
Prediction of Stock Market Price Index using Machine Learning and Global Trade Information

Eugene L.X. Wong
Pratt School of Engineering
Duke University
eugene.wong@duke.edu

Abstract

Globalization has led to an increasingly integrated global economy, one with less trade barriers and more capital mobility between countries. Consequently, no country is an island of its own. In this paper, it aims to investigate how global trade affects a country's stock market and also determine if such information with the use of machine learning techniques can predict a country's stock market index.

1 Introduction

In many machine learning literature, a common approach is to feed the model with as much data as possible before selecting the features of importance. If the features are intuitive and clearly relational to the target variable, it is easy to comprehend the underlying logic. However, in some cases, the relationship between the feature and the target variable may lack clarity, and the explainability of the effect remains inconspicuous. For example, why is it that the movement of U.S stocks affects Japan's stock market index. Contrarily to most literature, this paper will first explore the hypothesis, that is, the degree of trade relations between the target country and other countries will influence it's stock market index. It will also showcase a viable case where implicit information from the trade relations can be used to predict the stock market index. In this paper, the target country used in the analysis will be Japan and therefore, the stock market index in focus will be Tokyo Stock Price Index, commonly known as TOPIX, an important stock market index that tracks all domestic companies in Tokyo Stock Exchange First Section. The input to the algorithm will be open, close, high and low index price data from countries that have significant trade relations with Japan. Various machine learning models such as Logistic Regression, Random Forest and Support Vector Machines (SVM) will be deployed to output the predicted price direction of TOPIX, that is, positive, negative or neutral, in the next trading day. Finally, each of the model's performance will be evaluated using an array of metrics such as accuracy, profit and Sharpe Ratio.

2 Related work

In the recent years, a wide range of machine learning and deep learning techniques have been implemented to financial market forecasting. It is not surprising to see an increasing adoption of these algorithms because they tend to outperform traditional stochastic methods (1) (2). In several literature (3) (4), popular algorithms such as logistic regression, decision tree and neural networks have often been deployed with an objective of maximizing the portfolio returns. While most literature emphasizes the prediction accuracy of their models, there is only one literature (5) that has attempted to unravel the underlying relationship between features and how they influence the market. In due to the scarcity in research in this area, this paper will focus to test a hypothesis related to the spillover effects from trading partners and to showcase its influence on the stock market.

3 Dataset and Features

3.1 Dataset

The dataset encapsulates various indices as shown in Figure 1, with each index comprising of Open, Close, High and Low prices¹. The dataset extends across the periods from March 2010 to November 2020, of which 80% (2010-03-23 to 2018-09-12) of the data will be used to train the model and the remaining 20% (2018-09-13 to 2020-11-06) will be used to evaluate the model performance. In this section, the details of how the features were engineered, the normalization method, and how labels (Buy/Sell/Neutral) were created will be described here.

Figure 1: Countries' indices

3.2 Feature Engineering

The derivation of inter-day, intra-day and volatility features will be explained here.

3.2.1 Inter-day Features

The inter-day returns of prices are computed as follows:

$$R_{mn} = \frac{x_m(t) - x_n(t+1)}{x_n(t+1)} \quad (1)$$

where R_{mn} is the general expression for Close-to-Close (R_{cc}) and Close-to-Open (R_{oc}) returns. The time, t , refers to the present trading day (now).

3.2.2 Intra-day Features

The intra-day returns of prices are computed as follows:

$$R_{mn} = \frac{x_m(t) - x_n(t)}{x_n(t)} \quad (2)$$

where R_{mn} is the general expression for Open-to-Close (R_{co}), High-to-Close (R_{hc}) and Low-to-Close (R_{lc}) returns. The time, t , refers to the present trading day (now).

¹Exception of FBMKLCI & PSEI as HLO prices are not available

3.2.3 Volatility Feature

The rolling volatility is computed as follows:

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (3)$$

where s is the general expression for volatility and \bar{x} is the mean of x over N period. In order to calculate the rolling volatility, the time period is shifted by one trading day everyday, while encapsulating N period of x .

