

# MIDTERM PROJECT II

MACHINE LEARNING WITH PYTHON

# Ashutosh Agrawala

**TABLE OF CONTENTS**

[Introduction 02](#_TOC_250014)

[Extraction of Data 02](#_TOC_250013)

[Handling of Missing Data 02](#_TOC_250012)

[Normalize or Standardize Data?. 03](#_TOC_250011)

[Features used for predicting results 03](#_TOC_250010)

[Simple Moving Average (SMA) 03](#_TOC_250009)

[Exponential Moving Average (EMA) 03](#_TOC_250008)

[Moving Average Convergence Divergence (MACD) 04](#_TOC_250007)

[Implementation of Features 04](#_TOC_250006)

[Machine Learning Models Used 04](#_TOC_250005)

[K-nearest neighbor 04](#_TOC_250004)

[Random Forest Classifier 05](#_TOC_250003)

[Expectations from the models 05](#_TOC_250002)

[Overfitting of Data 06](#_TOC_250001)

Accuracy of Models (Real time applications) 07

[Conclusion 08](#_TOC_250000)

## INTRODUCTION

(Purpose of the project and the Problem addressed)

In this project, Machine Learning models have been implemented to predict daily stock returns. The training data used for this model corresponds to 60% of the daily returns of 10 stocks from 2000-01-01 to 2018-01-10. The output for the model is binary - if the stock return is expected to be positive, the result is 1 and if the return is negative, the result is 0. This model is then tested on a larger list of tickers for the same period. Finally, 20 stocks shortlisted which are best predicted by the model.

Three primary factors are used to determine the stock price pattern.

1. Simple Moving Average (window – 26 days, 10 days & 5 days)
2. Exponential Moving Average (window – 26 days, 10 days & 5 days)
3. Moving Average Convergence Divergence

Two Machine Learning models have been used to predict the results.

1. K-nearest neighbor
2. Random Forest Classifier

## EXTRACTION OF DATA

The NYSE and NASDAQ tickers were read from csv files. A subset of these tickers whose market cap was 500 million USD was taken. This provided us with 3091 tickers. The adjusted close prices for these tickers were then sourced in from Quandl using the Quandl library in Python. Data was not available for all tickers, however around 2000 tickers’ adjusted price data was collected.

### HANDLING OF MISSING DATA

There were some data-points that were missing in the sourced data. In this scenario, the data was populated using forward fill first and then backward fill. If the stock price for the next day is unknown, we assume today’s price to carry forward. The backward fill was performed in order to avoid a scenario of first day missing data.

### NORMALIZE OR STANDARDIZE DATA?

In the business world, "normalization" typically means that the range of values are "normalized to be from 0.0 to 1.0". "Standardization" typically means that the range of values are "standardized" to measure how many standard deviations the value is from its mean.

In this project, the data was normalized as the prices in the range of 0 to 1 makes it easier to evaluate its relative value. Thus, each column was then normalized and values between 0 to 1 were obtained instead of absolute prices of stock.

## FEATURES USED FOR PREDICTING RESULTS

Three features were used primarily in the project and used as the X variables in the ML models –

### SIMPLE MOVING AVERAGE (SMA)

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. The simplest form of using a simple moving average in analysis is using it to quickly identify if a security is in an uptrend or downtrend. Many traders watch for short-term averages to cross above longer-term averages to signal the beginning of an uptrend.

Three windows were used in calculating the SMA – 26 days, 10 days and 5 days. The reason for taking 3 windows is to capture both short term and long-term average.

### EXPONENTIAL MOVING AVERAGE (EMA)

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.

Like all moving average indicators, they are much better suited for trending markets. When the market is in a strong and sustained uptrend, the EMA indicator line will also show an uptrend and vice-versa for a down trend. A vigilant trader will not only pay attention to the direction of the EMA line but also the relation of the rate of change from one bar to the next.

Similar to SMA, three windows were used to compute the EMA – 26 days, 10 days and 5 days. This gives the model a better insight into both long term and short-term price movements.

### MOVING AVERAGE CONVERGENCE DIVERGENCE (MACD)

The Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security’s price. The MACD is calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA. The result of that calculation is the MACD line. A nine-day EMA of the MACD, called the "signal line", is then plotted on top of the MACD line which can function as a trigger for buy and sell signals. Traders may buy the security when the MACD crosses above its signal line and sell, or short, the security when the MACD crosses below the signal line. The MACD helps investors understand whether bullish or bearish movement in the price is strengthening or weakening.

### IMPLEMENTATION OF FEATURES

Three functions have been defined to compute the three features. The SMA and EMA functions take in the prices and the required window as the input whereas the MACD function only takes the prices array as input. The SMA, EMA and MACD values wrt each ticker and each date are computed by calling the defined functions which are then passed in as X-variables to the ML model.

## MACHINE LEARNING MODELS USED

Two Machine Learning models were developed – K nearest neighbor and Random Forest Classifier. The training data for the models was 60% of the 10 stocks as provided in the csv file. These models were then tested on the remaining 40% of the prices.

Thereafter, the accuracy and precision of the models was computed on a broader dataset of tickers.

### EAREST NEIGHBOR

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.

