ISYE 6767

Design and Implementation of Systems to Support Computational Finance

Midterm Project 2

Data Analysis and Machine Learning

Omkar Kulkarni

GT ID: 903467583

1. Project Objective

The objective of the project is to predict stock price movement in financial markets by applying machine learning models. The focus of the project has been towards extracting and cleaning data, choosing appropriate factor variables that could explain the output from the target variable, choosing the model which gives us the best results, training and testing the model to calculate its accuracy, precision, recall score and other performance metrics

2. Data Analysis

2.1. Data Extraction

The project requires us to extract daily financial data for multiple stocks to feed them as factor variables or use them in calculation of other factor variables. Here we have used Quandl API to extract the data from the web. Quandl is a platform that provides financial, economic, and alternative data sourced from a large pool of publishers. All Quandl's data are accessible through an API. We can access this API by importing the "quandl" package defined for Python.

2.2. Data Filling

After extraction, the data had to be cleaned by eliminating or filling discrepancies or missing data. Data filling involves assigning values to missing data or data with another discrepancy. Filling can be classified into multiple type, e.g., forward filling, backward filling, etc. Forward filling involves assigning values from the previous to current cell. Since we cannot fill current missing value from the next day's data, I chose forward filling over backward filling.

2.3. Data Elimination

After filling missing data points, I removed blank data in the beginning of the dataframe, since we cannot backfill data from days ahead, and we cannot keep this data for the analysis.

2.4. Data Normalization

Normalization or data standardization involves standardizing all data to a specific scale. The goal of data normalization is to reduce and try to eliminate scale sensitivity, so that the model or algorithm can converge faster to reach its output. In this project, I have used the Standard Scaler from the scikitlearn module to standardize the data by removing the mean and scaling it to unit variance.

3. Features and effectiveness of the features in prediction:

3.1. Change in Volume

Volume has been one of the most popular indicators that is used to predict stock price movements. Understanding volume can provide insight into a stock's behaviour to help us understand. The most important rule is: volume precedes price. Typically, before a stock price moves, volume comes into play. The beauty of this indicator is its flexibility. Changes in volume can be used intra-day to determine short-term price movement or over several days to determine a stock's long-term trend direction. In general, a price change on relatively low volume for a particular stock suggests an aberration, whereas a price change on high volume portends a genuine trend reversal.

3.2. Simple Moving Average

A simple moving average (SMA) is an arithmetic moving average calculated by adding recent closing prices and then dividing that by the number of time periods in the calculation average. A simple, or arithmetic, moving average that is calculated by adding the closing price of the security for a number of time periods and then dividing this total by that same number of periods. Crossovers are one of the main moving average strategies. The first type is a price crossover, which is when the price crosses above or below a moving average to signal a potential change in trend. A moving average can also act as support or resistance. In an uptrend, a 50-day, 100-day or 200-day moving average may act as a support level. This is because the average acts like a floor (support), so the price bounces up off of it. In a downtrend, a moving average may act as resistance; like a ceiling, the price hits the level and then starts to drop again

3.3. Exponential Moving Average and MACD

An exponential moving average (EMA) is a type of moving average (MA) that places a greater weight and significance on the most recent data points. The exponential moving average is also referred to as the exponentially weighted moving average. An exponentially weighted moving average reacts more significantly to recent price changes than a simple moving average (SMA), which applies an equal weight to all observations in the period.

The 12- and 26-day exponential moving averages (EMAs) are often the most popularly quoted or analyzed short-term averages. The 12- and 26-day are used to create indicators like the moving average convergence divergence (MACD) and the percentage price oscillator (PPO). In general, the 50- and 200-day EMAs are used as signals of long-term trends. When a stock prices crosses its 200-day moving average, it is a technical indicator that a reversal has occurred.

3.4. Relative Strength Index (RSI)

The relative strength index (RSI) is a momentum indicator that measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset. The RSI is displayed as an oscillator (a line graph that moves between two extremes) and can have a reading from 0 to 100.

The standard is to use 14 periods to calculate the initial RSI value. Traditional interpretation and usage of the RSI is that values of 70 or above indicate that a security is becoming overbought or overvalued and may be primed for a trend reversal or corrective pullback in price. An RSI reading of 30 or below indicates

an oversold or undervalued condition.

