Interim Project 2

Project Report

Kenny McAvoy

Student ID: [redacted]

ISYE 6767

Georgia Institute of Technology

November 23, 2020

Table of Contents

Chapter 1: **Project Specification**

The project is written in Python 3, using commonly utilized python packages, such as Pandas, Numpy, Quandl, and Scipy. A list of the required packages, and versions can be found in the README.txt file. The folder structure of the project, utilized to import data and output results can be found in the README.txt file, as well.

Chapter 2: **Introduction**

This project aims to use two machine learning models (Logistic Regression and K Nearest Neighbors) to predict whether a stock will rise in closing price the next day, using inter-day data. The project is written in Python 3, using commonly utilized python packages, such as Pandas, Numpy, Quandl, and Scipy. The models were trained using 60% of the data available, while testing on the remaining 40%.

Chapter 3: Data Considerations

A number of steps were taken to ensure the data used in this project correctly handled missing values, with no look-ahead bias. This was done by ensuring forward filling of data, making sure that only historical values would replace missing data. Likewise, the data was normalized, bringing all values into the range between 0 and 1. This practice helps to make predictions easier for ML models, as the data is all in the same scale. Normalization also ensures that the differences the data should highlight between each training entry are maintained. In order to counter and divisions by zero, etc., created during the technical indicator computation portion of the project, all infinity values were replaced with NaN values. The steps of the data cleaning and preparation are as follows:

1. Import of relevant data
2. Normalization of Data
3. Relevant Calculations
4. Replacement of infinity values
5. Removal of irrelevant data
6. Forward Filling
7. Removal of NaN values

Chapter 4: **Feature Selection**

There are a number of technical indicators commonly used to help analyze the movements of a stock's price. In this project the following indicators are used:

Simple Moving Averages (SMA):
Simple moving averages help to determine the existence of a trend within a stocks pricing. In order to both evaluate short-term and longer term trends, SMAs of 5, 10, and 100 days were utilized. The 5 and 10 day SMAs help to reflect shorter term trends in a stocks price, while the 100 day SMA can provide support to a long term trend.

Exponential Moving Averages (EWM):
Exponential moving averages perform in a similar manner to SMAs, however, exponetial moving averages are able to react more quickly to large changes in a stocks closing price. This is because a EWM places more weight on recent data over past data. EWMs of 5, 12, and 26 days were utilized in the project. These values were chosen as the MACD indicator commonly uses the 12 day and 26 day EWM, while the 5 day EWM will provide a good indicator of short-term market reactions in-conjunction with the 5 day SMA.

Moving Average Convergence Divergence Indicator (MACD):
The Moving Average Convergence Divergence indicator is a trend following indicator. The MACD commonly utilized the 12 day and 26 day EWM. The MACD is useful as an indicator of speed of convergence or divergence of a stocks price from its current position. Likewise, it can help a trader to understand when momentum trends may be shifting or the presence of a bearish or bullish market. The MACD is commonly used in conjunction with the RSI indicator.

Bollinger Bands (BB):
Bollinger Bands represent an envelope of where a stock price may move within two standard deviations of its historical volatility. In this implementation, the 21 day historical volatility was used to calculate the upper and lower bands. Bollinger Bands generally represent when a stock is overbought or oversold. When the price reaches wither the upper band or lower band it is likely it will revert back to a price within the envelope.

Stochastic Oscillator (K):
The Stochastic Oscillator indicator returns a signal between 0-100, representing whether a stock may be overbought or oversold. The K can help one to determine if a price reversal may be likely by using the current price in relationship to the past average high and low prices of the stock. In this implementation, the commonly used 14 K was used and the crossover sigal line for K is its 3 day moving average. A crossover of K and its signal line indicates a shift in momentum from the current price direction. K is commonly used in conjunction with RSI as they represent similar ideas, but based on different underlying frameworks for calculation.

Relative Strength Index (RSI):
The Relative Strength Index is an indicator that represents a similar trading indications as the stochastic oscillator. However, the RSI takes into account the speed of changes in price movements. The RSI is commonly calculated using a 14 day moving window. Using the average returns over this window, the RSI indicator can help a trader to identify when a stock is being overbought or sold, based on the returns and losses over the window. Generally, when the RSI falls below 20 or over 80, a reversal in price is likely.

Indices Data:

In conjunction with the technical indicators, the SP 500 and VIX normalized closing data were appended to the training entries. These two indices help represent market movements as a whole, where the SP 500 is commonly used as an index for market health and the VIX is a representation of trading confidence. These two metrics can help determine if traders are confident in market growth, or there is skepticism and impending reversion of prices.

Chapter 5: **Modeling**

In this project the two models used are the Logistic Regression (LG) and the K Nearest Neighbors Classifier (KNN). Both of these models are best implemented during a classification problem, as presented in this project. However, Logistic Regression and KNN differ significantly in the methods used to determine the classification of their outputs. The models were trained using 60% of the data available, while testing on the remaining 40%.

