

Stock price prediction using feature engineering and machine learning
techniques

by

Aditya Vijay Narkar

B.E., University of Mumbai, 2016

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2019

Approved by:

Major Professor
Lior Shamir

Copyright

© Aditya Vijay Narkar 2019.

Abstract

The correct prediction of stock prices is a challenging task, as stock prices are affected by a large number of parameters. Moreover, many of these parameters, such as investor sentiment or future market potential, cannot be measured and quantified directly, while having a substantial impact on individual stocks and the stock market as a whole. In this project, I analyzed the changes in the stock price to predict the stock's direction in the future. That is done by extracting multiple descriptors from past data and using them to predict the price change of the stock up to 100 days in the future. Experimental results are collected using 10 stocks and Random Forest, SVM, and KNN classifiers and compared against a baseline ZeroR prediction. The project's goal is to assist the stock traders by providing data-driven insights about the predicted time and direction of changes in the stock price.

Table of Contents

List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Related work	2
2 Data Collection	3
2.1 Adjusted Data	4
3 Data Preparation	5
3.1 Features	5
3.1.1 Moving Averages	5
3.1.2 Oscillators	7
3.1.3 Differences	9
3.1.4 Fourier Transform	10
3.1.5 Central Moments and Entropy	11
3.2 Class label creation	12
3.3 Trend Deterministic Values	12
3.3.1 Moving Averages	12
3.3.2 Oscillators, MACD and Momentum	13
4 Algorithms	14
4.1 Random forest	14
4.2 K-Nearest Neighbors	15

4.3	Support Vector Machine (SVM)	15
4.4	ZeroR	15
5	Training, Testing, and Model Selection	16
5.1	Train-Test-Split using shuffle	16
5.2	K-fold cross validation	17
5.3	Model Selection	17
6	Feature Selection	20
6.1	Definition.	20
6.2	Algorithms for Feature Selection	21
6.2.1	Recursive Feature Elimination	21
6.2.2	Mutual information	21
7	Experiment	22
8	Results	25
8.1	Discretized features only	25
8.1.1	Testing with train-test-shuffle	25
8.1.2	Testing with FD-CV	27
8.1.3	Testing with TS-CV	27
8.2	Feature selection	29
8.2.1	Testing with train-test-shuffle	29
8.2.2	Testing with FD-CV	29
8.2.3	Testing with TS-CV	29
8.3	Analysis	33
9	Prediction	34
10	Conclusion	37

11 Future Work	38
Bibliography	39

List of Figures

5.1	Cross validation train and test sets	18
7.1	Experiment Flow	22
7.2	Experiment details	23
8.1	Accuracy graph of training algorithms using <code>train_test_split_shuffle</code> with discretized features only	26
8.2	Accuracy graph of training algorithms using FD-CV with discretized features only	27
8.3	Accuracy graph of training algorithms using TS-CV with discretized features only	28
8.4	Accuracy graph of training algorithms using <code>train_test_split</code> with shuffle with Feature Selection	30
8.5	Accuracy graph of training algorithms using FD-CV with Feature Selection	31
8.6	Accuracy graph of training algorithms using TS-CV with Feature Selection	32
9.1	Prediction aggregator	35

List of Tables

5.1	Hyperparameters	19
9.1	Prediction results with different test methods	35
9.2	Prediction results with different algorithms	36

Chapter 1

Introduction

To have a financial stability and secure their financial future, people tend to save in different forms. They often manage their monthly, quarterly, or yearly budget so that they can reduce the overall spending and save money for the future use. Individuals and households often manage their finances such that they allocate some of their income for investment, where their money can be used to generate more money in the future. The stock market is one of the most common platforms which people use to invest their funds.

In a stock market, people can buy the shares of a stock of a corporation. The stock (also capital stock) of a corporation is all of the shares into which ownership of the corporation is divided.^[10] A person who owns a percentage of the share has the ownership of the corporation proportional to his share. The shares form stock. The stock of a corporation is partitioned into shares, the total of which are stated at the time of business formation.^[26]

To make money in the stock market, the simple principle is to "Buy low and sell high". That means, when you buy a share of a stock at price X and sell the same share at price Y , you will make profit if $Y > X$. From now on, in this report, stock and shares of a stock is mentioned interchangeably.

The complexity arises while making the decision when to buy a stock and when to sell a stock. The stock price fluctuates because of many reasons. That can be supply or demand

of a stock, news related to the corporation of a stock, news related to the industry of a corporation of a stock, corporation's earning report, etc. All these reasons make hard for a stock trader to correctly predict the future behavior of a particular stock.

Motivation. Substantial amount of research has been invested in the area of stock price prediction. Many of the research efforts were done by using the hand picked stocks from the stock market. That, however, might lead to some overoptimistic results due to bias in the selection of the stocks. Even after getting the positive results from the experiments, the results do not provide information of when to sell the stock and capitalize on the profit. In this report, I am building an application to help a stock trader to specify the target and the stop-loss range in percentage and create labels for the data based on the specified ranges. I am also using randomly selected stocks for this task. Finally, I will be filtering out the stocks of which their price is the most predictable, and use them for the prediction task. I will be using the predictions of various stocks to check how they fair in reality.

1.1 Related work

Patel, Shah, Thakkar, and Kotecha^[18] did a comparative study of Artificial Neural Network (ANN), Support Vector Machine (SVM), Random forest, and Naive-Bayes algorithms in the field of stock price prediction. The study concluded that the Trend Deterministic Values (TDVs) are better than the continuous values for predicting stock's future direction(up or down). Although, the study was detailed and helpful to know that TDVs are better, in my opinion, they failed to show how the data are being split between testing and training data.^[19] showed the technique of fusing two models together for prediction purposes in the domain of stock price prediction. The technique mentioned in that paper can be used as a heuristic in this study.

Chapter 2

Data Collection

This experiment was performed on the NYSE stock data. To collect the data with as less human bias as possible, first, the list of all symbols has been collected from [8](#). Symbol is a short for of a full stock name. From the collected list of symbols, 10 symbols were chosen at random. For these 10 symbols, the daily adjusted complete data were collected from (<https://www.alphavantage.co/>). Data for following 10 symbols were collected till September 10, 2019:

1. LAZ (Lazard Ltd)
2. CUZ (Cousins Properties Inc)
3. ABR (ARBOR RLTY TR I/SH)
4. BHP (BHP Group Ltd)
5. KMT (Kennametal Inc.)
6. APH(Amphenol Corporation)
7. CWEN(Clearway Energy Inc Class C)
8. IX(ORIX Corporation)

9. INXN(InterXion Holding NV)

10. HUN(Huntsman Corporation)

2.1 Adjusted Data

Adjusted closing price amends a stock's closing price to accurately reflect that stock's value after accounting for any corporate actions. It is considered to be the true price of that stock and is often used when examining historical returns or performing a detailed analysis of historical returns. Some times some corporate actions, such as stock splits, dividends / distributions and rights offerings, affect a stock's price and adjustments are needed to arrive at a technically accurate reflection of the true value of that stock.^[7]

For prediction, if closing prices are not adjusted and used for feature engineering, then there is a risk of model learning incorrect insights. For this purpose, adjusted prices are collected.

In later sections 'closing price' or 'closed price' is used while talking about adjusted closing price of the stock.

Chapter 3

Data Prepartion

3.1 Features

Features are created to provide an algorithm with extra information or insights about the data. Features, especially in stock data, can help the algorithm learn about underlying extra information about stock's trend.

In this experiment, my idea is to design as many features as possible to get better insight of stock's trend. These features later help us to determine trend of the stock.

While designing these features, I take into account that the features use past data so that the atemporal algorithms like Random forest, can use the temporal information of stocks.

3.1.1 Moving Averages

Moving averages are used to smooth out the price actions of stock. This helps reduce the noise present in the stock's historical data. That way moving averages can focus on the overall trend of the stock's historical data.

