**Abstract**

In this report, we present a machine learning based approach for predicting next day movements in the S&P 500 index by classifying them into five distinct categories: 1) Large downward movement (more than 1% decline), 2) Normal downward movement (more than 0.3% but less than 1% decline), 3) Flat (less than 0.3% movement), 4) Normal upward movement (more than 0.3% but less than 1% increase), and 5) Large upward movement (more than 1% increase). By leveraging an extensive set of technical indicators, including opening price, intra-day high, intra-day low, closing price, Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), consecutive upward/downward count, and stochastic oscillators, our model aims to provide accurate predictions of market movement in upcoming day.

The report details the methodology used for preprocessing the historical data, feature engineering, and model selection. Additionally, we discuss the rationale behind choosing a suitable machine learning algorithm, taking into account its ability to handle non-linear relationships and potential overfitting.

Through a comprehensive evaluation, we compare the accuracy and robustness of our model with other established prediction techniques. The results demonstrate the efficacy of our proposed approach in capturing the nuances of the S&P 500 index and its potential to aid investors and traders in making informed decisions. Furthermore, the report proposes avenues for future research to enhance the model's predictive capabilities and suggests a plausible option strategy based on the result of the model..

**1. Introduction**

Option trading offers traders the opportunity to optimize their investment strategies by leveraging financial instruments that can be highly beneficial when used correctly but entail significant risks when misused. In this paper, we propose a machine learning-based approach to assist traders in making informed decisions about employing options in their strategies by predicting the direction and range of the S&P 500 index's movement. We focus on three specific scenarios: flat days, normal days, and large movement days, which correspond to different degrees of price changes in the S&P 500 index.

1.1 Classification of S&P 500 Movement Scenarios

To determine the suitability of using options in various market situations, we define the following classes:

A) Flat day (less than ±0.3% change): Using options is not recommended due to the low potential for significant portfolio value changes and the inherent risk involved in options trading.

B) Normal days (between ±0.3% and ±1.0% change): Options trading becomes a viable consideration for maximizing profit or hedging one's portfolio.

C) Large movement days (greater than ±1.0% change): The market exhibits a strong directional move, making options trading highly beneficial for either hedging or profit maximization.

1.2 Objectives and Scope

The primary goal of this study is to develop a predictive model that accurately forecasts the direction and range of the S&P 500 index's movement. By doing so, traders can optimize their option trading strategies, mitigating risk and maximizing potential returns. Furthermore, we aim to discuss the implications of these predictions on the decision-making process regarding the use of options in investment strategies.

In the subsequent sections of this paper, we will present the methodology used to develop the predictive model, including data preprocessing, feature engineering, and model selection. Additionally, we will discuss the evaluation of the model's performance and compare it with other established prediction techniques. Finally, we will propose avenues for future research to enhance the model's predictive capabilities and suggest an option strategy that could be used based on the model’s outcome.

