Survey of Machine Learning Techniques for Stock Market Pattern Detection and Prediction

Achsah Ledala, Ikechukwu Uchendu, Luke Stanton, Riley Annis

Abstract

Every Monday through Friday, excluding holidays, people have the opportunity to purchase shares of publicly traded companies at the New York Stock Exchange. Each share is worth some dollar amount depending on how well the company is performing. Throughout the day, the value of a share will either increase, decrease, or stay the same. Buying a share of a company one day and selling it for more on another would result in a net profit. In this paper we explore how different machine learning techniques can be leveraged to pick stocks that can be sold for profit. Studying and analyzing past data and using it to make predictions will help investors make informed decisions about buying or selling stocks thus giving them an edge in the market.

The main data set we explored in our project was the stocks that make up the S&P 500 from 2013 to 2018.

Introduction

A great deal of research is undertaken before purchasing an item, be it be a car, house, etc, to make the right decisions. It is even more so in the case of the stock market. To avoid huge losses, researching stocks before investing in them is of great importance. Analysis of stocks relies on utilizing past stock prices to predict future trends. The analysis is useful as it helps to understand the behaviour of the stock in the short term as well as long term period. This information is useful to make informed decisions to maximize the return on the stock investment.

Problem Statement

The problem we are trying to address is the idea of developing some sort of model that can accurately predict the behaviour of the stock on the stock market. The financial gains someone could make with a model like this would be endless.

Numerous publications can be found that use different machine learning techniques to attempt to predict stock prices. We have implemented techniques discussed in two articles, where researchers apply machine learning techniques such as linear regression, k-nearest neighbors, moving average, and more in attempts to create models to predict stock prices.

There are a few problems we aim to solve in this project:

1. Given that a user has a certain stock, we want to predict if they should buy or sell it to maximize their return on a given day.
2. Given stock data from a certain company, we wish to predict its closing price given the opening price.

Technical Approach

Regression:

Moving Average:

Moving Average technique helps to gauge the direction of the current trend. The predicted closing price is calculated based on the average of the latest set of previously observed values. As new data values becomes available the old data values are dropped from the calculation and replaced by the new data values. The resulting average when plotted shows the smoothed data trend rather than focusing on the day to day fluctuations that are inherent in the financial markets.

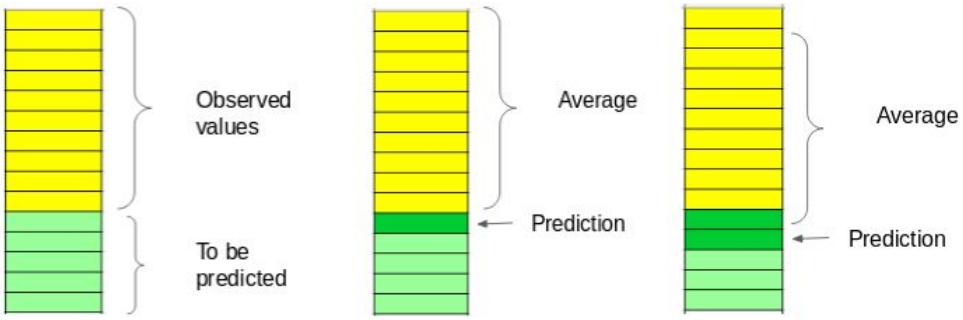

Illustration of Moving Average method

Reference:https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/

Linear Regression:

A linear regression line is used to determine trend direction.In predictive analytics linear regression is used to predict a future numerical value for a variable. In our algorithm, the independent variable is the date and the dependent variable will be the price of the stock. The goal of the process is to find the best-fitting line that minimizes the sum of squared errors with the actual value of a stock price and our predicted stock price over all the points in our dataset.