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.^[12] It is a function which takes N observations and window size S as input and returns the mean for each window. Starting from first observation, first window will contain 0 to S observations, second window will contain 1 to S+1 observations, and so on. Therefore, there will be N-S+1 possible windows if N is equal to or greater than S.

$$SMA(t) = \frac{C_1 + C_2 + \dots + C_{S-1} + C_S}{S} \quad 14$$

Weighted Moving Average

Weighted Moving Average(WMA) is similar to SMA with a little addition of weights. WMA assigns higher weight to the recent data as compared to the older data. Weighted moving average is calculated as follows:

$$WMA(t) = \frac{1 \times C_1 + 2 \times C_2 + \dots + S-1 \times C_{S-1} + S \times C_S}{S} \quad 18$$

Where S = 50.

To calculate the multiplying factor for each term we can use the following formula: $\frac{i}{S}$, where i is the position in the window.

Exponential Moving Average

Similar to weighted moving average, exponential moving average also puts more emphasis on the recent data points. 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.^[11]

Exponential moving average is calculated as follows:

$$EMA(t) = C_t \times k + EMA(t-1) \times (1 - K)$$

3.1.2 Oscillators

Momentum

The momentum indicator compares the current price with the price in the past to get the relation of current price with the past price. The momentum indicator is simply a difference of the current price and a past price.^[2]

$$\text{Momentum}(t) = C_t - C_{t-k}$$

Where, k is number of days to look in past.

Stochastic K

The Slow Stochastic Oscillator is a momentum indicator that shows the location of the close relative to the high-low range over a set number of periods.^[4] Stochastic oscillator generates overbought signal at value 80 and oversold signal at value 20.

$$\%K = \frac{C_t - LL_{t-14}}{HH_{t-14} - LL_{t-14}} \times 100$$

Stochastic D

Stochastic %D is a 3-Day moving average of Stochastic %K.^[4]

$$\%D = \frac{\%K_t + \%K_{t-1} + \%K_{t-2}}{3}$$

Moving Average Convergence Divergence - MACD

MACD is one of the most often used indicators by the different stock trending simulator software. 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.^[15]

$$\text{MACD} = \text{EMA}(12) - \text{EMA}(26)$$

Relative Strength Index

Relative strength index measures the speed and change of price movements. RSI ranges from 0 to 100. RSI predicts overbought condition when RSI value is greater than 70 and predicts oversold condition when RSI value is lesser than 30. [3]

$$RSI(t) = 100 - \frac{100}{1 + \frac{Avg-Gain}{Avg-Loss}}$$

Where,

Average gain and average loss is calculated for last 14 days.

$$Gain - Loss(t) = C_{t-1} - C_t$$

$$Gain(t) = \sum_{i=0}^{13} Gain - Loss(t - i), \text{ where } Gain-Loss(t-i) > 0$$

$$Loss(t) = \sum_{i=0}^{13} Gain - Loss(t - i), \text{ where } Gain-Loss(t-i) < 0$$

$$Avg - Gain = \frac{Gain(t-13)+Gain(t-12)+\dots+Gain(t-1)+Gain(t)}{14}$$

$$Avg - Loss = \frac{Loss(t-13)+Loss(t-12)+\dots+Loss(t-1)+Loss(t)}{14}$$

Williams %R

Williams %R, developed by Larry Williams, is a momentum indicator that shows the relation between current closing price and high and low of the past N days. [5] This indicator gives us overbought and oversold ranges. Readings from 0 to -20 are considered overbought. Readings from -80 to -100 are considered oversold.

$$WilliamsR(t) = \frac{HH_{t-n} - C_t}{HH_{t-n} - LL_{t-n}} \times 100$$

Accumulation/Distribution (A/D) Oscillator

A/D oscillator follows the trend of a stock. It is an indicator which calculates money flow based on the historical data to help determine the stock's trend.

$$MoneyFlow(t) = \frac{(C_t - L_t) - (H_t - C_t)}{H_t - L_t}$$

$$AD(t) = MoneyFlow(t - 1) + MoneyFlow(t) \quad 6$$

Commodity Channel Index (CCI)

CCI is a momentum based oscillator which helps to determine overbought and oversold conditions.^[17] CCI is unbounded indicator and therefore there is a flexibility while deciding overbought and oversold regions. For this experiment, values such as 200 or above indicate overbought condition whereas values such as -200 or below indicate oversold condition.

$$CCI(t) = \frac{Typicalprice - MA}{0.15 * MeanDeviation}$$

Where,

$$Typicalprice = \sum_{i=1}^P \frac{H_t + L_t + C_t}{3}$$

P = Number of periods (For this experiment, P = 20)

$$MA = (\sum_{i=1}^P TypicalPrice) \div P$$

$$Mean\ Deviation = (\sum_{i=1}^P |TypicalPrice - MA|) \div P$$

3.1.3 Differences

%Difference from n days

In this feature, a difference of the current closing price from the closing price n days before has been calculated. This feature can provide the information about the movement of the stock from its n-previous closing prices.

$$\%difference(t) = \frac{C_t - C_{t-n}}{C_t}$$

Where, n = 90 in our experiment.

%Difference from lowest low

This feature calculates the price deviation of the current closing price from the lowest low price from the past. In this way, we can get the idea if the closing price is hovering near the lowest low or it is far away from it.

$$\%difference-LL(t) = \frac{C_t - LL_{t-n}}{C_t}$$

Where, n = 90 in our experiment.

Difference from highest high

This feature calculates the price deviation of the current closing price from the highest high price from the past. In this way, we can get the idea if the closing price is hovering near the highest high or it is far away from it.

$$\% \text{difference-HH}(t) = \frac{C_t - HH_{t-n}}{C_t}$$

Where, $n = 90$ in our experiment.

3.1.4 Fourier Transform

The Fourier transform (FT) decomposes a function of time (a signal) into its constituent frequencies.^[21] Our idea behind using Fourier transform is to find the distribution of the closing prices over some period of time in past.

FT-Min

FT-Min calculates the Fourier transform of a closing price for past n days and then find the minimum frequency from it. The idea behind using this feature is that it can tell us more about where the closing price was traded for a less amount of time.

FT-Max

FT-Min calculates the Fourier transform of a closing price for past n days and then find the maximum frequency from it. The idea behind using this feature is that it can tell us more about where the closing price was traded for a most amount of time.

FT-Mean

FT-Min calculates the Fourier transform of a closing price for past n days and then find the maximum frequency from it. The idea behind using this feature is that it can tell us more about the average closing price traded.

3.1.5 Central Moments and Entropy

Skewness

Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean.^[25] Negative skew tells us that the data is concentrated on the higher side and there are few lower values. Positive skew tells us that the data is concentrated on the lower side and there are few higher values. This feature can help our algorithms determine if the data is on the higher side or towards the lower side of the distribution.

Kurtosis

Kurtosis is a measure of the "tailedness" of the probability distribution of a real-valued random variable. Similar to skewness, kurtosis can help us determine the underlying trend in a data. Higher kurtosis is the result of extreme but infrequent deviations whereas lower kurtosis is the result of frequent modestly sized deviations.^[23] This feature can help our algorithms determine if the past data showed any signs of extreme deviations. These deviations can be caused by various sources but our algorithms can learn if the particular stock is having such behavior consistently or not.

Standard deviation(SD)

Even though standard deviation is not a central moment, it is the positive square root of the second central moment, i.e. Variance. Standard deviation is a measure of the amount of variation or dispersion of a set of values (Bland and Altman^[13]). A low standard deviation indicates that the values tend to be close to the mean of the set, while a high standard deviation indicates that the values are spread out over a wider range.^[24]

3.2 Class label creation

Class labels define the class in which the observation belong. In this experiment, the classes were created to help the stock trader set its P% profit target and S% stop-loss. By using target and stop loss, following three class labels were created:

- +1: This class label states that in X future days, the stock price is going to reach P% profit target before it reaches S% stop-loss.
- -1: This class label states that in X future days, the stock price is going to reach S% stop-loss before it reaches P% profit target.
- 0: This class label states that in X future days, the stock will neither reach P% profit target nor it will reach S% stop-loss.