3.2.4 Normalization

The indices in Figure 1 have different range of values, and if used to train the models directly, it may lead to sub-optimal hyperparameters. Consequently, a MinMax normalization method is deployed to rescale each feature to a common scale. As such, this will allow gradient descent to converge quickly and to reach an optimal point. The MinMax normalization is defined as:

$$x_{std} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4)$$

$$x_{rescaled} = x_{std} (max - min) + min \quad (5)$$

where the features are scaled to the range between the arbitrarily min and max values. In this paper, the values of min and max were set to 0 and 1 respectively.

3.2.5 Label Creation

TOPIX (Open-Close) returns were partition by threshold values to create three labels, namely, Buy, Sell and Neutral. The algorithm implemented to create this threshold is defined as:

Algorithm 1: An algorithm to create labels

```

if Threshold Value >= [Median( $R_{oc}$ ) + (0.5 * Median( $R_{oc}$ ))] then
  | Assign Buy label ;
else if Threshold Value <= [Median( $R_{oc}$ ) - (0.5 * Median( $R_{oc}$ ))] then
  | Assign Sell label ;
else
  | Assign Neutral label ;
  
```

The proportion of Buy, Sell and Neutral labels in the preprocessed dataset are exhibited in Figure 2. Although the ratio of Buy/Sell to Neutral is relatively large, training the model according to this data distribution does not make it a bad model. Essentially, the fitted model will be more skewed to execute a buy/sell position and less so to take no action, resulting in higher daily trading frequency.

Counts	
Buy	1822
Sell	1826
Neutral	26

Figure 2: Label counts in preprocessed data

Figure 3: New features and data points generated

4 Methods

4.1 Logistic Regression

The logistic regression classifier was trained with a One-vs-All strategy that involves training N distinct binary classifiers, each to predict a specific class (i.e. Buy, Sell and No Action). The logistic regression hypothesis is defined as:

$$h(x) = g(w^T x) \quad (6)$$

where g is the sigmoid function and the parameters of the model is given by w . The sigmoid function is defined as:

$$g(z) = \frac{1}{1 + e^{-z}} \quad (7)$$

The objective of logistic regression is to minimize the cost function which is given by:

$$J(w) = \frac{1}{m} \sum_{i=1}^n \left[y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log (1 - h(x^{(i)})) \right] \quad (8)$$

In order to improve the model's accuracy and to reduce overfitting, L2 regularization was implemented. The Ridge Penalty, L2 norm, adds the squared magnitude of the coefficients as a penalty term to the cost function. Increasing the lambda reduces the magnitude of the parameters, thus less predictive features. The regularized cost function in logistic regression is defined as:

$$J(w) = \frac{1}{m} \sum_{i=1}^n \left[y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log (1 - h(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2 \quad (9)$$

4.2 Random Forest

In the random forest classifier, each decision tree in the ensemble is built from a sample drawn with replacement from the training set. The decision tree is defined as follows: Let the data at node m be represented by D . At each node m , the split is parameterized by $(j; t_m)$ where j is the feature and t_m is the threshold that partitions the data into $D_{left}(j; t_m)$ and $D_{right}(j; t_m)$ subsets. The mathematical expressions are given by:

$$D_{left}(j; t_m) = \{ (x; y) \mid x_j \leq t_m \} \quad (10)$$

$$D_{right}(j; t_m) = D \setminus D_{left}(j; t_m) \quad (11)$$

The impurity at node m is computed using the Gini impurity function, H , which is defined as:

$$H(X_m) = - \sum_k p_{mk} \log(p_{mk}) \quad (12)$$

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k) \quad (13)$$

The goal is to minimize the impurity $G(Q; \theta)$, defined as:

$$G(Q; \theta) = \frac{n_{\text{left}}}{N_m} H(D_{\text{left}}(\theta)) + \frac{n_{\text{right}}}{N_m} H(D_{\text{right}}(\theta)) \quad (14)$$

by optimizing the parameter, θ , that minimizes the above equation (14).

$$\theta = \operatorname{argmin}_{\theta} G(Q; \theta) \quad (15)$$

Each decision tree usually exhibits high variance and overfits on the training data. In order to reduce the variance, the random forest combines a diverse number of trees and takes an average of the predictions. This process cancels out some errors hence yielding a better model.