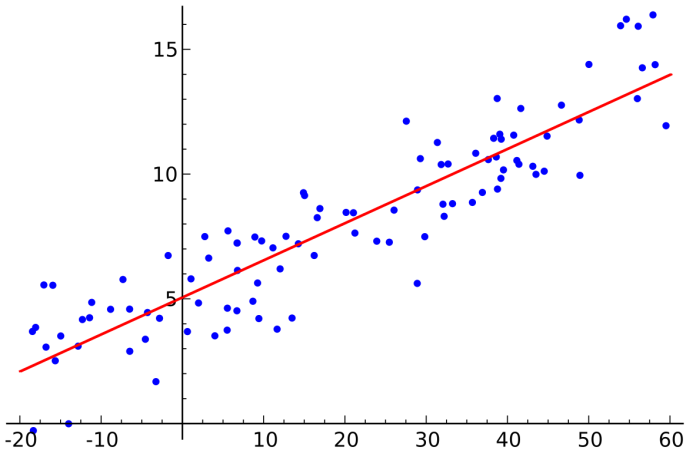

Illustration of linear regression

Reference:https://www.google.com/search?q=linear+regression&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjN9s3L1oLiAhUORKwKHQfYCoMQ_AUIDygC&biw=1163&bih=525#imgrc=l9uzNFRB9iqHhM:

K-Nearest Neighbors:

The stock prediction problem can be mapped into a similarity based classification. The historical stock data and the test data is mapped into a set of vectors. In order to predict a class label for unknown record, kNN selects k records of training data set that are closest to the unknown records. This means that the new point is assigned a value based on how closely it resembles the k points in the training set. The k value that we used in this algorithm is 5.

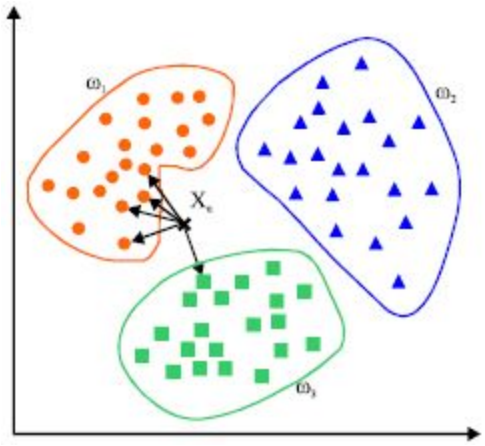

Illustration of K nearest neighbor.

Reference:https://www.google.com/search?biw=1163&bih=525&tbm=isch&sa=1&ei=TPHNXK7lOpC60PEPzreXsAI&q=knn+&oq=knn+&gs_l=img.3..35i39j0l9.29329.31165..31724...0.0..0.75.778.12.....1....1..gws-wiz-img.....0i30j0i8i30j0i24.MaefgLXs6NU#imgrc=r2kMbdhcx_W9rM:

LSTM RNN:

Recurrent neural networks are extremely useful for time-series data. A recurrent layer in a neural network propagates information along the entire layer, meaning that the output of a given neuron depends on all previous neurons in its layer. Long Short-Term Memory specifically is useful when specific positions within the time series data is relevant for the final decision, as each neuron retains information about its prior output. For our LSTM RNN implementation, we constructed a model comprised of two sequential LSTM layers of 50 units each, followed by a dense layer with a single perceptron to produce the predicted stock value.

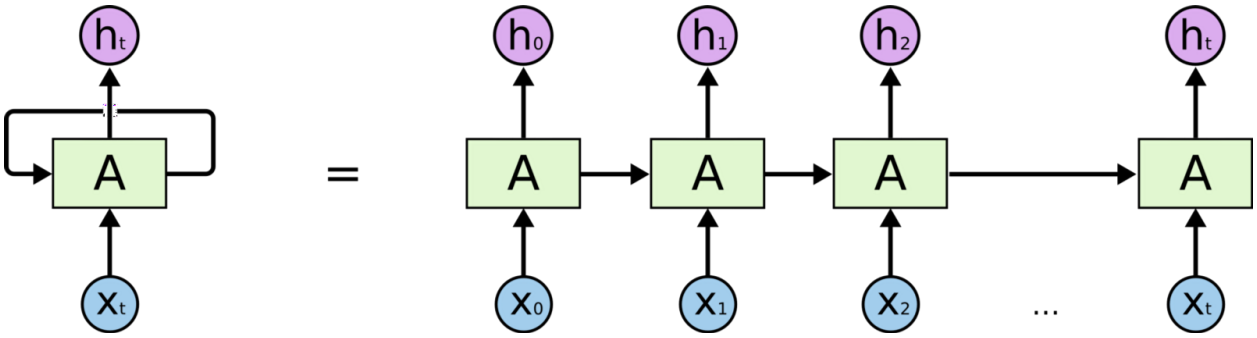

Illustration of LSTM-RNN

Reference:https://www.google.com/search?biw=1163&bih=525&tbm=isch&sa=1&ei=uPLNXKmCBLey0PEP_vOv-Aw&q=lstm+rnn&oq=lstm+rnn&gs_l=img.3..0j0i8i30l3j0i24l6.26740.27644..27958...0.0..0.70.254.4.....1....1..gws-wiz-img.....0i67.m6Z5jezk7YI#imgrc=9FjNWDtBUJlT2M:

Classification:

We used stock data to predict whether to buy or sell on a given day. The training data was preprocessed to accommodate this purpose. Given a window size, w, contiguous subsets of size w were sequentially selected from the data to form a feature matrix. The ground truth label y_i will be 1 (sell) if the opening price the day after the window ends is greater than the opening price on the previous day, and 0 otherwise (buy). For example, consider a sequence of daily opening prices given by ($25, $12, $10, $35, $19, $57). If we chose $w = 3$, we would then construct a matrix $X \in R^{3 \times 3}$ and ground truth vector $y = (1, 0, 1)$. For clarity, X is shown below. Notice that we do not include the last window of ($35, $19, $57) since there is no opening price outside the window to predict.

Example Feature Matrix		
$25	$12	$10
$12	$10	$35
$10	$35	$19

Essentially, w acts as the number of elements in a sliding window that creates feature vectors at each step. We utilize elementary classification methods such as logistic regression and linear support vector machines (SVM) to evaluate the effectiveness of this method.

Experimental Results

Classification:

Here we analyze the performance of the classification models. We created feature matrices via the method described in Section 3 based on stock data from American Airlines (AAL). The data was split into training and testing sets containing 80% and 20% of the full data set respectively. The confusion matrices for SVM and logistic regression are listed for a few different window sizes on the next page.

Notice that the classification performance was poor across the board, but especially so for the large window size. This is most likely due to the fact that the classification methods do not take into account the time series nature of our data. As the window size becomes larger, we get a higher dimensional feature vector with a smaller number of samples in comparison. It is difficult to extract patterns from that since we are looking for arbitrary trends within each feature vector.

A simple extension to our existing classification methods would be to utilize a highly accurate regression method such as the LSTM (Figure 4). The LSTM would simply need to predict the opening price of a stock on at least two consecutive days from the testing set. With this prediction, we say “sell” if the second price was higher than the first, and “buy” otherwise. This is essentially an indirect method towards classification, since it requires computing an exact estimate of the stock price.

[AAL] One Month ($w = 30$) Logistic Regression - Testing Set Confusion Matrix		
	Buy	Sell
Buy	78	28
Sell	110	30

[AAL] One Month ($w = 30$) SVM - Testing Set Confusion Matrix		
	Buy	Sell
Buy	14	92
Sell	13	127

[AAL] Three Month ($w = 90$) Logistic Regression - Testing Set Confusion Matrix		
	Buy	Sell
Buy	99	0
Sell	135	0

[AAL] Three Month ($w = 90$) SVM - Testing Set Confusion Matrix		
	Buy	Sell
Buy	99	0
Sell	135	0

Regression:

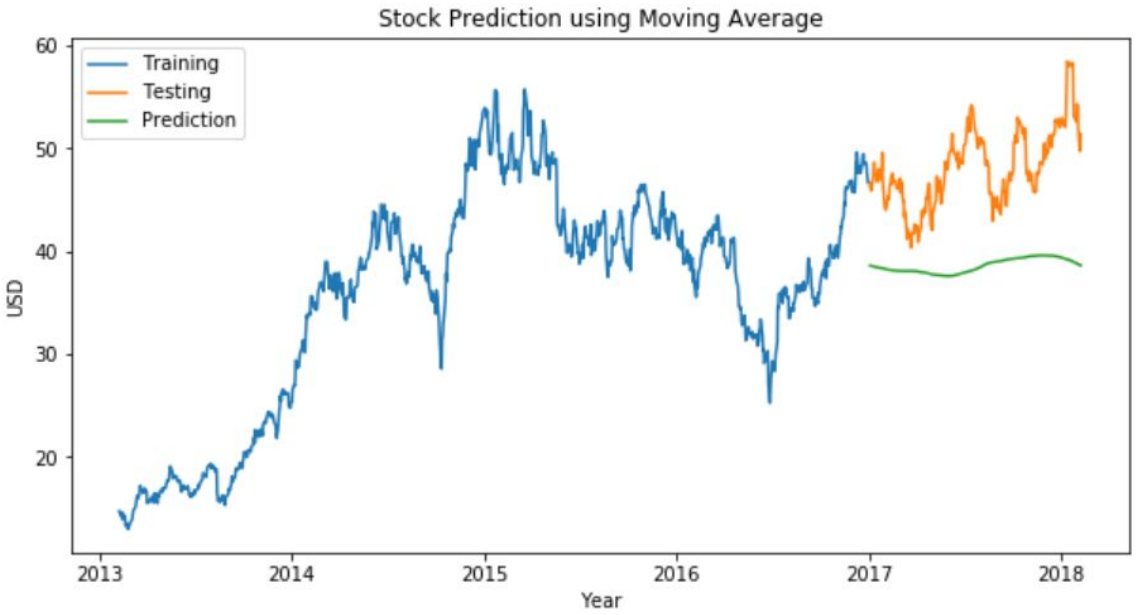

Figure 1: The above plot shows the performance of predicted prices against the actual values using Moving Average model on AAL stock's price over time.

Figure 2: The above plot shows the performance of predicted prices against the actual values using Linear Regression model on AAL stock's price over time.

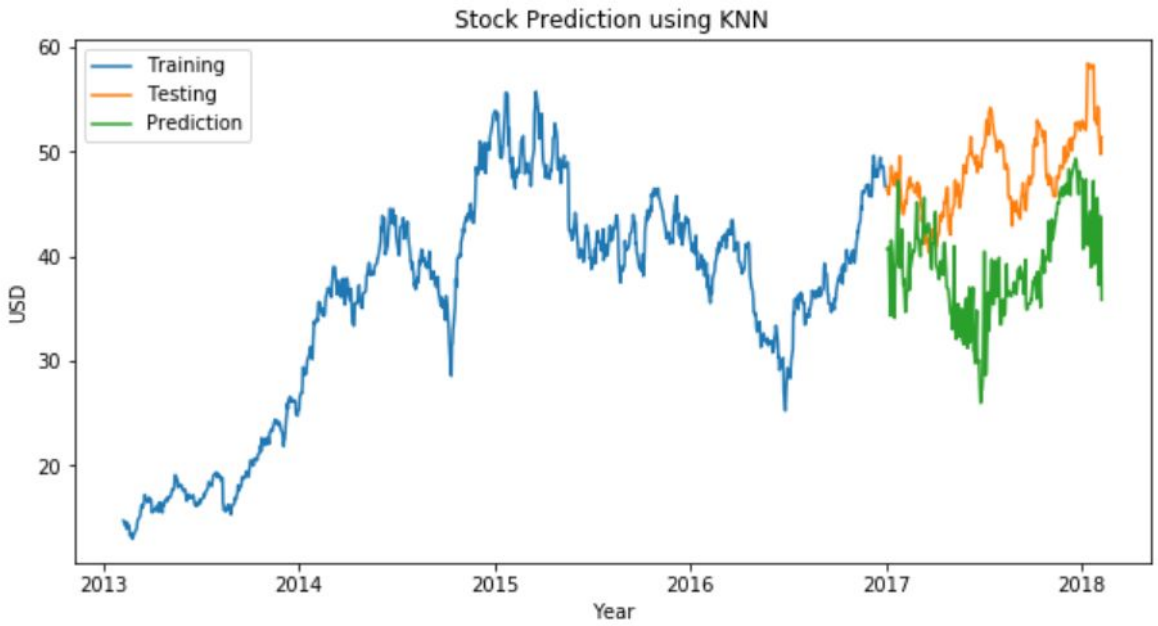

Figure 3: The above plot shows the performance of predicted prices against the actual values using K- Nearest Neighbor model on AAL stock's price over time.

