QUANT REPORT

2023 Spring Vol. 1

KRISS
HKUST KSA Research & Investment Strategy Society

Predicting the Stock Price Flow Using Random Forest Algorithm

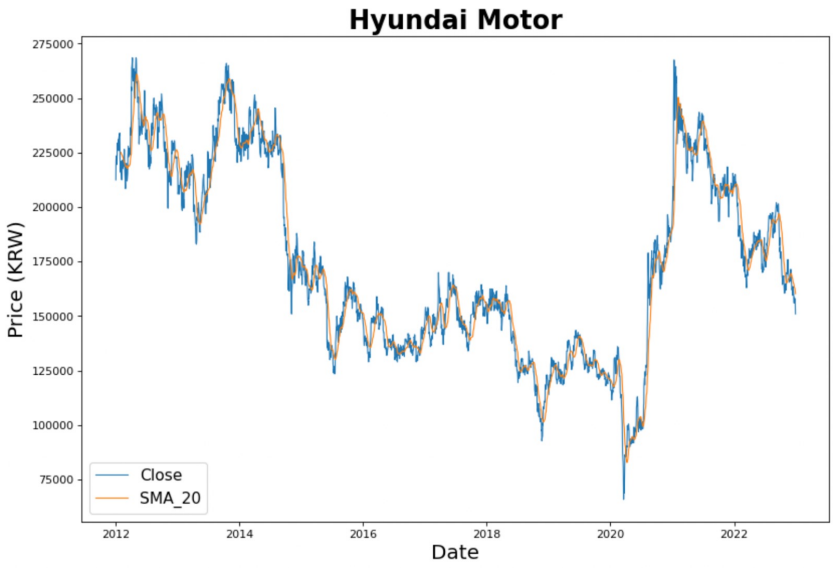
HONG, Jung Bin
Quant Analyst
IEDA @ HKUST

KIM, Min Ji
Quant Analyst
MATH-CS @ HKUST

CHO, Hangsun
Head Quant Analyst
CPEG @ HKUST

CHOI, Bokyung
Head Quant Analyst
MATH-CS @ HKUST

The use of machine learning algorithms in the stock market has become increasingly popular in contemporary finance. Stock price forecasting and predictions can be challenging, as stock prices are trend oriented and prone to many external factors such as global politics, economy, and internal factors including the company's financial performance. Machine learning and AI algorithms apply statistical models on historical data to draw insights and then to generate predictions. This report will specifically focus on predicting the stock price of 'Hyundai Motor', through the random forest algorithm, a branch of machine learning. The algorithm will then be backtested to evaluate how accurate the created predictions are to the actual price flow.



Theoretical Background

Decision Trees

Decision Tree is a type of supervised machine learning algorithm that learns existing data by creating a combination of decision rules in the form of trees and then carrying out predictions. Features related to decision making make up the decision nodes in the tree, acting as a means to split the data. Questions are used to help individual trees to reach a final decision, indicated by the leaf node. Observations that meet the criteria will be placed along the "Yes" branch, while those that do not will be categorized along an alternate path. Decision trees aim to identify the best split for data, and are commonly trained through the Classification and Regression Tree (CART) algorithm.

Decision Trees

Random Forest is an algorithm that complements the overfitting problem of the decision tree algorithm. It collects and combines the output results from multiple decision trees to reach a single result. Random forest uses ensemble learning methods such as bagging(also known as bootstrap aggregation). During the bagging process, a random sample of data is selected with replacement from a training set, enabling individual data points to be chosen multiple times. The models are then trained independently and based on the type of task (e.g. regression or classification), the average or majority of the predictions yields a more accurate estimation.

Methodology

Feature Selection

Besides from the default data features offered from the FinanceDataReader library, additional features have been extracted to improve the prediction model:

RSI – Relative Strength Index

The Relative Strength Index is a momentum indicator used in technical analysis to indicate whether a stock is overbought or oversold. Overbought and oversold simply indicates that the stock price is approaching the highs and lows from its

recent price range. It is calculated by taking the average of closing prices' upward movements, divided by the average of closing prices' downward movements over a period of time. This ratio is then utilized to calculate the RSI. The RSI is commonly used to identify potential trend reversals, with a reading above 70 implying the market is overbought and a reading below 30 implying it is oversold.

$$RS = \frac{Average\ Gains}{Average\ Losses}, \qquad RSI = \frac{100}{1 - RS}$$

Williams %R

The Williams %R is another momentum indicator that indicates if a stock has been overbought or oversold within a time frame. The indicator moves between 0 and -100, where a value close to 0 is overbought and a reading below -80 is oversold. The indicator is calculated by the following equation:

$$\%R = \frac{Highest\ High\ - Closing\ Price}{Highest\ High\ - Lowest\ Low} \times (-100)$$