3.3 Trend Deterministic Values

Some of the technical indicators have an involved meaning in their raw values. These technical indicators are called trend deterministic indicators in this experiment. These indicators provide information to understand the current trend of the stock. For such technical indicators,^[18] found out that the performance of the prediction models can improve if we use trend deterministic values(TDVs). Trend deterministic are binary values, +1 and -1, that indicate stock is likely to go up or down respectively in near future. Following is the list of technical indicators with the explanation of how to convert their raw values to TDVs.

3.3.1 Moving Averages

Moving averages summarizes the trend of a stock for past 'n' days. Different moving averages gives different weights to past data. But all the moving averages predicts the trend of a stock in a similar fashion. All of the moving averages use closing price of the stock and the value

of moving average to identify the trend of stock. If stock's closing price is trading above the moving average, then the stock is in uptrend(+1). Otherwise, the stock is trading in downtrend(-1) as per the insights given by moving averages.

3.3.2 Oscillators, MACD and Momentum

Stochastic %K, Stochastic %D, RSI, A/D oscillator, CCI, and Williams %R are stochastic oscillators which are trend indicators of a stock. That means, if the oscillators are increasing, then there is a higher chance of stock's value going up. Stochastic %K, Stochastic %D, and RSI indicate 70 or above as overbought condition, the stock's price is likely to go down in near future (-1), and 30 or below as oversold condition, the stock's price is likely to go down in near future(+1). If the raw value is trading in between 30 and 70, then if raw value at time t is greater than time $t-1$. then +1, else -1.

A/D oscillator states that if the raw value at time t is greater than time $t-1$. then +1, else -1. CCI is unbounded technical indicator. For this experiment, values equal or above 200 are address as overbought condition (-1) and equal or below -200 are oversold condition(+1). For the raw values lingering in between 200 and -200, if the raw value at time t is greater than time $t-1$. then +1, else -1.

Williams %R indicate overbought condition at value -20 (-1) or above and oversold condition at -80 or below (+1). For the raw values lingering in between -80 and -20, if the raw value at time t is greater than time $t-1$. then +1, else -1.

MACD and Momentum also follow the stock trend and can be discretized similar to the A/D oscillator. That is if the raw value at time t is greater than time $t-1$. then +1, else -1.

Chapter 4

Algorithms

Three algorithms were used for the task of stock price prediction: ZeroR or “Dummy” classifier, Random Forest, and KNN. These algorithms are classification algorithms. Also, these are atemporal algorithms. Therefore, the time sensitive information of the data does not matter to these algorithms. Following sections will provide an overview over how these algorithms work.

4.1 Random forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks.^[16]

Random forests constructs multiple decision trees at training time and outputs the class that is the mode of the class for the classification task.

Based on previous search done in the field of Stock price prediction, Random forest algorithm performed really well.^[18]

4.2 K-Nearest Neighbors

In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors.^[22] KNN uses distance metrics to calculate the distance of the unseen data point from the known data points. Based on the user defined k, k nearest neighbors are selected and the majority class is defined as the final class of the unseen data point.

4.3 Support Vector Machine (SVM)

Support-vector machines are supervised learning models used for classification and regression analysis.^[27] SVM creates a hyperplane or a set of hyperplanes to distinguish the data points which belong to different classes. SVM can be used for classification, regression, or other tasks like outliers detection.^[9] In this experiment, SVM is used for classification.

4.4 ZeroR

ZeroR or “Dummy” Classifier uses a naive strategy to predict the value for any given input. ZeroR algorithm with the ‘most frequent’ strategy to classify the given input was used in this experiment. This dummy classifier selects the most frequent label from the given training data and uses it for the prediction. This algorithm was used as a base classifier to compare the results of other algorithms against it.

Chapter 5

Training, Testing, and Model Selection

Our data is divided into two sets: test set and model selection set. To select the best model, the model selection set is used. To get the final score of the best model, the test set is used. Test set is comprised of the latest F days, where F was equal to the number of future days used for the prediction task. The model selection set is divided into two parts: training set and validation set. The best model with average highest accuracy was selected out of all the possible models described in section 5.3. The model's final score is then obtained using test set.

Model selection set was created using two different methods.

5.1 Train-Test-Split using shuffle

Since all the models for this experiment are atemporal models, the sequence while training the model should not make difference. While testing for X future days, last X data points from the data set are kept aside as a holdout set for calculating model's score after model selection. The remaining data were shuffled before splitting before creating training and testing sets. The models were then trained on the 80% of the data and validated on remaining 20% of

the test data. After selecting the best model with highest accuracy as described in section 5.3 on the validation data with all the hyperparameters mentioned in table 5.1, the model was then tested on the holdout set to get the final accuracy of the model.

5.2 K-fold cross validation

This train-validation-test set creation technique keeps the data in sequential order unlike Train-Test-Split. Train and validation set are divided into k equal size folds as depicted in figure 5.2. Different models as described in section 5.3 are validated on validation sets. The model with highest average accuracy is selected and the model's final score is obtained by testing the model on the holdout set. In this experiment, the test sets depicted in figure 5.2 are constructed in following ways:

- 5.2.1 An experiment is conducted with test sets are size of the number of future days. The tests are created in such a way to validate the model for only X future day predictions. Maximum 10 folds were constructed based on the size of the data using this testing technique. In further sections, this testing technique is referred as FD-CV.
- 5.2.2 The other experiment is conducted with tests are size of each fold. This is a traditional way the cross-validation tests are conducted for the purpose of model selection. In further sections, this testing technique is referred as Time-Series-CV.

5.3 Model Selection

Random forest, SVM, and KNN were used for the prediction task. All these algorithms have a set of hyperparameters, given in table 5.1, that can be adjusted to the optimal setting increase the performance of the model.

These hyperparameters were adjusted for each stock X for each future day Y to maximize the model's average validation score. To select the best model for Random forest algorithm

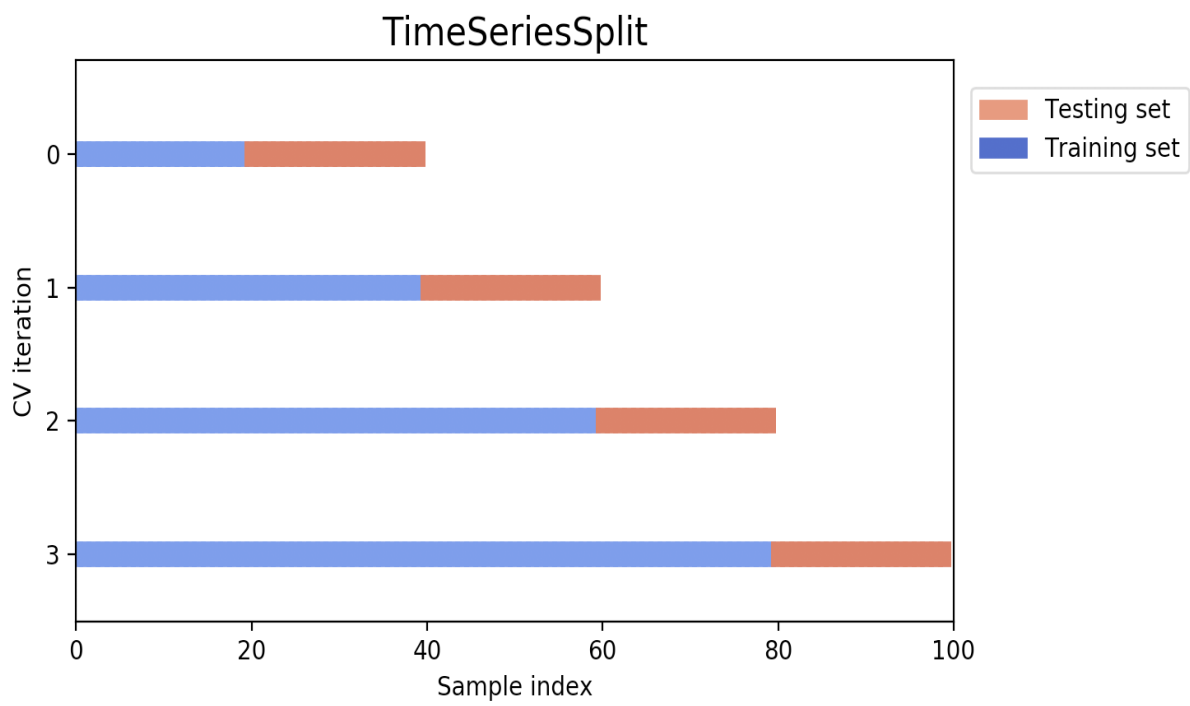


Figure 5.1: *Cross validation train and test sets*

Hyperparameter	Settings	Algorithm
Estimators	10, 20, 30, 40, 50	Random Forest
Max Depth	10, 20, 30, 40, 50	Random Forest
C	0.5, 1, 5, 10, 100	SVM
Kernel	linear, polynomial, rbf	SVM
Degree	1, 2, 3, 4, 5	SVM with polynomial
Neighbors	3, 5, 7, 9, 11	KNN
Distance functions	euclidean, manhattan, chebyshev, hamming, canberra, braycurtis	KNN
Feature selection	10, 11, 12, ... , 23	RF, SVM, KNN