Logistic Regression follows a similar fundamental base as linear regression, however it fits a Sigmoid function instead. The Logistic Regression will attempt to fit a Sigmoid function to the data presented in the model, and then use that function to calculated the probability of either classification of 1 or 0 based on the function. Given a probability greater than 0.5 for classification 1, the logistic regression will return a classification value of 1 and likewise for 0 when the probability of classification as a 0 is less than 0.5. The logistic regression, however can only be applied to a linear classification problem and has a greater sensitivity to outliers.

The KNN Classifier uses the training data to create clusters of data. These clusters represent the classification of the data, which in this case, is either a 0 or 1. The KNN classifier will then attempt to match each tested data point to a respective cluster, based on the distance of the testing point from a number of training points. The number of points the KNN considers is one of it's key hyper-parameters that can be tuned. After calculating the distances from each of the testing points to each of the training points, over the sample space, the KNN Classifier uses a voting system based on the hyper-parameter. Given a hyper-parameter value of N, the KNN model will consider the closest N trained points to the testing point, and based on the number of classified 0 or 1 trained points closest to the testing point, the KNN model will classify the testing point as either a 1 or 0. Due to the importance of this hyper-parameter, each model was trained to find the best N hyper-parameter for the given data over a range from 1 to 30 neighbors. KNN has the disadvantage of being a very costly ML model, as the distance from the testing point to every trained point must be calculated to determine the neighborhood, however, it does not require a linear classification problem.

I expected each of these models to perform well, but given the linear and consecutive nature of the data, I expected the logistic regression to work faster, and perform slightly better, which is confirmed by the results.

For the large universe of stocks, the stocks are then ranked as to which stocks are best predicted by each model and overall. This ranking is by first the F1 score and then the AUC value. The rank of the stocks are presented in the outputted evaluation metrics as 'model_rank', and 'model_rank_overall', where 'model_rank' represent the ranking of the stocks within each respective model, and 'model_rank_overall' represent the best 20 stocks predicted overall.

Chapter 6: **Evaluation**

The key evaluation metrics considered in this project are, the F1 score, ROC curve, concordance statistic (AUC), Accuracy, Precision, Recall, and KS Statistic. Each of these metrics provide great detail into the quality of a classification model, however I placed greater weight on the F1 score and AUC, as these two metric encompass many of the qualities the other metrics show.

The F1 score is a weighted average of the precision and recall of a models predictions, with the best value being a 1. This score helps to determine if the model is correctly classifying the proper points as either a zero or one and not blindly picking all the points, which would create a skewed recall value and a poor precision. Likewise, accuracy may be skewed by the number of testing points that classify as either a 1 or 0. If there are very few points classified as 1 in the testing data, and the model classifies all the points as 0, the accuracy will be quite high, however the precision and recall will be 0.

The ROC curve and AUC represent the relationship of the true positive rate to the false positive rate, meaning how well a true prediction by the model represents a testing point that is actually true (classified as 1) and vice-versa. The AUC is an aggregate measure of the relationship presented by the ROC curve over all classification thresholds. In essence, it is the probability of the model to predict a true positive vs. a false positive, vs. a 50/50 guess.

The KS statistic is a statistical measure of how well each model classifies a given data point. The KS score determines if the model correctly differentiates all points as true positive or false negative. A higher KS score means the model differentiates the given data points quite well.

Chapter 7: **Summary and Conclusions**

This project aimed to predict whether the closing price on a given day of trading will be an increase over the previous day, i.e. a positive return. The project trained two machine learning models, Logistic Regression and K Nearest Neighbors, for each stock within the small universe and the large universe of stocks. For the large universe of stocks, the stocks are then ranked as to which stocks are best predicted by each model and overall.

Given the models tested, the logistic regression model performed better overall vs. the KNN model, which frankly also performed quite well. For future testing of this problem, I would analyze further the logistic regression given the computational time limitations of the KNN model over larger data sets, and the advantage/disadvantages given in the modeling chapter.

An improvement may be only using historical data, by performing rolling windowed regression or a rolling expanding window regression, where the training data would better simulate life-like trading circumstances, where the model can only be trained on data from previous days to predict the following day. To address possible over-fitting, the previously described method may help, but in the current implementation, data was ensured to have no look ahead bias and trained to ensure minimal out of sample error. Likewise, the number of features used are significantly smaller than the number of data point present, which will help prevent over-fitting.

Though the models can predict a rise in closing price the following period, I would not suggest building a trading strategy on them. The models do not predict how large the increasing in closing price may be or take into any account other possible trades one came make, or transaction costs. For example, on a given day the price may increase by $1 and the models correctly predict that, however the return of trading on another asset may be higher or the transaction cost of said trade is greater than $1. For this model to be used in a trading strategy, a confidence interval of how much the stock price will rise would be helpful, while also taking into account the possible portfolio of other assets than may be traded on, whilst accounting for transaction costs.