Simple Moving Average (SMA)

The simple moving average is an indicator that shows the average stock price movement over a designated time frame. For this feature, three intervals have been selected: SMA20, SMA120, SMA240 to represent the short, mid and long-term intervals, where the number represents the number of days for which the stock price average will be taken.

$$SMA = \frac{A_1 + A_2 + \cdots + A_n}{n} \quad \text{where } A_k \text{ is the closing price at day } k$$

Assumption

Since the goal was to generate a model to predict whether the stock price will go up or go down the next day, a new prediction column was created containing values of either 0 or 1. New values from the test data set were trained and classified them into these two groups. The 'Change' column was selected to obtain the price change. If the change is negative, then it indicates that the current price at time 't' is lower compared to time 't-1'. By applying the same logic for the positive value of the 'Change' column, the prediction columns were filled for the dataset prepared.

The output of the prediction column was classified as '1' if the closing stock price at time 't' was greater than the closing price at 't-1'. For simplicity purposes, to keep the prediction column a binary classifier, as long as the price did not decrease from time 't-1', the prediction was classified as 1. Below is the final dataframe used to build the random forest model.

Dataframe used to build the random forest model

Date	Open	High	Low	Close	Volume	Change	RSI_14	RSI_5	SMA_20	SMA_60	SMA_240	r_percent	SMA_120	Prediction
2012-06-26	240000	241000	237000	238500	292563	-0.010373	45.395795	31.754021	241500.0	248558.333333	235129.166667	-83.720930	235129.166667	-1.0
2012-06-27	238000	238500	228000	231000	1072171	-0.031447	38.595735	20.064365	241175.0	248466.666667	235283.333333	-89.473684	235283.333333	-1.0
2012-06-28	232500	235000	229500	234000	462154	0.012987	42.431095	32.829564	240725.0	248483.333333	235387.500000	-78.947368	235387.500000	1.0
2012-06-29	229000	233000	226500	232500	933968	-0.006410	41.067587	29.885027	240150.0	248358.333333	235462.500000	-80.000000	235462.500000	-1.0
2012-07-02	236500	237500	234500	235500	438405	0.012903	44.905451	42.794840	240025.0	248033.333333	235562.500000	-70.000000	235562.500000	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2022-12-23	157000	158000	155500	157000	336597	-0.009464	37.449320	35.414268	163100.0	166850.000000	179083.333333	-76.190476	179083.333333	-1.0
2022-12-26	157000	159000	157000	158000	387872	0.006369	40.016663	43.012960	162625.0	166458.333333	178916.666667	-66.666667	178916.666667	1.0
2022-12-27	158500	160000	157000	158000	637139	0.000000	40.016663	43.012960	162075.0	166150.000000	178754.166667	-66.666667	178754.166667	1.0
2022-12-28	156000	156500	154000	154000	556921	-0.025316	33.650036	24.853271	161325.0	165725.000000	178600.000000	-100.000000	178600.000000	-1.0
2022-12-29	153000	154000	150500	151000	552974	-0.019481	29.730354	17.675514	160300.0	165291.666667	178400.000000	-96.551724	178400.000000	-1.0

Process

In this report, the python library FinancialDataReader was utilized to retrieve the default data of the stock (005380). The default data includes: 'Open', 'High', 'Low', 'Close', 'Change', and 'Volume'. Among the default features, other than 'Close' and 'Volume', no other feature showed a significant correlation with stock price prediction. Therefore a total of 8 features for the random forest model used for predicting the flow of the stock price.

```
# Select X & Y Columns (X = features, Y = predictins)
X_Cols = df[['Close', 'Volume', 'RSI_14', 'RSI_5', 'SMA_20', 'SMA_120',
'SMA_60', 'r_percent']]
Y_Cols = df['Prediction']

# Split Dataset into Training and Test set (8:2)
X_train, X_test, y_train, y_test = train_test_split(X_Cols, Y_Cols,
test_size=0.2, random_state=0)
```

The dataset included historical data of Hyundai Motors in a time interval between 2012 - 2022. Using the train_test_split function from the SciKit Learn package, the data was split into a training set and test set with a ratio of 8:2 during this time frame. The random forests algorithm also has various hyperparameters (model external adjustment variables). Depending on these settings, the performance of the model can vary greatly. Therefore, after some trial and error was done to find the appropriate hyperparameters for the model. The final hyperparameter values used in the model training is as listed below:

```
# Create a Random Forest Classifier
rand_frst_clf = RandomForestClassifier(n_estimators = 100,
max_leaf_nodes = 40, max_features = 8, oob_score = True, criterion =
"gini", random_state = 0

# Split Dataset into Training and Test set (8:2)
rand_frst_clf.fit(X_train, y_train)

# Make predictions
y_pred = rand_frst_clf.predict(X_test)
```

The number of max features was set to 8, allowing all 8 features to be included in building each decision tree for the random forest. The number of estimators (number of decision trees) was set to 100 - which is the default value. By trial and error, the default value produced satisfactory training results with quicker running time.