5 Analysis/Experiments/Results/Discussion

5.1 Analysis

According to IMF data (6), GDP growth across advanced and developing economies has trend synchronously during past major events. This can be illustrated with the GDP dipping during the 'Lehman Shock' in 2008, or the GDP rising during the world's economic recovery in 2010. One of the underlying reasons behind this contemporary phenomenon can be attributed to countries' dependence on each other for trade. Consequently, the effects of growth or downturn are not isolated but shared across countries. As such, the first objective to this paper is to investigate if the degree of trade between countries plays a cardinal role in its influence to a country's stock market. Based on World Bank's data (7), illustrated in Figure 4, it has shown that the USA is Japan's number one or two import partner in the past decade or so. Given this strong trading relationship, in the event that USA experiences an economic downturn, it's more likely than not that Japan's economy will be adversely affected collaterally. One important assumption that this paper makes is that the economic activity of a country is reflected by its stock market price index. For example, a rising stock market price index will be reflective of an upbeat economy. Likewise, a downward trending market indicates a weakening economy. Hence, the second objective is to utilize this information to predict Japan's stock market price index, specifically, TOPIX, using machine learning techniques.

Figure 4: Top 20 countries that Japan exports to and the coverage to each of these countries

Figure 4 shows the percentage of imports by each of these countries which in total constitutes approximately 78% to 80% of Japan's entire exports. Figure 5 showcase the ranking of each import partner across the years, and the ranking having an inverse relationship to trading importance. A lower rank indicates that a country has a higher trading (import) volume with Japan. In this paper, a subset of 13 member countries (constituting to approximately 70% of Japan's exports) were eventually selected from this list, having fulfilled the following criteria:

1. It must be ranked among the top 15 import partners

2. It must consistently be grouped within the top 15 import partners for at least 12 out of 15 years

The reason for this selection process is to identify countries that are significant enough to influence Japan's economy, and also be consistent in this relationship (not a one-off event).

Figure 5: Ranking of import partner from 2005-2018

(a) Countries' indices (b) Pearson Corr. Coef. (c) Spearman Corr. Coef.

Figure 6: Relationship between TOPIX and other indices

In evaluating the degree of correlation between the closing price of TOPIX and other indices, two test static methods were deployed, namely, Pearson and Spearman correlation coefficient. Pearson correlation coefficient measures the linear relationship between two datasets. On the other hand, Spearman correlation coefficient measures the monotonicity of the relationship between two datasets. As observed in Figure 6, both correlation coefficients yields very similar results, indicating signs

of an overt association between TOPIX and other indices. In the computation of the correlation coefficient for both cases, all indices except TOPIX were lagged by one (1) day to observe their predictive usefulness towards TOPIX. In Figure 6, it shows that the correlation score of STI and TOPIX is smaller compared to the correlation with other indices in both cases. The reason for the low correlation score is due to the decoupling of the closing prices between 2010 to 2012. However, after 2012, the closing prices of STI is closely tracking TOPIX as seen in Figure 7b.

(a) S&P 500 Vs. TOPIX

(b) STI Vs. TOPIX

Figure 7: Closing Prices of S&P500 and STI tracking TOPIX

5.2 Experiments

A total of three (3) machine learning algorithms, namely, Logistic Regression, Random Forest and Support Vector Machine were implemented with their hyperparameters optimized. The objective of these experiments is to determine which model is best suited to predict TOPIX. This section will provide details on the hyperparameters used for each model and the model's performance. The primary metrics used to evaluate the performance of each model are accuracy, precision and recall.