Table 5.1: *Hyperparameters*

for stock X for the given future day Y, 25 models were created (5 Estimators settings with 5 Max depth settings, total 25 settings). The best model with highest average accuracy on the validation set was chosen. Similarly, 45 different settings were tried out for SVM and 30 different settings were tried out for KNN to select the best model for a given stock X for a future day Y. When considered feature selection for the testing, total of 14 different settings are tried out with each algorithm. Therefore, 350 settings for Random forest, 630 settings for SVM, and 420 settings for KNN are tried out to select the best model for a given stock X for a future day Y.

Chapter 6

Feature Selection

In the field of stock price prediction, to train the atemporal algorithms, features are one of the ways to summarize the temporal information. In this study, features are created to summarize the maximum possible temporal information.

6.1 Definition.

Feature selection, or variable selection, or attribute selection, is the way to select best features from the set of features for a given dataset.^[20]

Algorithms learn from the provided set of features. It is possible for an algorithm to learn with the less number of features than the complete set of features. To confirm this, I tested Random forest algorithm with features ranging from 10 to 22, over 100 stocks. Following are the results of this test.

6.2 Algorithms for Feature Selection

6.2.1 Recursive Feature Elimination

Recursive Feature Elimination (RFE) selects features by recursively considering smaller and smaller sets of features. RFE uses 'coef_' or 'feature_importances_' attributes of the estimator. Estimator is first trained on initial set either 'coef_' or 'feature_importances_', least important features are pruned. The same procedure is then repeated on the pruned set until the desired number of features are selected.

In the experiment, RFE is used for selecting the desired number of features for Random forest and SVM with tests mentioned in [5.2.1](#).

6.2.2 Mutual information

Mutual information of two random variables is a measure of the mutual dependence between the two variables. Mutual Information measures the importance of a term or a feature by testing how much the presence/absence of a feature contributes to making the correct classification decision.^[1] Based on the mutual information calculated for the given features, best 'k' features are selected. This algorithm is used for selecting best features for KNN and SVM with tests mentioned in [5.2.2](#).

Chapter 7

Experiment

To assess the efficacy of the algorithms described in Section 4, the methods were applied to the stock price data described in Section 3. Figure 7.1 describes the steps of the experiments.

The experiment is carried out in the following order:

1. Data collection: In this stage of the experiment, the data were collected for 10 randomly chosen stocks.
2. Data preparation: In this stage, the data has been cleaned to handle empty values present if any and different features were added which are described in sections 3.1.1-3.1.5.

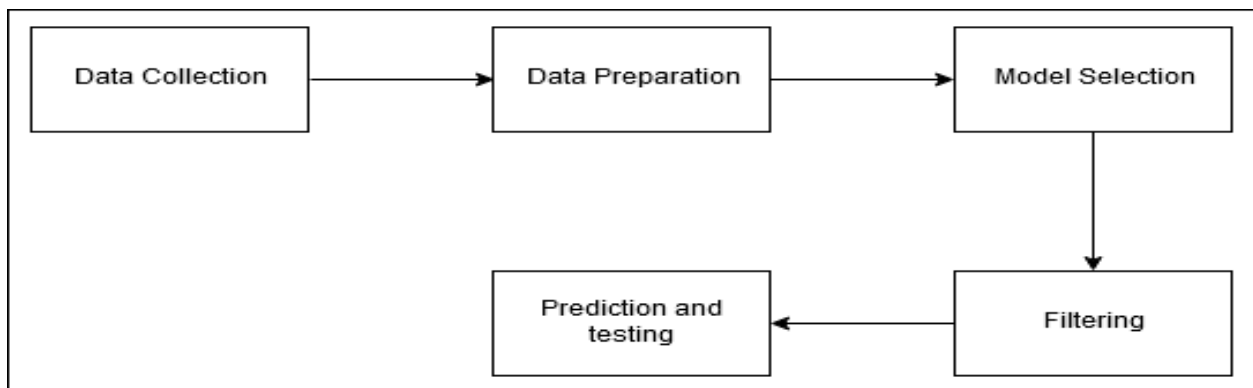


Figure 7.1: *Experiment Flow*

Overall experiment

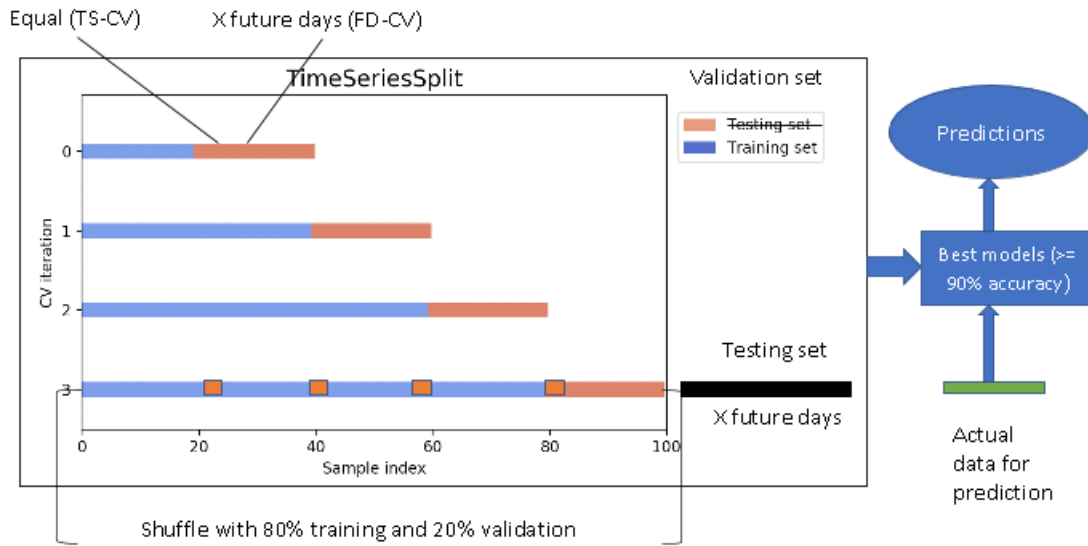


Figure 7.2: *Experiment details*

3. Model selection: Algorithms were trained on the prepared data to learn the involved correlation of the features to predict the stock's future behavior accurately. The best model was selected and the final accuracy of the best model is obtained according to strategy mention in 5.3.
4. Filtering: The models were filtered out according to the accuracy threshold. These selected models were then used for the prediction task on the real world data.
5. Prediction and testing: Given an unseen data without their class labels to the filtered models, the predictions of these models were compared against the real performance of the stock.

Figure 7.2 provides a deeper explanation of the experiment. The inner rectangle sheds the light on the different test methods. The output of the inner rectangle is the best model selected from different hyperparameter settings using model selection algorithm discussed in section 5.3. The best model's accuracy was calculated on the test set. The outer rectangle spits out the accuracies of the best models which was given to the accuracy filter. This filter

selected only those models which had accuracy equal to or greater than 90%. These models are then retrained with the complete available data. The 'actual data for prediction' is the data for which the class labels are not defined. The 'actual data for prediction' is then feed it into those models for prediction task. Final prediction creation is discussed in [9](#).