Increasing the value of max_leaf_nodes may increase the model's ability to capture complex relationships in the data, which can result in better performance on the validation or test data. However, this will allow a more complex model that may overfit the training data, meaning it may perform well on the training data but poorly on new, unseen data. On the other hand, decreasing the max_leaf_nodes can result in a simpler model that may generalize better to new, unseen data, but it may also lead to underfitting, where the model is too simple to capture the underlying relationships in

the data. Therefore after taking into account both the advantages and disadvantages, the max_leaf_node was set to 40.

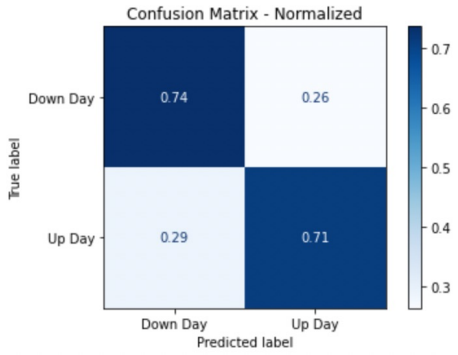
From the trained model, we have validated the model with the aforementioned test set. A training accuracy of 0.80 and testing accuracy of 0.74 was obtained from the trained model.

```
accuracy_score of train data(0.8 of sample): 0.8083976833976934
accuracy_score of test data(0.2 of sample): 0.7432432432432432
```

Results

Confusion Matrix and Classification Report

Confusion Matrix

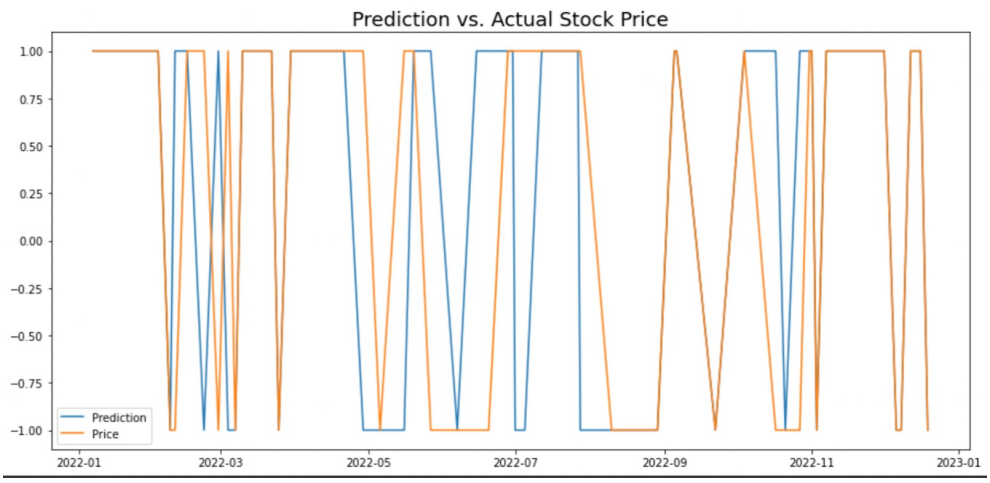


Classification Report

	precision	recall	f1-score	support
Down Day	0.713656	0.704348	0.708972	230.000000
Up Day	0.766323	0.774306	0.770294	288.000000
accuracy	0.743243	0.743243	0.743243	0.743243

The confusion matrix helps illustrate the performance of the generated model by showing how accurate the predictions were to the actual value. The classification report gives a more detailed overview of the performance by computing the Precision, Recall, and the F1_Score. ‘Precision’ measures the proportion of correctly predicted samples from the predicted pool of positives whereas ‘Recall’ is the measure of how many positive cases the model predicted correctly out of the total number of actual positives. The F1_Score is a harmonic mean of the two indicators. Ideally, all these indicators should be as close to 1 as possible. For the generated classification report, the precision, recall and F1_Score all surpass 0.7 - which is an acceptable level of accuracy.

Prediction vs. Actual stock price rise and fall



To verify the statistics indicated in the aforementioned reports, a graph of the 'Predictions vs. Actual Stock Price' was plotted for the year 2022. This was done by using the probability of the up/days from the random forest selection. Based on the graph, the most obvious observable trend is that the prediction modeled by the random forest has a tendency to precede the actual stock price movement. As the date gets closer to the end of the year, the predictions and the actual stock price movement synchronize and the model does a better job at predicting the movement of the stock price.