3.5. Bollinger Bands

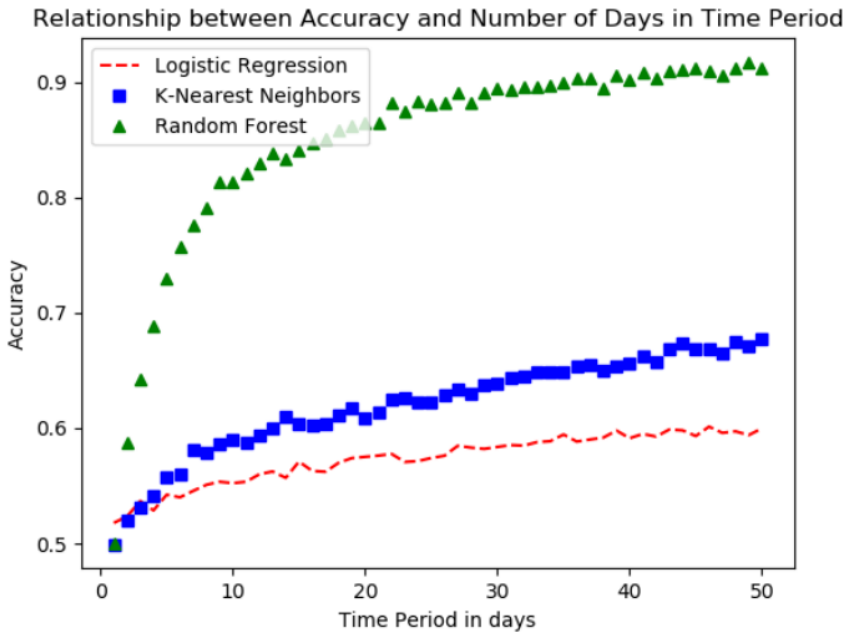
Bollinger Bands are a set of lines plotted two standard deviations (positively and negatively) away from a simple moving average of the security's price. Because standard deviation is a measure of volatility, when the markets become more volatile, the bands widen; during less volatile periods, the bands contract. Many traders believe the closer the prices move to the upper band, the more overbought the market, and the closer the prices move to the lower band, the more oversold the market.

---

4. Machine Learning Models

4.1. Model Selection

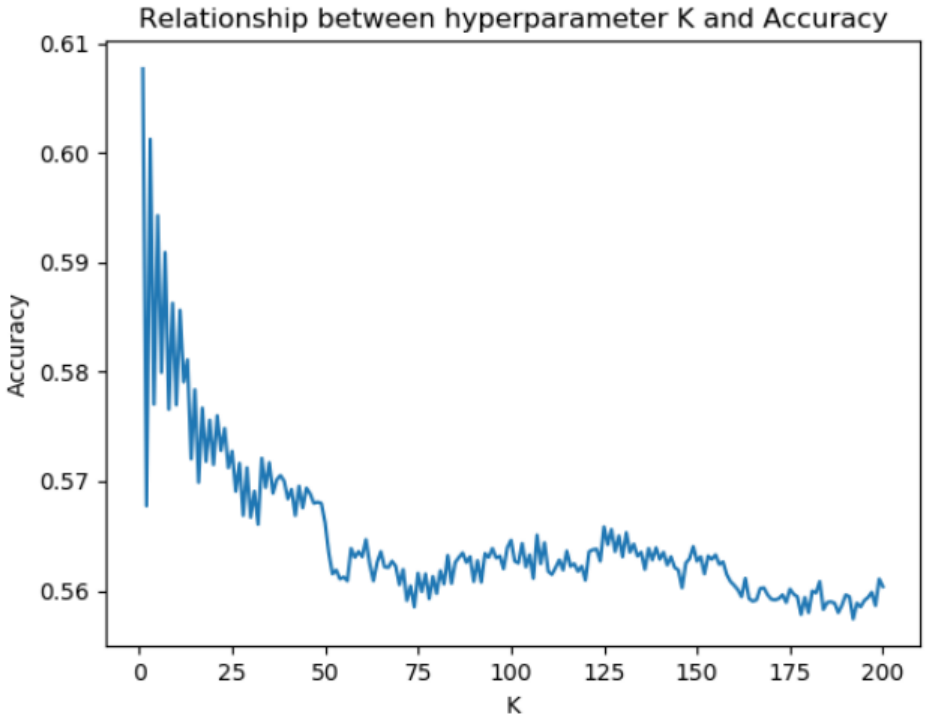
I initially trained and tested the data with multiple models – Logistic Regression, K-Nearest Neighbors, Random Forest over different time periods. We can see that logistic regression provides the least accuracy amongst all. Logistic regression is basically a supervised classification algorithm derived from linear regression. Logistic regression gives a value between zero and one and based on a threshold calculated by fitting the model based on the training data, it gives a categorical output. Since, the output is not exactly categorical, it performs poorly as compared to K-Nearest Neighbors and Random Forest algorithms which are based on clustering of data and decision trees.