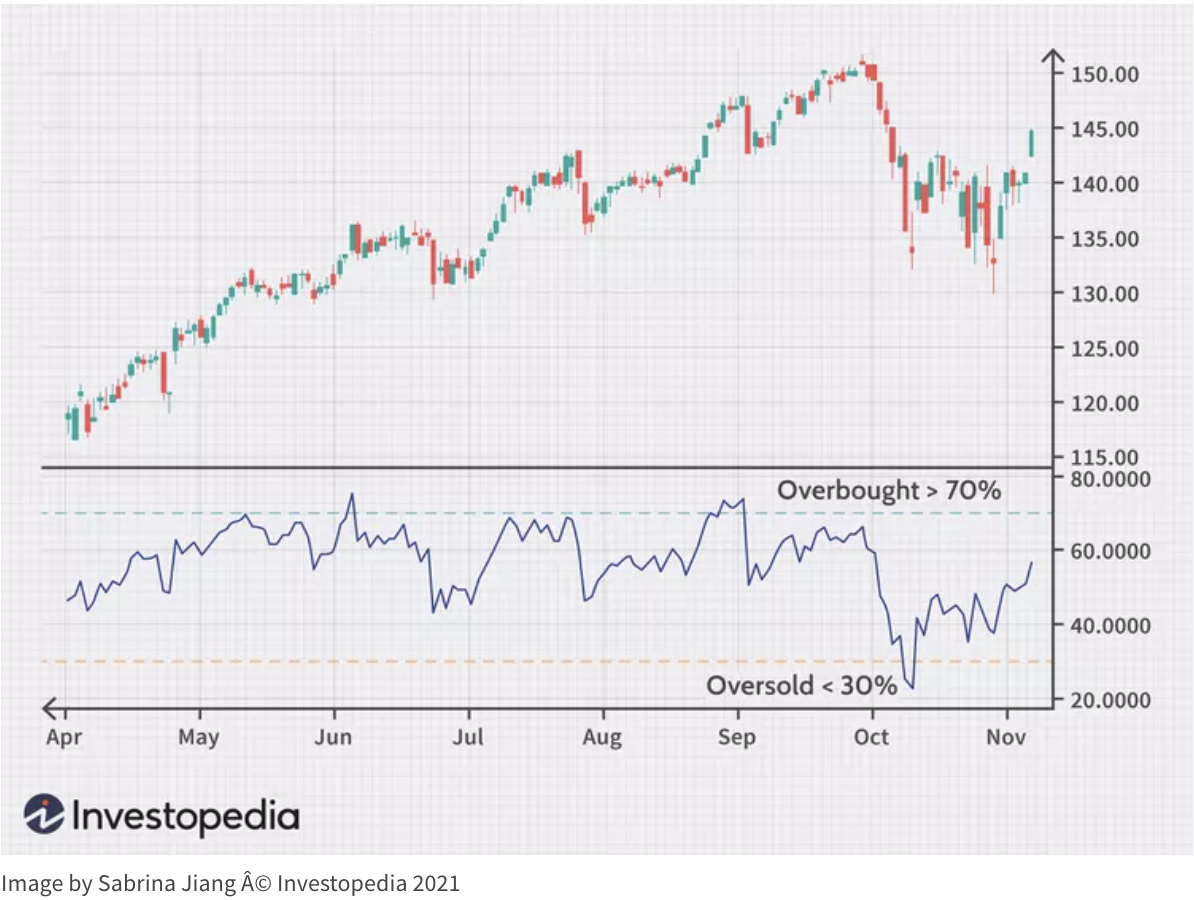
**2. Data Preparation and Feature Engineering**

2.1 Data Sources and Collection

For this machine learning project, we used a variety of technical indicators, such as RSI, MACD, and stochastic, to predict the S&P 500 index movement. Although these indicators are readily available online, they could not be directly downloaded in a format suitable for data processing, such as an Excel file. Consequently, we opted to compute these indicators manually using the historical price data obtained from Yahoo Finance that includes only opening price, closing price, intra-day high, and intra-day low.

2.2 Calculating Technical Indicators

2.2.1 Relative Strength Index

The Relative Strength Index (RSI) is a momentum indicator that measures the speed and change of price movements. The RSI oscillates between 0 and 100 and is typically used to identify overbought or oversold conditions in a market. The calculation for RSI is as follows:

1) Calculate the average gain and average loss for a given period (usually 14 days).

2) Compute the Relative Strength (RS) by dividing the average gain by the average loss: RS = Average Gain / Average Loss.

3) Calculate the RSI using the following formula: RSI = 100 - (100 / (1 + RS)).

2.2.2 Moving Average Convergence Divergence

The Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. The MACD is calculated by subtracting the 26-day Exponential Moving Average (EMA) from the 12-day EMA:

1) Compute the 12-day EMA of the closing prices.