Chapter 8

Results

The experiment is performed using 3 different testing techniques discussed in section 5. For each test 10 randomly chosen stock's data were used. The best models were selected for Random forest, SVM, KNN, and ZeroR discussed in section 4 using model selection algorithm discussed in 5.3. For each algorithm and for each stock the experiments are conducted for future days ranging from 10 to 100 with the interval of 10, yielding 10 different settings for future days.

Following section discusses the results obtained by conducting these tests.

8.1 Discretized features only

Discretized features used in this experiment are discussed in section 3.3. The following three results show the accuracy graph using discretized features only.

8.1.1 Testing with train-test-shuffle

The figure 8.1 shows the results obtained for all 4 algorithms learned using the `train_test_split` testing technique mentioned in section 5.1.

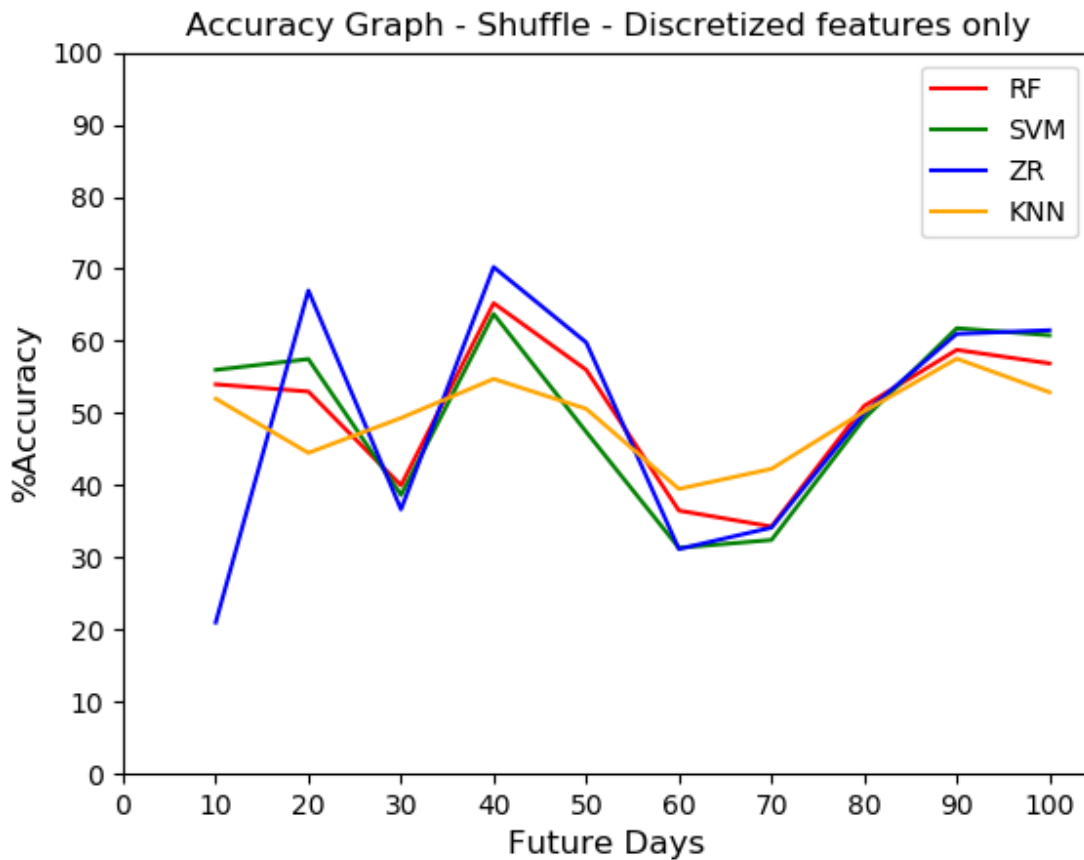


Figure 8.1: Accuracy graph of training algorithms using `train_test_split_shuffle` with discretized features only

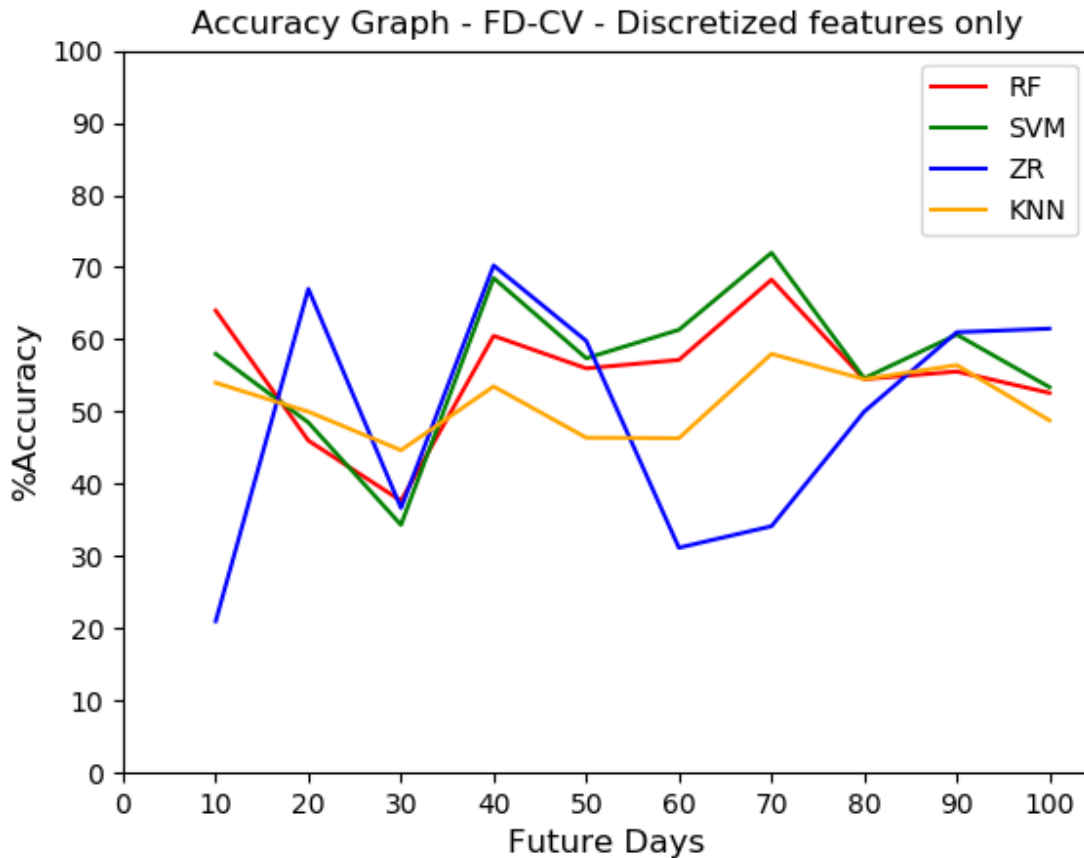


Figure 8.2: Accuracy graph of training algorithms using *FD-CV* with discretized features only

8.1.2 Testing with *FD-CV*

The figure 8.2 shows the results obtained for all 4 algorithms learned using the *FD-CV* testing technique mentioned in section 5.2.1.

8.1.3 Testing with *TS-CV*

The figure 8.3 shows the results obtained for all 4 algorithms learned using the *TS-CV* testing technique mentioned in section 5.2.2.