5.2.1 Logistic Regression

In the logistic regression algorithm, the following hyperparameters in Figure 8 were optimized accordingly to achieve high accuracy. The hyperparameters are defined as:

1. C: The value determines the inverse of the regularization strength. The smaller the value, the stronger the regularization.
2. Penalty: Regularization method such as L1 and L2. L1 adds the "absolute value of magnitude" of coefficient as penalty term to the loss function. L2 adds "squared magnitude" of coefficient as penalty term to the loss function.
3. Solver: Algorithms used in the optimization process. Newton's Method which uses quadratic approximation to minimize the cost function. Liblinear uses coordinate descent algorithm to minimize the cost function. LBFGS is a quasi-Newton method used to minimize the cost function.

Figure 8: Hyperparameters for Logistic Regression

In order to optimize the model, a tuning algorithm, Grid Search, together with KFold Cross Validation = 10 were deployed to compute the optimum values of the the hyperparameters. The accuracy of each model given the hyperparameters were computed and ranked as shown in Figure 9.

Figure 9: Optimization of hyperparameters

Based on Figure 9, the best hyperparameters selected are:

1. C: 100
2. Penalty: L2
3. Solver: Newton-cg

These hyperparameters were provided to the model which is then fitted on 80% of the dataset and tested on the remaining 20%. The model's performance is illustrated in the classification report as shown in Figure 10. The model displays a high precision of 64% for the 'Buy' label and moderate levels of 54% for the 'Sell' label. In terms of recall, it trends at a low 45% in case of the 'Buy' label, but exhibits a high recall rate of 73% for the 'Sell' label. Generally, the precision metric is more important than recall because a low precision model will output a wrong trading signal which leads to a guaranteed loss to the investment. On the other hand, recall plays a role in identifying opportunities for trade to occur. The higher the recall, the model has a higher chance to identify a potential winning trade. While having a low recall will result in missed opportunities to trade, executing a bad trade is more detrimental to the portfolio in the long run.

Figure 10: Classification Report

Considering that the model has an accuracy of approximately 58% on the test data ($n=390$), the 95% confidence interval ($z = 1.96$) is calculated as follows:

$$z \cdot \sqrt{\frac{\text{Accuracy} \cdot (1 - \text{Accuracy})}{n}} \quad (16)$$

Based on equation (16), the radius of the 95% confidence interval is given by 4.884%. Hence, the true classification accuracy of the model is likely to be between the 52.88% and 62.64%.

5.2.2 Random Forest

In the random forest algorithm, the following hyperparameters in Figure 11 were optimized accordingly to achieve high accuracy. The hyperparameters are defined as:

1. Max Depth: The value that determines maximum depth of the tree. If not specified, the nodes will be expanded until all leaves are pure or have met the split threshold.
2. Max Features: The maximum number of features taken into consideration when executing the best split. 'Sqrt' indicates that the number of features considered is equivalent to the square root of its total number of features. 'Log2' indicates that the number of features considered is equivalent to log2 of its total number of features. 'None' indicates that all features are considered.
3. Estimators: The value that determines the number of trees in the forest.

Figure 11: Hyperparameters for Random Forest

In order to optimize the model, a tuning algorithm, Grid Search, together with KFold Cross Validation = 10 were deployed to compute the optimum values of the hyperparameters. The accuracy of each model given the hyperparameters were computed and ranked as shown in Figure 12.

Figure 12: Optimization of hyperparameters for Random Forest

Based on Figure 12, the best hyperparameters selected are:

1. Max Depth: 3
2. Max Features: 'Sqrt'
3. Number of Estimators: 100

These hyperparameters were provided to the model which is then fitted on 80% of the dataset and tested on the remaining 20%. The model's performance is illustrated in the classification report as shown in Figure 13. The model displays a moderate precision of 57% for the 'Buy' label and 53% for the 'Sell' label. In terms of recall, it trends at 58% in case of the 'Buy' label, and exhibits a similar recall rate of 54% for the 'Sell' label. The model has an overall accuracy of approximately 55.5% on the test data (n=390). Using equation (16), the radius of the 95% confidence interval is given by 4.914%. Hence, the true classification accuracy of the model is likely to be between the 50.56% and 60.38%.