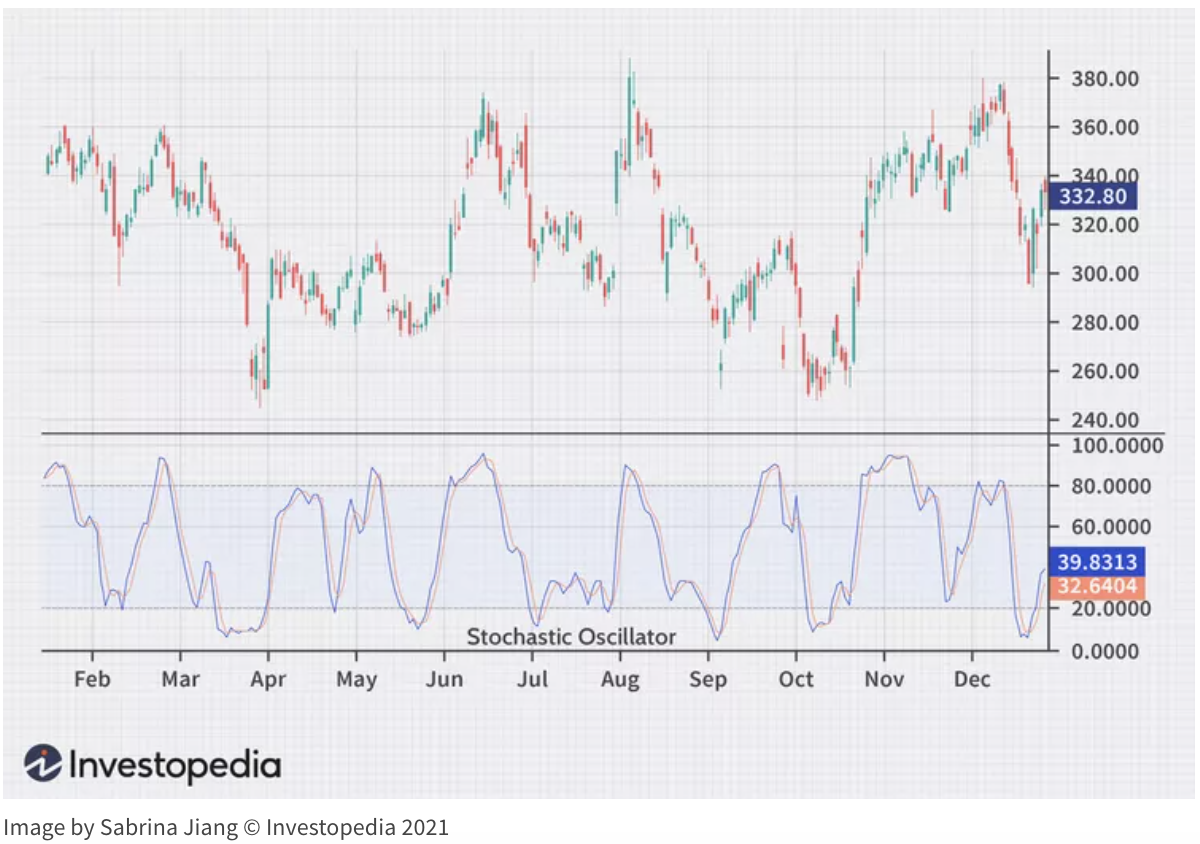
2) Compute the 26-day EMA of the closing prices.

3) Subtract the 26-day EMA from the 12-day EMA to obtain the MACD line.

4) Calculate the 9-day EMA of the MACD line, known as the "signal line."

5) Plot the MACD line and the signal line to observe potential buy and sell signals.

2.2.3 Stochastic Oscillator



The stochastic oscillator is a momentum indicator that compares a particular closing price of a security to a range of its prices over a certain period. It is used to generate overbought and oversold trading signals, utilizing a scale from 0 to 100. The calculation for the stochastic oscillator involves the following steps:

1) Determine the highest high and the lowest low over a given lookback period (typically 14 days).

2) Calculate the %K line using the following formula: %K = [(Closing Price - Lowest Low) / (Highest High - Lowest Low)] x 100.

3) Compute the %D line, which is a 3-day simple moving average of the %K line.

Once the RSI, MACD, and stochastic indicators are calculated, they can be incorporated into the data set as additional features to train the machine learning model.

2.3 Incorporating Additional Features

In addition to the technical indicators RSI, MACD, and stochastic, our project incorporated several other features to enhance the machine learning model's predictive capabilities. These features include days of consecutive movement, total percentage change during consecutive movements, federal interest rates, and each day's classification. Incorporating a diverse set of features allows the model to capture a broader range of market behaviors and trends, improving its ability to make accurate predictions.

2.3.1 Days of Consecutive Movement and Total Percentage Change

To provide our model with information about the recent market trends, we calculated the number of consecutive days the market moved either upward or downward. This feature captures short-term momentum and helps identify potential trend reversals or continuations. In addition to the number of consecutive days, we also computed the total percentage change during these consecutive movements. This additional feature helps quantify the magnitude of the market moves and can be indicative of the strength of the underlying trend.