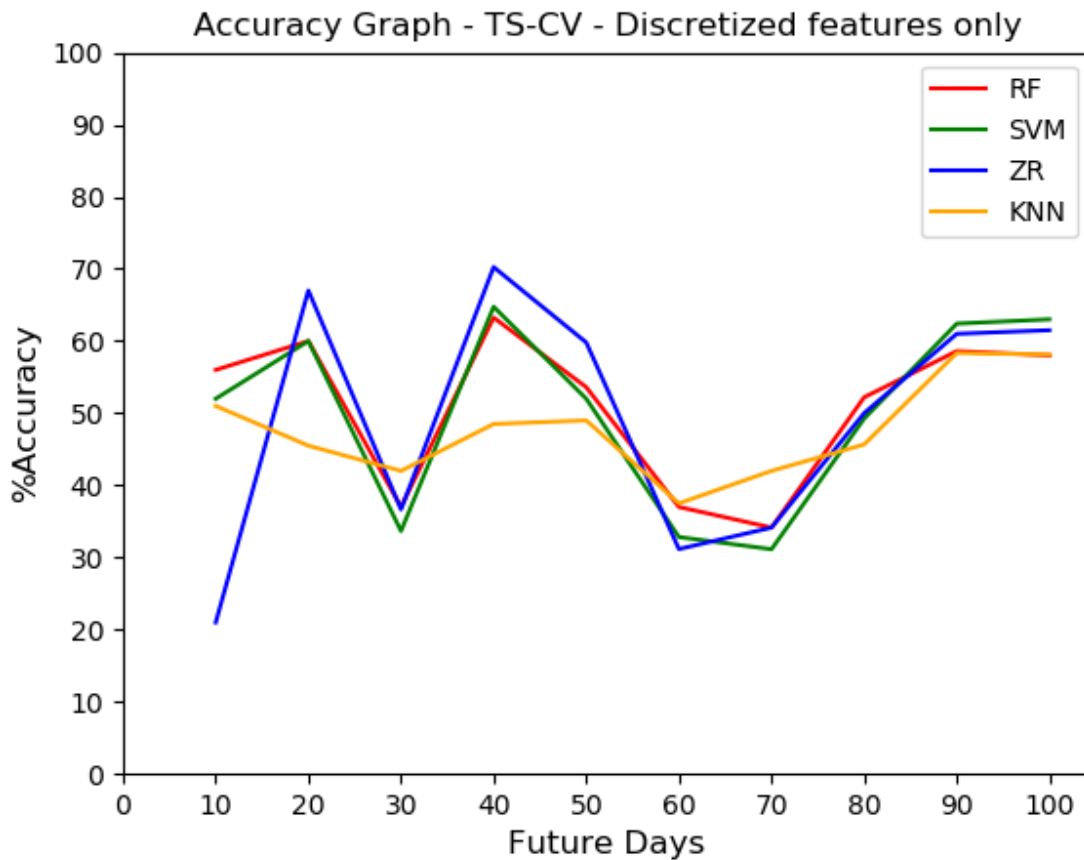


Figure 8.3: Accuracy graph of training algorithms using TS-CV with discretized features only

8.2 Feature selection

Out of 23 features used in this experiment, the subset of features were used to increase the performance of the models. These features were vetted using feature selection algorithms discussed in 6. Following graphs show the accuracy of using 3 different testing techniques with feature selection.

8.2.1 Testing with train-test-shuffle

The results shown in figure 8.4 were obtained for all 4 algorithms learned using all features with feature selection technique discussed in 6. The tests were conducted using the `train_test_split` testing technique mentioned in section 5.1 with Feature Selection.

8.2.2 Testing with FD-CV

The figure 8.5 shows the results obtained for all 4 algorithms learned using the FD-CV testing technique mentioned in section with feature selection 5.2.1.

8.2.3 Testing with TS-CV

The figure 8.6 shows the results obtained for all 4 algorithms learned using the TS-CV testing technique mentioned in section 5.2.2 with Feature Selection.

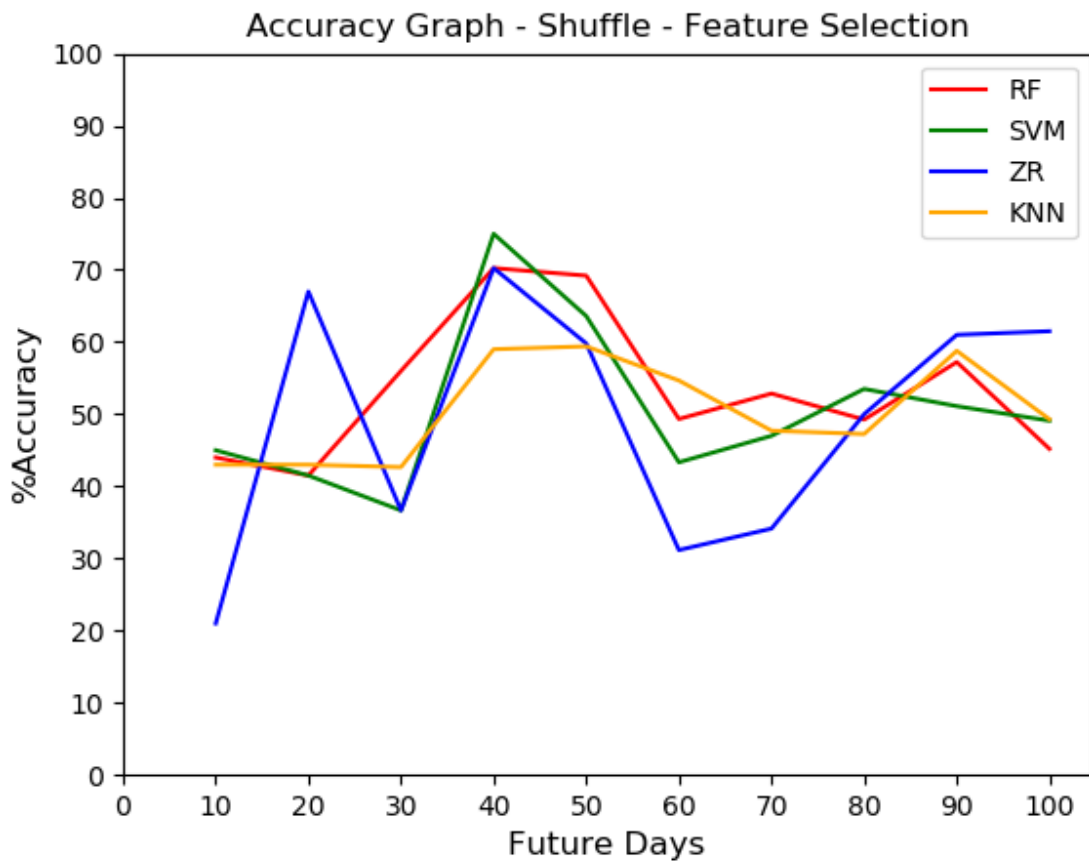


Figure 8.4: Accuracy graph of training algorithms using `train_test_split` with `shuffle` with Feature Selection