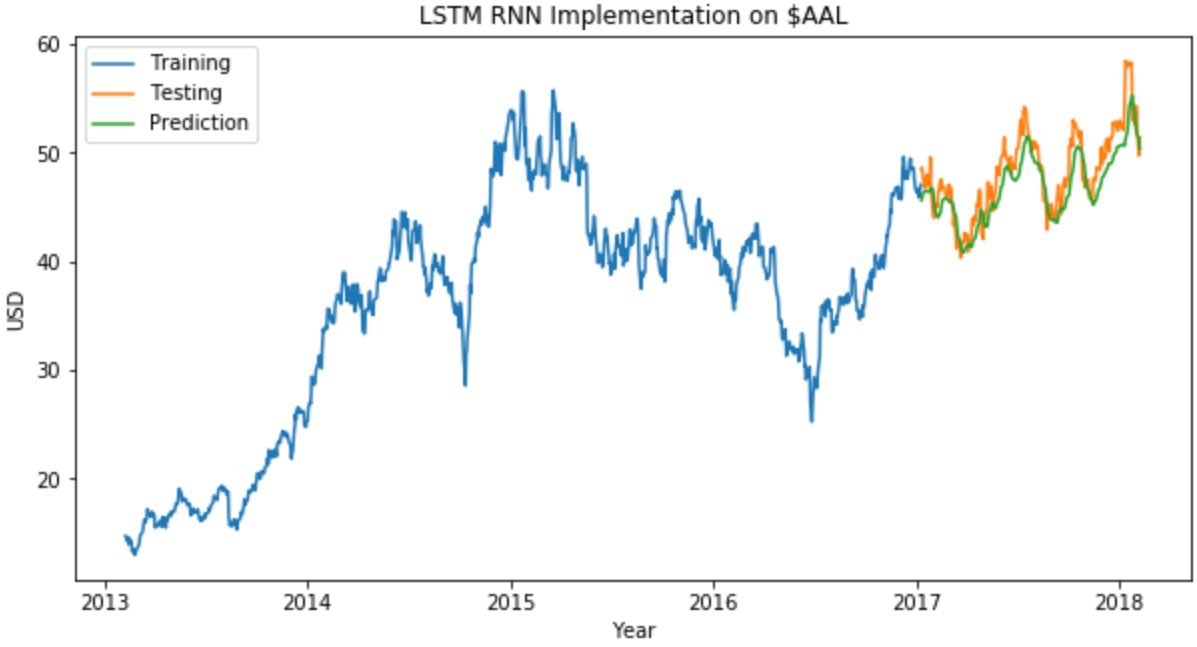

Figure 4: The above plot shows the results of predicted prices against the actual prices using LSTM RNN regression on AAL stock's price over time.

Model	Prediction Error rate
Moving Average	RMSE on test set= 10.14
Linear Regression	RMSE on test set= 6.69
K Nearest Neighbor	RMSE on test set= 10.98
LSTM RNN	RMSE on test set= 2.008

Table1: The above table shows the error rate of each model on AAL stock prediction

Findings

As expected, the machine learning models that take into account time series data perform much better than those models that simply generate a prediction. LSTM performs very well thanks to its consideration of previous patterns seen in the testing data, with root-mean-square error of 2.008. Some sort of feature transformation could be used to bolster the performance of the models that do not use time series data. Such a transformation would make it such that a prediction would not have temporal dependencies.

Summary and Future Work

In conclusion, we have explored various machine learning models like moving average, linear regression, KNN, LSTM, logistic regression, and support vector machines to predict the stock price of AAL stock based on the historical data of the stock over a period of 5 years. We achieved the best performance using the LSTM model.

As future work, we would like to predict if the user should hold on to a specific stock. We would like to explore reinforcement learning techniques for stock prediction and classification. It would also be interesting to explore how well our existing machine learning models perform on other stocks after trained. We also plan to explore a method such as gradient boosting. This is a process to turn weak learners into strong learners iteratively.

Team Contributions

Achsah Ledala worked on Moving Average, Linear Regression, KNN models and writing the report.

Ikechukwu Uchendu worked on the classification models and writing the report.

Riley Annis worked on the LSTM RNN model and writing the report.

Luke Stanton worked on reinforcement learning, XGBoost, and writing the report.

Bibliography

https://www.kaggle.com/camnugent/sandp500

https://towardsdatascience.com/machine-learning-techniques-applied-to-stock-price-prediction-6c1994da8001

https://github.com/llSourcell/Reinforcement_Learning_for_Stock_Prediction

https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/

https://cleartax.in/s/stock-market-analysis

https://www.investopedia.com/university/movingaverage/movingaverages1.asp

https://www.researchgate.net/publication/328930285_Stock_Market_Prediction_Using_Machine_Learning