2.3.2 Federal Interest Rates

Federal interest rates play a crucial role in influencing stock market performance, as they impact borrowing costs and the availability of credit. To account for this economic factor, we included the federal interest rate as a feature in our dataset. To obtain the historical federal interest rate data, we examined the Federal Reserve's policy changes and manually input the corresponding rates into our dataset. By incorporating the federal interest rate, our model can better understand the relationship between macroeconomic variables and stock market movements, potentially improving its predictive accuracy.

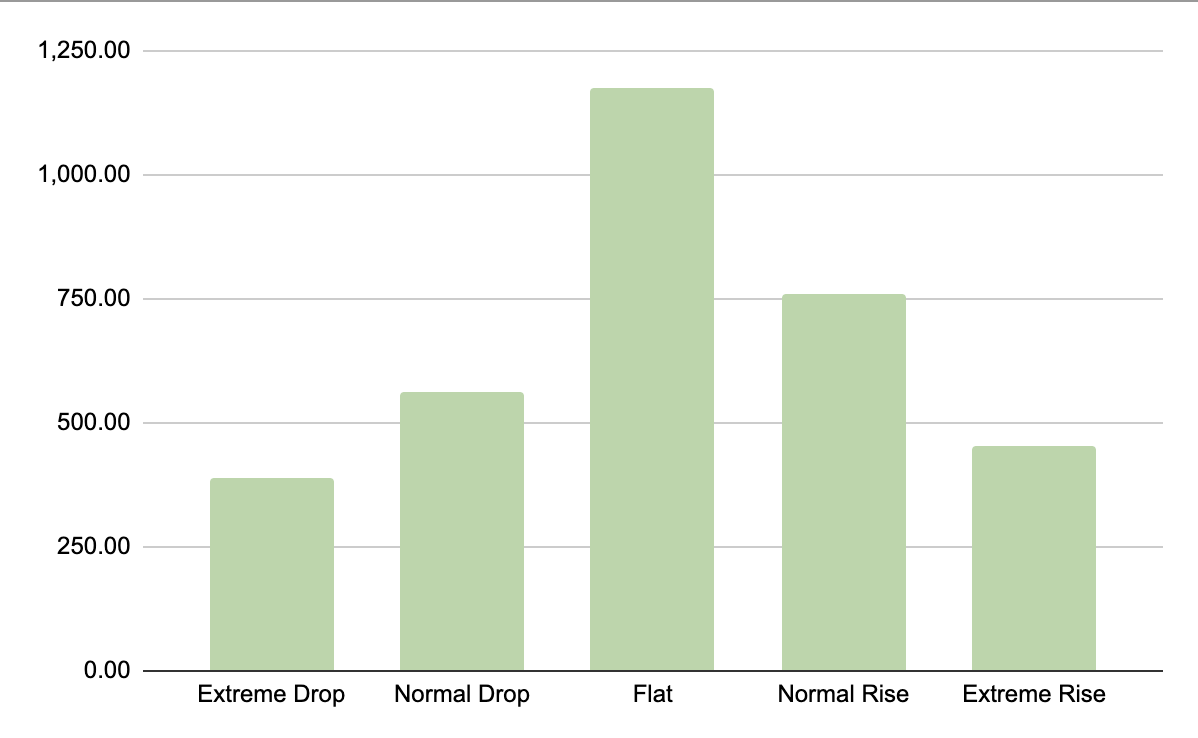
2.3.3 Daily Classification

Finally, we assigned a classification label to each day based on the closing price relative to the previous day's closing price. The classification categories include flat days, normal up/down days, and extreme up/down days, as defined in section 1.1. This daily classification serves as the target variable for our machine learning model, enabling the model to learn and predict the direction and range of S&P 500 index movements.

By integrating these additional features into our dataset, we aim to create a more comprehensive and informative representation of the stock market dynamics. This enriched dataset is expected to improve the performance of our classification based machine learning model, assisting traders in making informed decisions about option trading strategies.