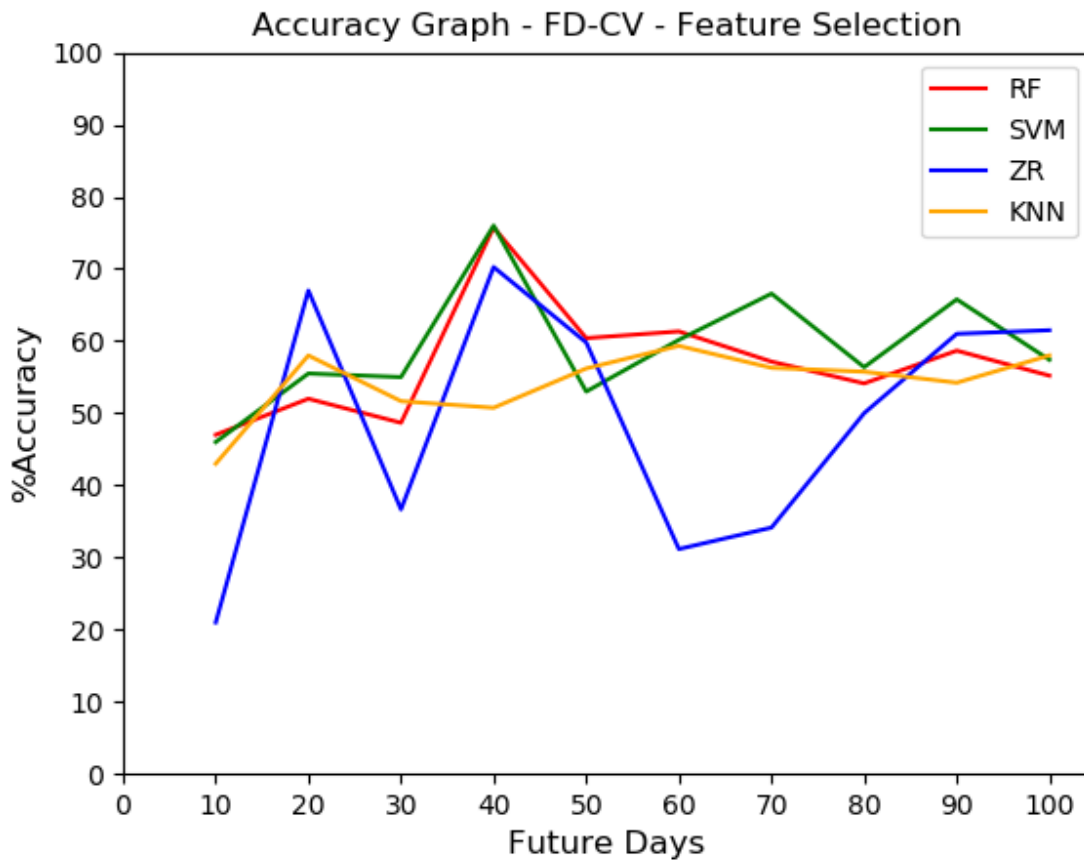


Figure 8.5: Accuracy graph of training algorithms using FD-CV with Feature Selection

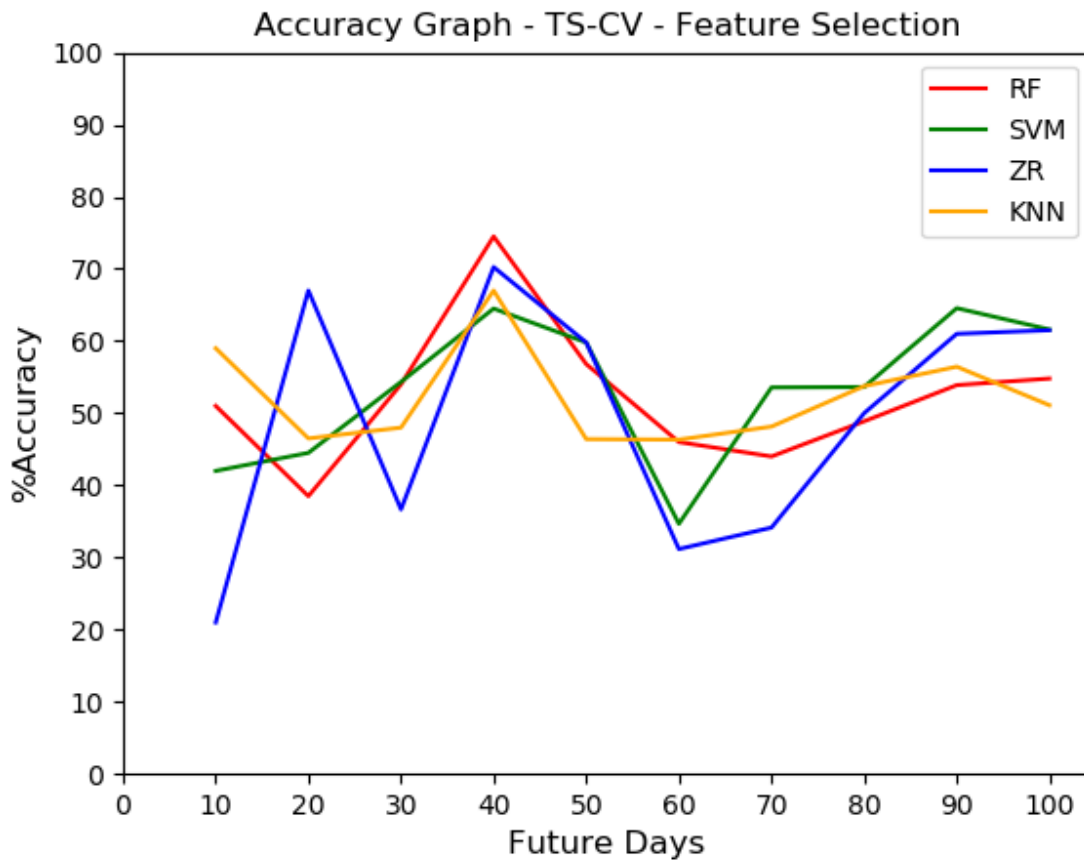


Figure 8.6: Accuracy graph of training algorithms using TS-CV with Feature Selection

8.3 Analysis

As per the gathered results, FD-CV test method trained models better than shuffled and TS-CV test models when tested with discretized features only. While training with all features using feature selection, FD-CV test method trained models better than shuffled and TS-CV test models.

When trained with discretized features only, shuffling the data for training does not produce better results than the baseline ZeroR model using all other algorithms as observed in 8.1. Predictions for all future days except 10, 60, and 70 are giving marginally better results with ZeroR model. Similar performance can be seen by test method FD-CV. The difference is that the FD-CV test method resulted in better trained models for future days 10, 60, and 70 compared to shuffled method as they resulted in 40% better accuracy. ZeroR model also dominated or performed equally well for TS-CV testing method.

For feature selection tests, Shuffled method resulted in marginally better accuracy with RF, KNN, and SVM than the ZeroR model as depicted in 8.4. FD-CV method also resulted in slightly better accuracy for future days 10 to 50 as captured in 8.5. For future days 60 to 80, FD-CV resulted in 40% better accuracy, same as using discretized features only. TS-CV method also showed slightly better results than ZeroR model which are shown in 8.6. Out of all the algorithms used in this experiment, SVM and Random Forest showed better accuracy to some extent.

Chapter 9

Prediction

After model selection, the list of best models for different future days were selected for testing and the final score was obtained by testing each model with test cases of size X for a stock Y . For the final prediction, only the models which performed with higher accuracy on the holdout set were selected. The higher accuracy threshold was kept at 90%. These models then retrained on the complete available data and subjected to the task of prediction on the unseen data. Here, the model's accuracy on the unseen data acted as a type of filter to select only best models which performed well on the unseen data. This increases the likelihood of the model to perform well on the latest unseen data for which the labels are not present in the data.

The final prediction is calculated by taking all the predictions from selected models into consideration and selecting the mode of the predicted class. If there is a tie between two classes, then it was resolved by treating it like 0 or hold class. If the final outcome is 0 or hold class, then the prediction is considered as don't buy. The results in table 9.1 and table 9.2 are observed using the prediction strategy mentioned here. Figure 9.1 depicts the prediction process.

As described in section 2, the data were collected until September 10, 2019. For the models which met accuracy threshold, the latest X days from the data were used for this

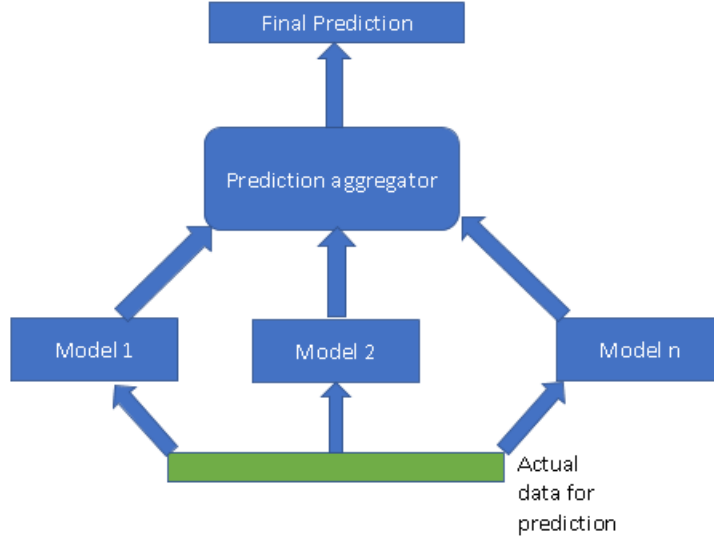


Figure 9.1: *Prediction aggregator*

prediction task. Here, X is equal to the future days for which model met the accuracy threshold. In this way, total X predictions were returned by the model. The maximum occurring class label from the returned predictions was chosen as the final prediction. This heuristic was applied to remove the noise and taking the overall picture into account.