From the above graph, we can see that the accuracy is around 50% when we predict stock price movement over a time period of 1 day, and increases with increase in time period. I have used time-period as 10 days to train and test the models.

4.2. K-Nearest Neighbours Algorithm

K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection. It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data. We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute. Now, given another set of data points (also called testing data), allocate these points a group by analyzing the training set. If we plot these points on a graph, we may be able to locate some clusters, or groups. Now, given an unclassified point, we can assign it to a group by observing what group its nearest neighbours belong to.

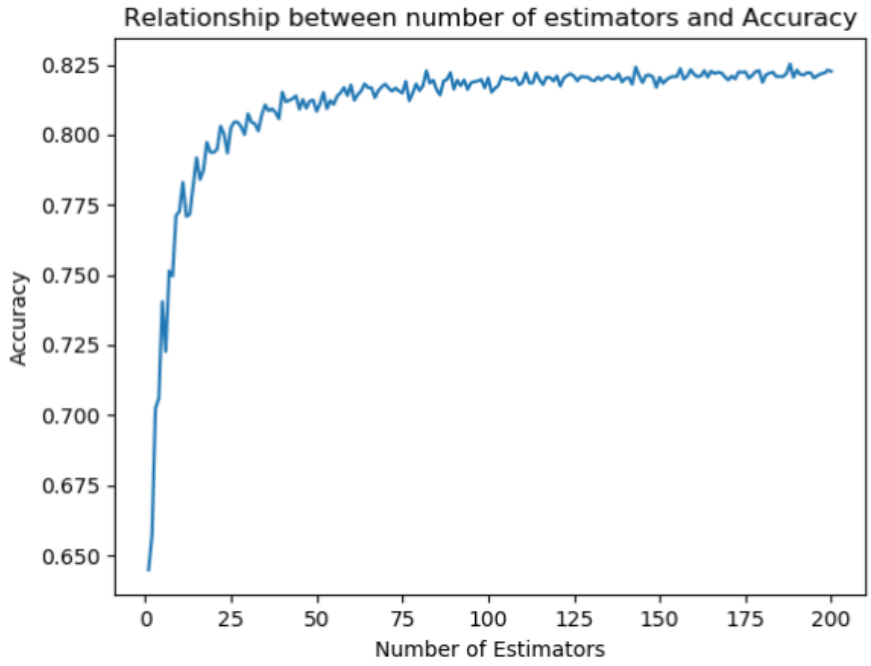


We can also observe a change in accuracy with change in ‘k’ hyperparameter in the model. Based on the above graph that I obtained, I have chosen to take k = 5 as the hyperparameter to train and fit my model.

4.3. Random Forest Algorithm

Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because it’s simplicity and the fact that it can be used for both classification and regression tasks. Random Forest is a supervised learning algorithm. Like you can already see from its name, it creates a forest and makes it somehow random. The forest it builds, is an ensemble of Decision Trees, most of the time trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models

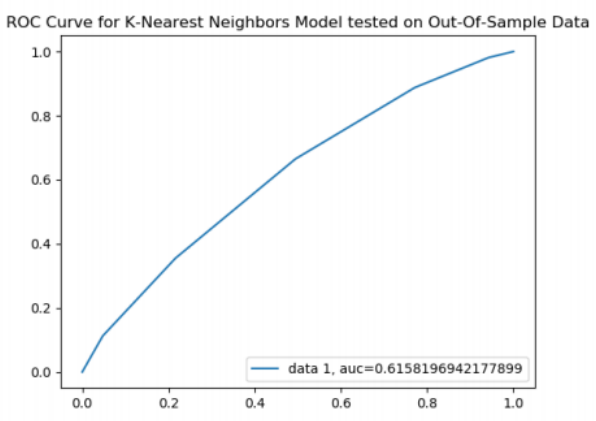
increases the overall result.