2.4 Addressing Imbalanced Data Distribution

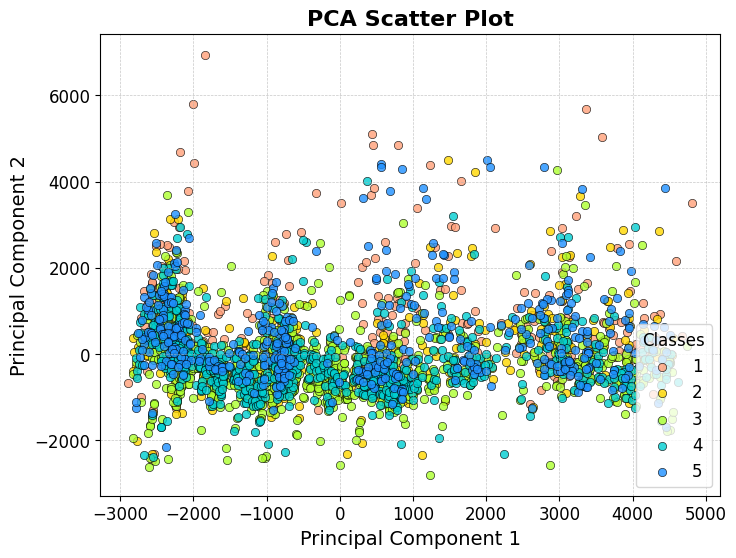
Upon analyzing the S&P 500 historical data from 2010 to the present, we observed an uneven distribution of instances across the five classifications as shown in the below chart. The majority of the instances were classified as flat days, followed by normal up and down days, and the fewest instances were attributed to extreme up and down days. This imbalance in class distribution can lead to issues in machine learning projects, as it may cause the model to be biased towards the majority class, reducing its ability to accurately predict minority classes. This is because the learning algorithm may not receive enough training examples from the minority classes to learn their distinguishing characteristics effectively. Consequently, the model may be biased towards the majority class, producing predictions that are less reliable for the minority classes.



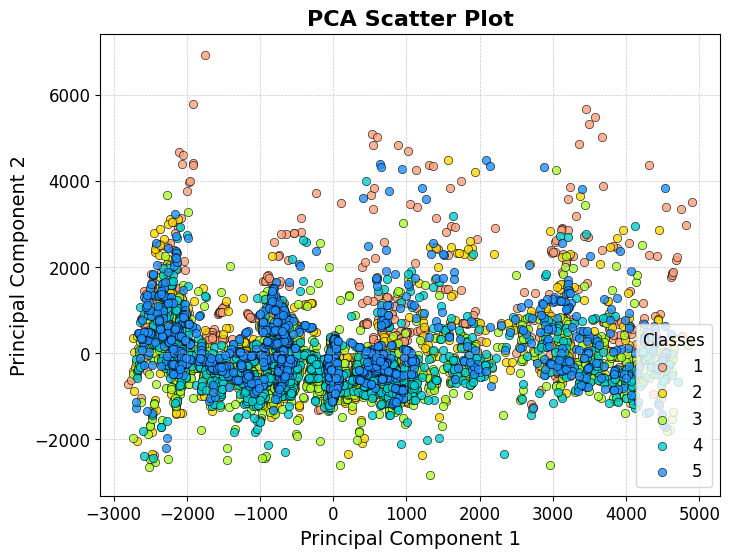
To address the issue of imbalanced data, we employed the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE is an oversampling method that generates synthetic instances for the minority classes to balance the class distribution. The algorithm achieves this by selecting instances that are close in the feature space, drawing a line between the instances, and generating synthetic instances along the line. By creating these synthetic instances, SMOTE allows the learning algorithm to obtain more information about the minority classes, improving its ability to generalize and predict instances from all classes accurately.

In our project, SMOTE oversampled the extreme up and down days, and to a lesser extent, normal up and down days. By increasing the representation of these classes in our dataset, we aimed to enhance our model's ability to predict all classes more accurately, thus mitigating the bias towards the majority class. This balanced class distribution is expected to result in a more robust and reliable prediction model for S&P 500 index movements, ultimately assisting traders in making informed decisions regarding option trading strategies.

PCA Scatter Plot before normalizing looks like below.