Test method	# of models selected	# of stock(s) selected / 10	# of correct predictions
Shuffle-Disc	10	4	4
Shuffle-FS	14	5	2
FD-CV-Disc	17	5	4
FD-CV-FS	27	7	6
TS-CV-Disc	12	4	3
TS-CV-FS	12	4	3

Table 9.1: *Prediction results with different test methods*

The results in table 9.1 were observed for the best models from 3 different testing methods using two different feature methods - Discretization (Disc) and feature selection with all the features (FS). Two methods - Shuffle-Disc and FD-CV-FS stand out from everyone else. Although Shuffle-Disc produced minimum number of models with 90% or higher accuracy, it has 100% accuracy on the predictions it produced for the stocks. On the other hand,

FD-CV-FS produced maximum number of models with 90% or higher accuracy. It also selected most number of stocks (7 out of 10) and predicted the profit-loss behavior of 6 of them correctly.

Test method	Random Forest	SVM	KNN	ZeroR
Shuffle-Disc	2/3	4/4	1/1	3/5
Shuffle-FS	4/5	1/4	4/5	3/5
FD-CV-Disc	1/2	4/5	0/1	3/5
FD-CV-FS	3/6	5/6	4/4	3/5
TS-CV-Disc	2/3	3/4	1/1	3/5
TS-CV-FS	2/2	1/3	2/2	3/5

Table 9.2: *Prediction results with different algorithms*

The results in 9.2 looks at the algorithm wise prediction accuracy. This way we can compare the algorithms' accuracy with different test methods to ZeroR algorithm. Using test method shuffle-disc, all three algorithms used in this experiment performed substantially better than ZeroR algorithm. Using test method FD-CV-FS, all algorithm except Random forest achieved better accuracy than the ZeroR. Also, TS-CV-Disc test method helped all algorithms achieve greater accuracy than the ZeroR.

Finally, the average accuracy of all the test methods using combined prediction mentioned in 9.1 of all the algorithms is 75.952% which is greater than ZeroR (60%).

Chapter 10

Conclusion

In this experiment, three methods were tested for model selection, described in 5 with two different data preparation techniques (Discretized and feature selection) for 10 randomly chosen stocks from NYSE. After selecting best models and tested on the unseen data. As per the observations depicted in section 8, FD-CV with feature selection technique for model selection showed better performance than any other model selection technique. Random forest and SVM performed fairly better than KNN in all the techniques.

As seen in figures 8.3, 8.2, and 8.1 ZeroR model performed equally better than the other algorithms. Algorithms learned with Feature selection showed slightly better results than ZeroR algorithm.

After training and selecting the best models, the accuracy of each best model was obtained by testing on holdout data set. This accuracy was considered for further elimination of models which were lesser than 90% accurate. These higher accuracy models were then used for the final prediction task. The results showed that two methods, Shuffle training set with discretized features only and FD-CV method with feature selection, stand out and showed better results with their predictions.

Chapter 11

Future Work

Even though predictions performed to a level that is higher than a ZeroR baseline, the performance can be further improved by increasing the accuracy of models before filtering out the models with accuracy threshold. This will assure that there will be more models participating in the final prediction round which means more insights on the data. The goal of future work is to increase the models' accuracy in order to bring more and more models reach the accuracy threshold.

Another idea is to discretize as many features as possible. According to^[18], there was a performance increase by discretizing the features. Therefore, in future more discretized features can possibly increase the model's accuracy.

There is also a possibility of increasing models' accuracy by introducing more relevant features as introduced in this experiment. Letting models to work with more data with feature selection likely to increase the accuracy of models.

Finally, different test methods with different accuracy threshold can be tried out with randomly chosen stocks.

Bibliography

- [1] URL <https://nlp.stanford.edu/IR-book/html/htmledition/mutual-information-1.html>. This link gives the information and formula of mutual information.
- [2] Momentum versus rate of change: Which indicator does the job better? URL <https://commodity.com/technical-analysis/momentum/>.
- [3] Relative strength index (rsi). URL <https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/rsi>.
- [4] Slow stochastic. URL <https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/slow-stochastic>.
- [5] Williams %r. URL https://school.stockcharts.com/doku.php?id=technical_indicators:williams_r.
- [6] URL <https://www.investopedia.com/terms/a/accumulationdistribution.asp>.
The A/D indicator used in the experiment is calculated without multiplying the volume.
- [7] Adjusted closing price. URL https://www.investopedia.com/terms/a/adjusted_closing_price.asp.
- [8] URL <http://eoddata.com/stocklist/NASDAQ.htm>. [List of tickrs were gathered from this website].
- [9] Support vector machines. URL <https://scikit-learn.org/stable/modules/svm.html>.
- [10] Longman business english dictionary.

- [11] Exponential moving average - ema definition, 2019. URL <https://www.investopedia.com/terms/e/ema.asp>.
- [12] Simple moving average (sma), 2019. URL <https://www.investopedia.com/terms/s/sma.asp>.
- [13] J Martin Bland and Douglas G Altman. Statistics notes: Measurement error. *BMJ*, 312(7047):1654, 1996. doi: 10.1136/bmj.312.7047.1654. URL <https://www.bmj.com/content/312/7047/1654>.
- [14] Craig A Ellis and Simon A Parbery. Is smarter better? a comparison of adaptive, and simple moving average trading strategies. *Research in International Business and Finance*, 19(3):399–411, 2005.
- [15] Pablo Fernández-Blanco, Diego J Bodas-Sagi, Francisco J Soltero, and J Ignacio Hidalgo. Technical market indicators optimization using evolutionary algorithms. In *Proceedings of the 10th annual conference companion on Genetic and evolutionary computation*, pages 1851–1858. ACM, 2008.
- [16] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.
- [17] Mansoor Maitah, P. P. Procházka, Michal Cermák, and Karel rédl. Commodity channel index: Evaluation of trading rule of agricultural commodities. 2016.
- [18] Patel, Shah, Thakkar, and Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42(1):259–268, 2015.
- [19] Jigar Patel, Sahil Shah, Priyank Thakkar, and Ketan Kotecha. Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, 42(4):2162–2172, 2015.

- [20] Wikipedia contributors. Feature selection — Wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/w/index.php?title=Feature_selection&oldid=920990787. [Online; accessed 31-October-2019].
- [21] Wikipedia contributors. Fourier transform — Wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/w/index.php?title=Fourier_transform&oldid=923479351. [Online; accessed 31-October-2019].
- [22] Wikipedia contributors. K-nearest neighbors algorithm — Wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/w/index.php?title=K-nearest_neighbors_algorithm&oldid=920309707. [Online; accessed 30-October-2019].
- [23] Wikipedia contributors. Kurtosis — Wikipedia, the free encyclopedia, 2019. URL <https://en.wikipedia.org/w/index.php?title=Kurtosis&oldid=917517028>. [Online; accessed 14-October-2019].
- [24] Wikipedia contributors. Standard deviation — Wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/w/index.php?title=Standard_deviation&oldid=920310382. [Online; accessed 14-October-2019].
- [25] Wikipedia contributors. Skewness — Wikipedia, the free encyclopedia, 2019. URL <https://en.wikipedia.org/w/index.php?title=Skewness&oldid=920310357>. [Online; accessed 14-October-2019].
- [26] Wikipedia contributors. Stock — Wikipedia, the free encyclopedia, 2019. URL <https://en.wikipedia.org/wiki/Stock#Shares>. [Online; accessed 14-October-2019].
- [27] Wikipedia contributors. Support-vector machine — Wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/w/index.php?title=Support-vector_machine&oldid=921383403. [Online; accessed 30-October-2019].