental and technical analysis. Fundamental analysis involves analyzing the stock's intrinsic value, financial statements, tangible assets, management effectiveness, consumer behavior, and overall company outlook. Including this type of analysis is much harder than technical analysis because a lot of fundamental analysis is subjective and disagreed upon by many people. Short term statistics and news do not affect fundamental analysis because the predictions are more long term.

Instead of directly predicting stock prices, many companies use AI to more effectively and efficiently sort through millions of data points. The algorithms can help make technical analysis faster by letting the algorithm determine the positive or negative trajectories based on the data fed into it. For example, GreenKey technologies uses AI speech recognition and natural language processing to look through financial conversations and documents to try to recognize trends. Kavout corporation uses their software and machine learning algorithms to produce rankings of stocks for the day. Then investors can view the rankings and see if anything seems interesting to them which helps save them time in investigative research. Auquan, a data processing software, actually allows data analysts to create their own algorithms using data sets and their own knowledge, helping to democratize machine learning. AI technology is becoming widely used in finance and stock market predictions and the market is only becoming bigger.

Using a linear model vs. Neural Network

A linear model is less complex and just creates a prediction based on the weights it gives the different variables. A neural network takes more run time and computing power and can also lead to overfitting in many instances if the network has too many iterations. A neural network also has more flexibility than a linear model since the number of layers, nodes, and iterations can be adjusted.

Results

The best accuracy achieved on the linear model was about 1.1% difference from the actual price of apple stock. The variables taken into account for this model were the last five days, the price of oil, the price of the S&P 500 and the closing price of apple the previous day. The best accuracy achieved for the neural network was also around about 1.1%, but the layers, interactions, nodes, and other parts had to be adjusted to achieve this result.

Why chose the datasets

Checking the last few days helps the algorithms to spot trends or patterns based on how the stock recently moved. For some stocks checking the last three days works the best while for others checking the last ten might work the best. Also, the use of the price of oil helps calculate inflation worries which have an impact on many stocks. The S&P 500 index fund is used to model the market direction as a whole which could be helpful in determining where the stock will move. Also, the closing price gives a good idea of where the stock will be around at the open with the other factors helping to predict too.

Set up

The algorithms used for this project are run in google collaboratory notebooks and will use data found in different investing databases including Yahoo Finance. First all of the libraries have to be imported including matplotlib.pylot, datasets, linear_model, MLPRegressor, train_test_split, different ways to calculate errors, Yahoo Finance, pandas, and numpy. Then, using Yahoo finance, data frames are created which include the open price of a stock and prices of other parts of the market including oil for the past three and a half years. After that, two arrays are created called “X” and “y” where X includes everything the algorithm takes in and then y is the price for the next day. In order to train the algorithm, the data is then separated into train and test data using “train_test_split” which uses 75% of the data to train and 25% to test the accuracy. The first three weeks of a month are used for training and the last for testing. The model is created using a line of code, and then “fit()” is used to train the model on the X_train and y_train data. The prediction is created using “.predict(X_test)” which is where the algorithm actually creates the predictions based on the parameters and weights. Then the error is calculated by taking the absolute value of the prediction minus the actual value, dividing that by the total number of predictions, and then dividing that by the stock price to make sure the error doesn’t look higher for more expensive stocks.

Baseline Model

For the baseline model a linear model is used which uses linear regression to calculate stock prices. The model takes in values such as previous stock prices, the price of the S&P 500 the day before, and the price of oil the day before to create a prediction. Each of these values is assigned a weight to determine how important they are in calculating a future price. The combination that worked the best was taking the last five days into account with these weights from 5th to the most recent day: -.05261678, .08510924, .00577066, .04591285, .91503032. Also, weight were attributed to the price of oil and the price of the S&P 500, but the weights were relatively small. The best error was about 1.4% of the price of the stock being used. The companies Apple, Tractor Supply, and Tesla were tested on.

Neural Network model Model

A neural network model is tested to compare to the results of the linear model. To create the neural network we changed the type of model and trained it differently. The model goes through five thousand iterations, which are different times that the model adjusts its weights to create the best prediction. The weights are the values that are between nodes to output a new value on each node. Also, the number of nodes can be adjusted. Dropout is used to minimize overfitting, which is when the model works too closely for the training data, but not the test data. After changing the nodes, using dropout, and

Overfitting

Overfitting occurs when a model finds a pattern in training data that causes the model to predict very well on the training data, but then the pattern is not present in the testing data, so the model does not perform well when tested. The way to tell if there is overfitting is by comparing the train and test error. If the train error is significantly higher than the test then there is overfitting. Dropout is one way to solve overfitting with neural networks since it will randomly get rid of certain pathways so an algorithm cannot find a path or pattern that always works, but has to take into account all of the paths. This prevents overfitting since the algorithm will not figure out a very specific pattern that only works on train data and not test data, and will instead create a more general pattern.

Conclusion