Once we normalize the data, the scatter plot looks like below. When examined carefully, it is visible that there are clearly more numbers of class 1 and class 5 plots than shown in the previous plot. SMOTE balanced out the number of data points.



2.5 Weighting Based on Dates

Considering the time-series nature of our dataset, it is essential to give more weight to the most recent data points, as they are likely to have a stronger influence on predicting the next day's S&P 500 index movement. To achieve this, we first extract the year from the Date feature in our dataset. We then normalize this feature by mapping its values between zero and one, ensuring that the most recent data points receive higher weights. These weights are then applied to each data row in our dataset during the training process, enabling our classifiers to prioritize learning from more recent data points.

**3. Classification Algorithm**

3.1 Support Vector Machine

Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification and regression tasks. The primary goal of SVM is to find the optimal hyperplane that separates the classes with the largest margin possible. The margin is defined as the distance between the hyperplane and the closest data points, known as support vectors. The optimization problem in SVM aims to maximize this margin while minimizing classification errors. SVM can also be applied to non-linearly separable data using kernel functions, which transform the data into a higher-dimensional space where a linear hyperplane can effectively separate the classes.

For our project, we use a Support Vector Machine implementation with a Radial basis function (RBF) kernel. The RBF kernel is a popular choice for its effectiveness in handling complex and non-linear data. It is defined as:

where x and y are data points, ||x - y|| is the Euclidean distance between the two points, and γ is a positive parameter that controls the shape of the decision boundary. A larger γ value will produce a more flexible decision boundary, while a smaller value will produce a smoother boundary.

To train the SVM we use the Hinge Loss function, which is expressed as follows:

where is the true label, is the feature vector, w is the weight vector, and b is the bias term.

3.2 Random Forest Classifier

Random Forest Classifier (RFC) is an ensemble learning method that constructs multiple decision trees during the training phase and combines their predictions to produce a more accurate and robust output. By aggregating the results of multiple decision trees, Random Forest can mitigate the risk of overfitting, which is a common issue in single decision tree models. Each tree in the Random Forest is constructed using a random subset of the training data and a random subset of features, leading to a diverse set of trees that capture different patterns within the data. The final classification is determined by majority voting among the individual tree predictions.

RFC typically employs the Gini Impurity or the Entropy as a splitting criterion during the tree construction, which can be considered as surrogate loss functions for minimizing classification errors. For our implementation, we use the default impurity which is the Gini Impurity.

3.3 Feed-Forward Neural Network

Feed-Forward Neural Network (FNN)) is a type of neural network that consists of multiple layers of interconnected neurons. FNNs are designed to learn complex patterns within the data by using non-linear activation functions and adjusting the weights of connections between neurons during the training process. FNNs consist of an input layer, one or more hidden layers, and an output layer. The input layer receives the feature values, the hidden layers process the data and learn the underlying patterns, and the output layer produces the final classification probabilities. Here, we use a Neural Network with two hidden layers of size 30 and 20 neurons respectively, each using ReLU activation function and in the output layer has a total of 5 neurons equal to the number of classes and it was trained for 150 epochs

Loss Function: For our project, we use the Cross-Entropy Loss function for classification tasks, which can be expressed as:

where is the true label and is the predicted probability of the true label for data point i.

**4. Results & Discussions**

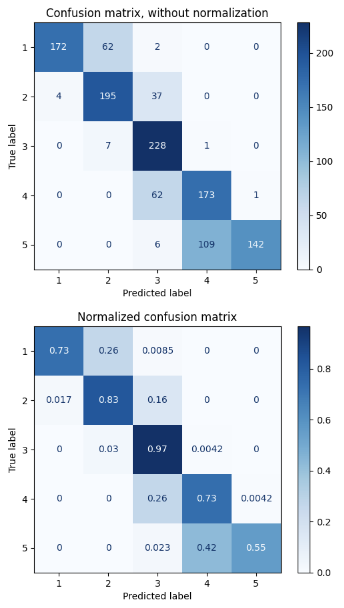
The results of our experiments with the various classifiers show promising outcomes for predicting the S&P 500 index movement using machine learning techniques. Each classifier achieved an accuracy rate of approximately 75%. While there is still room for improvement, this level of accuracy demonstrates the potential of machine learning in predicting stock market movements.