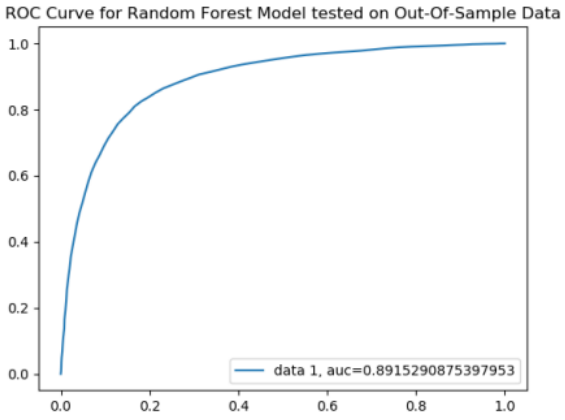


We can also observe a change in accuracy with change in number of estimators in the model. Based on the above graph that I obtained, I have chosen to take 75 as the number of estimators to train and fit my model.

5. Conclusion of the project:

Overall, we can see that the Random Forest model has a better accuracy, area under the ROC curve and other performance metrics.





For the large universe of stocks, the 20 best predicted stocks along with their accuracy and area under curve are:

Ticker	K- Nearest Neighbors		Random Forest	
	Accuracy	Area Under Curve (ROC)	Accuracy	Area Under Curve (ROC)
WBA	0.725664	0.809226355	0.922124	0.981980853
PBCT	0.677876	0.720333231	0.929204	0.97172276
DXC	0.789474	0.716666667	0.789474	0.325
OXY	0.60354	0.629415332	0.621239	0.649375916
AMTD	0.520354	0.499130435	0.640708	0.627602108
WIN	0.557522	0.525662252	0.59882	0.528357052
AWK	0.520354	0.545980544	0.624779	0.637594206
RGLD	0.534513	0.540254902	0.60177	0.617084967
BGFV	0.539823	0.542406349	0.59292	0.62192217
THG	0.555752	0.56051816	0.576991	0.594251063
CRUS	0.546903	0.527361395	0.584071	0.603887501
DIS	0.578761	0.600954175	0.552212	0.558010044
MVC	0.545133	0.50232313	0.578761	0.475258993
EIX	0.543363	0.558681672	0.569912	0.589601742
CLI	0.562832	0.581237292	0.548673	0.569087783
CCBG	0.564602	0.576033402	0.546903	0.576886149
SAM	0.538053	0.550575378	0.571681	0.589372498
TGH	0.539823	0.538418164	0.569912	0.579286019
MA	0.514184	0.512194754	0.592199	0.526380488
ANGI	0.543363	0.55704063	0.562832	0.571509136

As we can see, since we trained the model on a short set of data and tested on a huge dataset, we see a significant decrease in accuracy after the first 10 stocks.

Advantages and Disadvantages of K-Nearest Neighbors Algorithm:

- The K-Nearest Neighbors algorithm is very effective, provided the training data is large. It is also robust to noisy training data, particularly if we use inverse square of weighted distance as the distance. Also, learning can be done by local approximation using simple procedures.

- The right hyperparameter ‘k’ needs to be determined in order to get a good accuracy. As we increase the training data, the computation cost increases. The performance also depends on the attributes selected.

Advantages and Disadvantages of Random Forest Algorithm:

- The Random Forest algorithm is very accurate, and is generally one of the most accurate machine learning models. It runs effectively on large databases and handles thousands of input variables without deletion. It also gives estimates of what variables are important in the classification

- The main limitation of Random Forest is that a large number of trees can make the algorithm to slow and ineffective for real-time predictions. In general, these algorithms are fast to train, but quite slow to create predictions once they are trained. A more accurate prediction requires more trees, which results in a slower model.