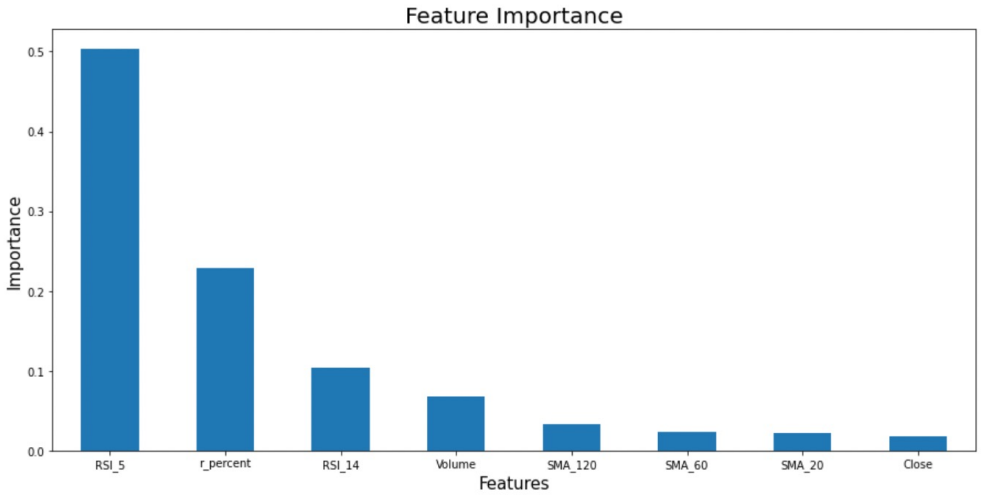
```
result['compare'] = result['Prediction'] == result['result']
cal = len(result.loc[result['compare'] == True])/len(result['compare'])
print(cal)

# 0.7741935483870968
```

In order to quantify the accuracy, the number of true positive and true negative predictions were calculated as a percentage, which resulted in 77.4%. This approximately matches the values calculated from the confusion matrix and the classification report.

Feature Importance

Prediction vs. Actual stock price rise and fall



Feature importance was calculated and extracted as a bar chart to observe which features had the most impact in predicting the stock price. From the chart displayed above, the 'RSI_5' and the 'Williams %R' were the two most dominant features. Both features are momentum indicators that take account of the most recent trends of the stock price through either its recent average or the maximum and minimum stock price from a recent time frame. Therefore, more weights are assigned to the most recent data that may better illustrate the current trend of the stock. This may show that although momentum indicators may at times capture false signals during trending zones, they are still effective indicators in estimating future stock prices because more often than not, stock markets do not undergo extreme bullish or bearish movements. Conversely, features such as the closing price, and simple moving averages showed minimal impact to the overall prediction.

Out of Bag Error

```
Random Forest Out-Of-Bag Error Score: 0.7268339768339769
```

The Out of Bag Error (OOB) is another method to validate the performance of a random forest. Recalling that duplicate entries were allowed in the bootstrapped dataset, some datasets from the original data are not selected in the training process. Running the omitted data through the random forest model and evaluating

the predicted values for these samples computes the out of bag error. The OOB Error for the random forest model was approximately 0.73, which reflects the model was trained with consistency and without running into overfitting problems.

Conclusion

The built random forest model can only estimate whether a stock price of Hyundai Motors will increase or decrease based on the recent price movement reflected by the selected features such as the RSI and Williams %R. The estimated prediction can only be a reference point to help plan for an investment strategy, as it only showed an accuracy of approximately 72%. This is because there are still limitations to its predictive power. Factors such as unexpected news events or market changes can greatly impact the stock prices and render the model's predictions inaccurate. It is important to recognize these limitations and utilize the model's predictions as just one of several tools for making informed investment decisions.

Furthermore, there are no indications of how much the stock price would increase or decrease depending on the predicted value. Since the random forest algorithm is particularly effective with a large dataset for regression analysis, for future research, a larger dataset would be required for quantitative prediction of the stock price.

<References>

Kim, D.Y. (2019, November 18). "코스피 단기예측 AI 모델" Samsung Securities Co., Ltd. Retrieved March 1, 2023, from https://www.samsungpop.com/common.do?cmd=down&saveKey=research.pdf&fileName=5020/20191105150227 89K_01_32.pdf&contentType=application/pdf

Mitchell, C. (2022, October 5). Williams %R: Definition, formula, uses, and limitations. Investopedia. Retrieved March 1, 2023, from https://www.investopedia.com/terms/w/williamsr.asp

What is a decision tree. IBM. (n.d.). Retrieved March 1, 2023, from https://www.ibm.com/hk-en/topics/decision-trees

KRISS
HKUST KSA Research & Investment Strategy Society

HKUST KRISS
KSA Research and Investment Strategy Society

LinkedIn: www.linkedin.com/company/80132446/
E-mail: kriss.hkust@gmail.com