Figure 13: Classification Report for Random Forest

5.2.3 Support Vector Machine

In the Support Vector Machine algorithm, the following hyperparameters in Figure 14 were optimized accordingly to achieve high accuracy. The hyperparameters are defined as:

1. C: The value determines the inverse of the regularization strength. The smaller the value, the stronger the regularization.
2. Gamma: The value determines the degree of fitting the hyperplane along the dataset. The higher the value, the degree of fitting increases. 'auto' will compute a value equivalent to $\frac{1}{n_features}$
3. Kernel: It defines the type of hyperplane used to partition the data. 'rbf' and 'poly' provides a non-linear hyperplane, and 'linear' provides a linear hyperplane.

Figure 14: Hyperparameters for SVM

In order to optimize the model, a tuning algorithm, Grid Search, together with KFold Cross Validation = 10 were deployed to compute the optimum values of the the hyperparameters. The accuracy of each model given the hyperparameters were computed and ranked as shown in Figure 15.

Figure 15: Optimization of hyperparameters for SVM

Based on Figure 15, the best hyperparameters selected are:

1. Max Depth: 50
2. Max Features: 'auto'
3. Number of Estimators: 'poly'

These hyperparameters were provided to the model which is then fitted on 80% of the dataset and tested on the remaining 20%. The model's performance is illustrated in the classification report as shown in Figure 16. The model displays a moderate precision of 53% for the 'Buy' label and low precision of 49% for the 'Sell' label. In terms of recall, it trends at 72% in case of the 'Buy' label, but exhibits a very low recall rate of 30% for the 'Sell' label. The model has an overall accuracy of approximately 51.9% on the test data (n=391). Using equation (16), the radius of the 95% confidence interval is given by 4.914%. Hence, the true classification accuracy of the model is likely to be between the 46.97% and 56.85%. Based on all three (3) model's performance, SVM is the worst performing algorithm given its low scores.

Figure 16: Classification Report for SVM

5.3 Results & Discussion

The general strategy to creating 'good' models is to tune the hyperparameters in order to optimize the objective function. However, this approach may not necessarily translate into 'good' financial performance. For example, a model with a prediction accuracy of 60% may generate a lower profit compared to a model with a prediction accuracy hovering just above 50%. One possible reason may be attributed to the model's sensitivity towards high risk situation. Consequently, the model may take losing positions that leads to high loss. In this section, it will introduce various metrics used to evaluate the model's financial performance, namely, returns, Sharpe Ratio and other metrics. In the conclusion of this analysis, the logistic regression model is determined to be the best suited model to predict TOPIX.

5.3.1 Returns

Among the three models, the logistic regression model yields the best cumulative returns as shown in Figure 20. Interestingly, the Random Forest model which has an overall accuracy of approximately 55%, compared to the logistic regression model with an overall accuracy of approximately 58%, yield a -32% difference in cumulative profit at the end of the test period. In Figure 17, 18, & 19, it shows the breakdown of the returns across the months and years. A general overview of these charts shows two important attributes of the best performing model, that is, consistency and positive returns. The left most chart for each model exhibits the months where the returns are positive (Pale yellow to green) and the months where the returns are negative (orange-yellow to red). For a model to illustrate both consistency and positive returns, the chart should reflect an almost single tone of light green to dark green. These characteristics is dominant in the logistic regression model, less so in the other two models.

The middle chart for each model exhibits the annual returns. An overview of these charts indicates a consistent annual positive returns for both the logistic regression and random forest models. However, a micro view of the year 2019, shows the cyclical returns of the random forest model, reinforcing the notion of its unreliability. The third chart on the right shows the distribution of the monthly returns. Naturally, a good performing model will have its returns skewed to the right where returns are positive. The logistic regression model again shows its relatively strong skewness towards the right and is less obvious for the random forest and SVM models. Based on the analysis of the returns, the logistic regression model is the best performing model.