The input consists of the k closest training examples in the feature space. The output is the property value for the object. This value is the average of the values of its k nearest neighbors.

ADVANTAGES:

* + - Robust to noisy training data
    - Effective if the training data is large DISADVANTAGES:
    - Difficult to decide the most appropriate value of K (number of nearest neighbors)
    - Computational cost is usually quite high as distance of each query instance to all training samples is computed

### RANDOM FOREST CLASSIFIER

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests usually avoid the decision trees’ habit of overfitting the training set. Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

ADVANTAGES:

* + - Easy to interpret and make for straightforward visualization
    - Performs well on large datasets and is extremely fast
    - Can efficiently handle a large number of input variables DISADVANTAGES:
    - Usually prone to overfitting of data with noisy classification/regression tasks

### EXPECTATIONS FROM THE MODELS

KNN would constitute for a good choice of the model as the short-term trends in stock prices are usually consistent. If the stock price is in an upward trend today, there is a high chance it would grow further in the coming days. In the project, K value is considered to be 5 as a 5-day pattern in price movement is assumed.

The stock price movements are usually very unpredictable, thus considering a model such as the Random Forest Classifier makes the prediction modelling efficient. Random Forest Model adds additional randomness in predicting results which makes it an apt choice in this case.

## OVERFITTING OF DATA

The accuracy results for the training and testing datasets are as follows: Out of sample (Testing data) - KNN

Accuracy value – 0.5012 Precision value – 0.4767 AUC value – 0.5002

In sample (Training data) - KNN Accuracy value – 0.7530 Precision value – 0.7501

AUC value – 0.7523

Out of sample (Testing data) – Random Forest Accuracy value – 0.5110

Precision value – 0.4852 AUC value – 0.5074

In sample (Training data) – Random Forest Accuracy value – 1

Precision value – 1 AUC value – 1

As per the above results, the out of sample accuracy is greater than 50%. In such a scenario, this model is not considered to be overfitted. Additionally, the in-sample results of Random Forest are to note here. The results are a 100% match this indicates the overfitting nature of Random Forest when compared to other similar models.

In order to mitigate overfitting, an optimum training and testing data ratio is required. When the same results were computed with 90:10 ratio, the model was quite overfitted. However, per multiple iterations it was observed that 60:40 ratio is quite optimum in this project.

## ACCURACY OF MODELS

#### (Real time applications)

The best 20 stocks predicted by the model on an average have 53% & 53.86% accuracy by the KNN and Random Forest models respectively. The medians are 52.93% and 53.70% respectively. The best stocks are as follows:

#### KNN Random Forest

|  |  |
| --- | --- |
| **Ticker** | **Accuracy Value** |
| BHF | 0.560606061 |
| ESPR | 0.534246575 |
| FRC | 0.533738938 |
| OAS | 0.533437014 |
| VCRA | 0.533108108 |
| NGHC | 0.532868526 |
| OGS | 0.53164557 |
| HPE | 0.530434783 |
| FIVE | 0.530292231 |
| UBNT | 0.529817954 |
| POR | 0.528939445 |
| KAR | 0.52892562 |
| BNFT | 0.528828829 |
| TSC | 0.528726062 |
| DYN | 0.528539659 |
| ARR | 0.528037383 |
| XON | 0.526362039 |
| PSTB | 0.526002972 |
| MTSC | 0.525669153 |
| CTRE | 0.525641026 |

|  |  |
| --- | --- |
| **Ticker** | **Accuracy Value** |
| UIHC | 0.553744664 |
| CONE | 0.545027408 |
| HPE | 0.544347826 |
| PFBC | 0.544098289 |
| DWDP | 0.543859649 |
| DOOR | 0.541531323 |
| ANDV | 0.540740741 |
| PSTB | 0.5397474 |
| PE | 0.538297872 |
| VRTS | 0.537254902 |
| ARI | 0.536931818 |
| HII | 0.536557283 |
| ARR | 0.535825545 |
| DYN | 0.535211268 |
| CBF | 0.534792807 |
| ANGI | 0.534098152 |
| ESRT | 0.534059946 |
| FIBK | 0.533199195 |
| BKU | 0.532430908 |
| TPRE | 0.532215357 |

Even though the model is relatively efficient as it has greater than 50% accuracy, it is not the best model to follow when investing in the real market. The accuracy here only indicates the stock price going up or down. It does not venture into the amount of loss suffered or profit made by the investor. With an accuracy of close to 50%, there is a high chance that the model might yield heavy losses and might not be the best suited for an average investor in the market. The model can be improved by adding more features/factors which can help in tracking the market better. We can also use this model more effectively if the time period of monitoring is not daily but for a longer

duration. Additionally, we should note here that the market is usually quite unpredictable, so it is difficult to develop a completely accurate model.

## CONCLUSION

Random Forest, on average, has provided better results than the K-nearest neighbor model in this project. This could be because Random Forest usually has a tendency to perform better on large datasets. Also, it operates primarily on randomness of data which is in-line with the market performance. However, KNN relies on short term patterns which need not always be true. As expected, the model yielded really high accuracy values when tested on the stocks used for training it. However, upon removing them on average the models have shown an accuracy of around 50% on the remaining stocks.