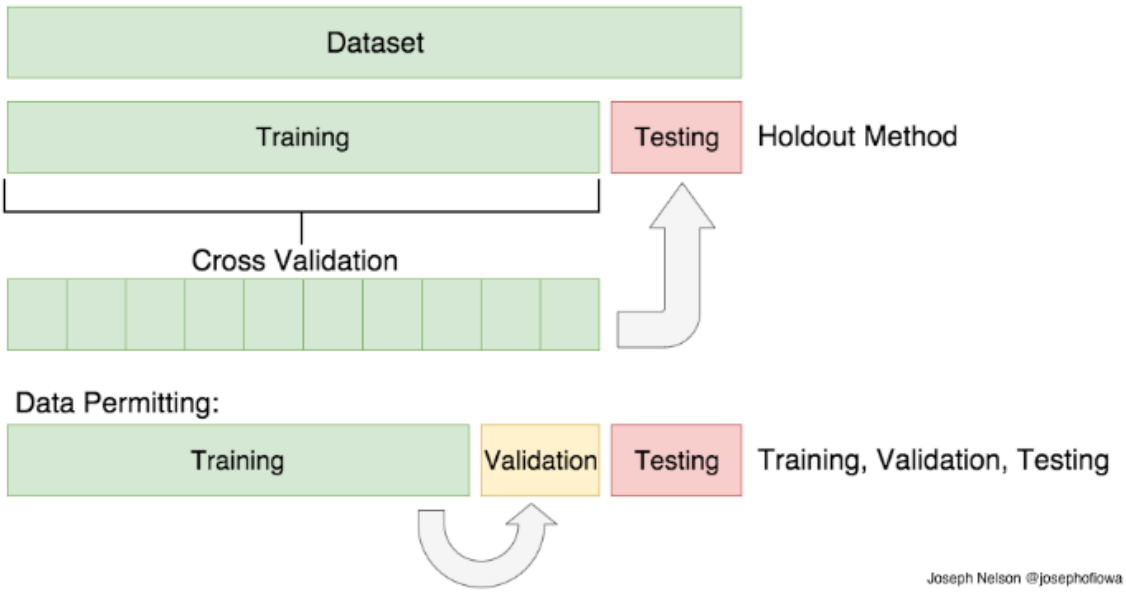
5. Additional considerations:

- The Random Forest Model has a high accuracy when you increase the time period for predicting stock price movements. However, in the short-term, the accuracy converges to 50% along with the other machine learning models. Hence, the Random Forest algorithm cannot be specifically used for intra-day trading based on the parameters I have defined. We can increase the number of factors fed to the model to improve its accuracy.

- A big problem in machine learning is over-fitting. We can experience over fitting when we have a huge training set and a small testing dataset. A 90%-10% or 80%-20% training-testing split results in an over-fitting model, which would not be able to perform well on the testing data. The 60%-40% split ensures we have a good fit model that explains both the training and testing data efficiently.

- **Overfitting definition:**
- Overfitting means that model we trained has trained “too well” and is now, well, fit too closely to the training dataset. This usually happens when the model is too complex (i.e. too many features/variables compared to the number of observations). This model will be very accurate on the training data but will probably be very not accurate on untrained or new data. It is because this model is not generalized (or not AS generalized), meaning you can generalize the results and can’t make any inferences on other data, which is, ultimately, what you are trying to do. Basically, when this happens, the model learns or describes the “noise” in the training data instead of the actual relationships between variables in the data. This noise, obviously, isn’t part in of any new dataset, and cannot be applied to it.

- In order to see if our models are overfitted or not we can also use **cross validation.** In cross validation we split our data into k subsets, and train on k-1 one of those subset. What we do is to hold the last subset for test. We're able to do it for each of the subsets.



- If after using the cross validation method we are getting the same accuracies for our model then our models are not over-fitted. If we get significantly lesser values then our model will be over-fitted
- Random Forest indicates a 100% accuracy for training data, which could be possibly assumed as overfitted. However, Random Forest does not overfit. The testing performance of Random Forest does not decrease (due to overfitting) as the number of trees increases.