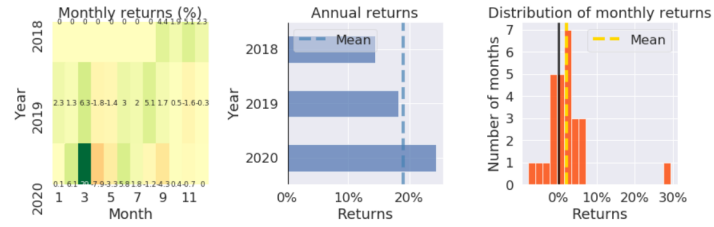


Figure 17: Monthly and Annual Returns of Logistic Regression Model

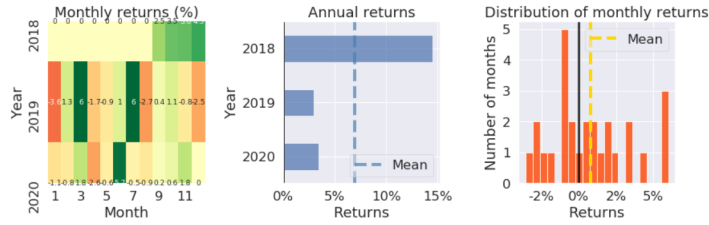


Figure 18: Monthly and Annual Returns of Random Forest Model

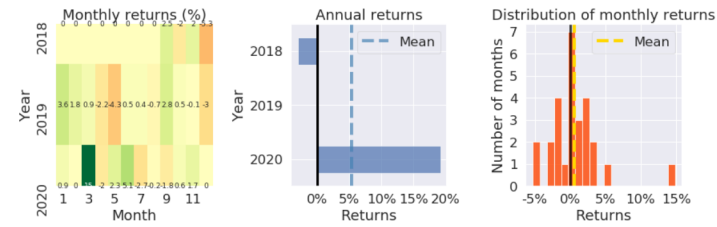


Figure 19: Monthly and Annual Returns of SVM Model

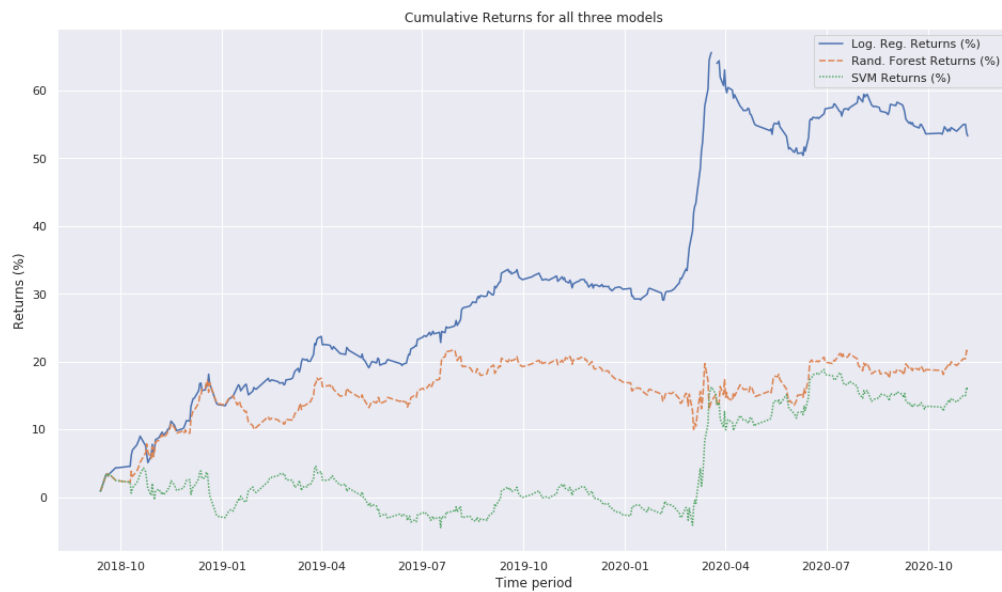


Figure 20: Cumulative Returns across 2 years

5.3.2 Rolling Sharpe Ratio

The Sharpe ratio evaluates the performance of the portfolio (model's prediction) compared to a risk-free asset. It indicates the additional amount of return received per unit increase in risk. Therefore, a higher Sharpe Ratio is reflective of higher returns and lower risk. In the analysis of the model's performance, the rolling Sharpe Ratio was deployed so as to track the Sharpe Ratio of the most recent 180 days. Based on Figure 21,22, & 23, the logistic regression model exhibits the most stable and the largest Sharpe Ratio among the three models. For the Random forest and SVM model, in some time periods, the risk appears higher than the returns, resulting in a negative Sharpe Ratio.

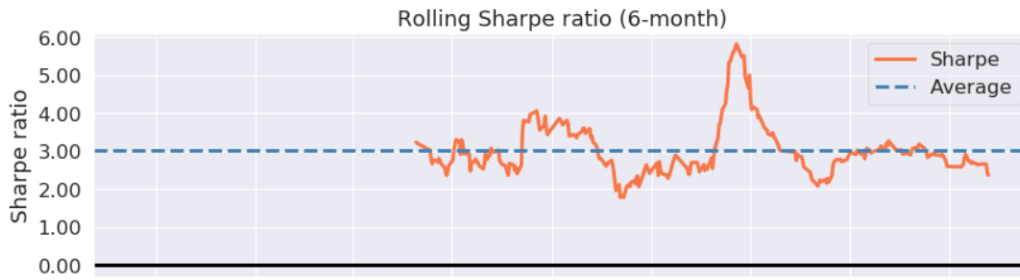


Figure 21: Rolling Sharpe Ratio of Logistic Regression Model Returns

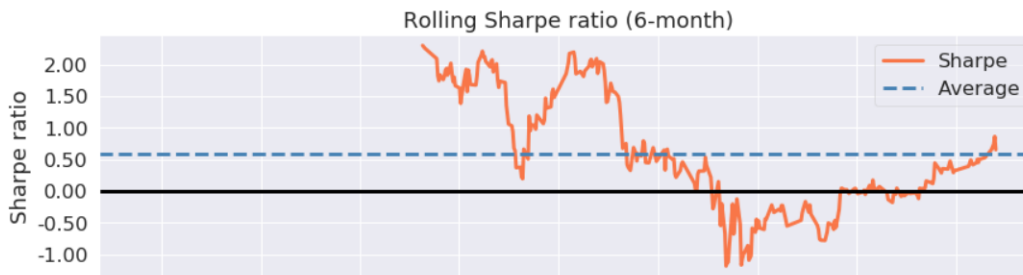


Figure 22: Rolling Sharpe Ratio of Random Forest Returns

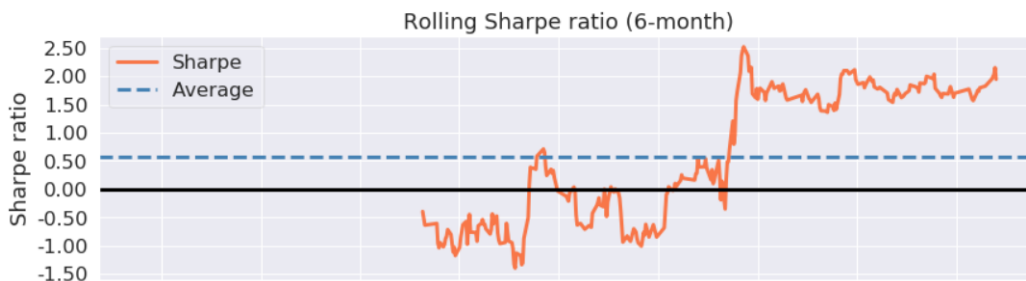


Figure 23: Rolling Sharpe Ratio of SVM Returns

5.3.3 Other Evaluation Metrics

There are other metrics used to evaluate the models such as daily value-at-risk, the tail ratio and etc. The general overview of the results in Figure 24 indicates that the logistic regression model is the best performing model.

	Backtest		Backtest		Backtest
Annual return	39.9%	Annual return	13.6%	Annual return	9.7%
Cumulative returns	68.1%	Cumulative returns	21.9%	Cumulative returns	15.5%
Annual volatility	13.0%	Annual volatility	13.2%	Annual volatility	13.2%
Sharpe ratio	2.64	Sharpe ratio	1.03	Sharpe ratio	0.77
Calmar ratio	2.79	Calmar ratio	1.21	Calmar ratio	1.11
Stability	0.90	Stability	0.34	Stability	0.44
Max drawdown	-14.3%	Max drawdown	-11.3%	Max drawdown	-8.7%
Omega ratio	1.62	Omega ratio	1.21	Omega ratio	1.15
Sortino ratio	4.65	Sortino ratio	1.52	Sortino ratio	1.19
Skew	0.72	Skew	-0.20	Skew	0.54
Kurtosis	3.58	Kurtosis	3.89	Kurtosis	3.72
Tail ratio	1.49	Tail ratio	1.15	Tail ratio	1.12
Daily value at risk	-1.5%	Daily value at risk	-1.6%	Daily value at risk	-1.6%

(a) Logistic Regression (b) Random Forest (c) Support Vector Machine

Figure 24: Various Performance Metrics

6 Conclusion/Future Work

The main objective of this paper was to investigate if the trade relations between Japan and other countries could affect her economy. The second objective was to determine if the data associated with this information can be used to build a model and to predict TOPIX. In summary, both objectives were met and supported with evidence. First, the correlation coefficient displayed high scores for most of these countries (Top 15 trading partners), indicating a strong linear and monotonic relationship. Next, the preprocessed data were then used to train and test the models. Although the accuracy is not 'high', the consistency and the ability to rake in positive returns were obvious for the logistic regression model, indicating some forecasting abilities. Based on these results, we can conclude that an upbeat or a downswing in the economy of these countries would have an impact on Japan's stock market price index, specifically for TOPIX. One interesting observation from this research is the poor performance of the Random Forest and SVM model compared to the logistic regression model. Intuitively, one would expect a non-linear relationship between the multiple indices and TOPIX. However, the results show otherwise despite optimally tuning the main hyperparameters. There are several reasons that could be attributed to this outcome such as the size of the dataset which may not be sufficient to train a good Random Forest or SVM model, other hyperparameters besides the main ones has to be taken into consideration, and more types of features should be engineered in order to draw out the underlying characteristics of the data.

References

- [1] Shigeo Kamitsuji and Ritei Shibata *Effectiveness of Stochastic Neural Network for Prediction of Fall or Rise of TOPIX* Asia-Pacific Financial Markets, 2003, vol. 10, issue 2, 187-204
- [2] Lukas Ryll Sebastian Seidens, 2019. *Evaluating the Performance of Machine Learning Algorithms in Financial Market Forecasting: A Comprehensive Survey* Papers 1906.07786, arXiv.org, revised Jul 2019.
- [3] Chih-Fong Tsai, Yueh-Chiao Lin, David C. Yen, Yan-Min Chen *Predicting stock returns by classifier ensembles* Applied Soft Computing, Volume 11, Issue 2, 2011
- [4] J. Gong and S. Sun *A New Approach of Stock Price Prediction Based on Logistic Regression Model* 2009 International Conference on New Trends in Information and Service Science, Beijing, 2009
- [5] Shen, Shunrong and Jiang, Haomiao and Zhang, Tongda *Stock market forecasting using machine learning algorithms* Department of Electrical Engineering, Stanford University, Stanford, CA, 2012
- [6] IMF Real GDP Growth
https://www.imf.org/external/datamapper/NGDP_RPCH@WEO/OEMDC/ADVEC/WEOWORLD
- [7] World Bank Trade Data
<https://wits.worldbank.org/CountryProfile/en/JPN>