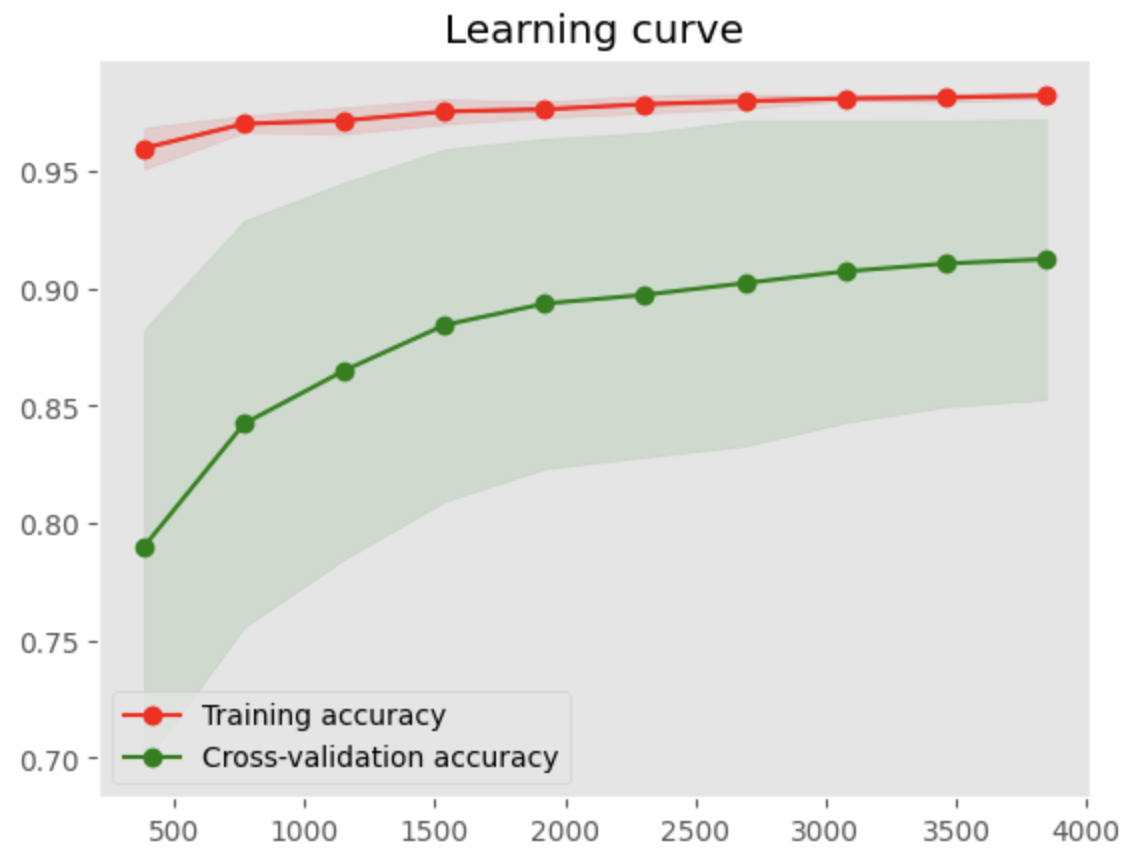
One thing to note is that despite using SMOTE to normalize the distribution of classes, the accuracy for class 3 (flat day) remains the highest. In contrast, the classes with fewer data points exhibit lower accuracy rates. This suggests that further investigation into class balancing techniques may be necessary to improve the model's performance across all classes.

Another interesting point is that our model excels in predicting the market's direction accurately. According to the test results, once the model predicts a normal or extreme movement in one direction, the actual outcome is never in the opposite direction. Although the movement may be less intense or flat, the model's directional prediction remains reliable. The model's accuracy in predicting market direction holds significant potential for profit generation, particularly for traders using options which is further explained in section 5.

4.1 Support Vector Machine

|  | Precision | Recall | F1-score |
| --- | --- | --- | --- |
| Class 1 | 0.98 | 0.73 | 0.83 |
| Class 2 | 0.74 | 0.83 | 0.78 |
| Class 3 | 0.68 | 0.97 | 0.80 |
| Class 4 | 0.61 | 0.73 | 0.67 |
| Class 5 | 0.99 | 0.55 | 0.71 |

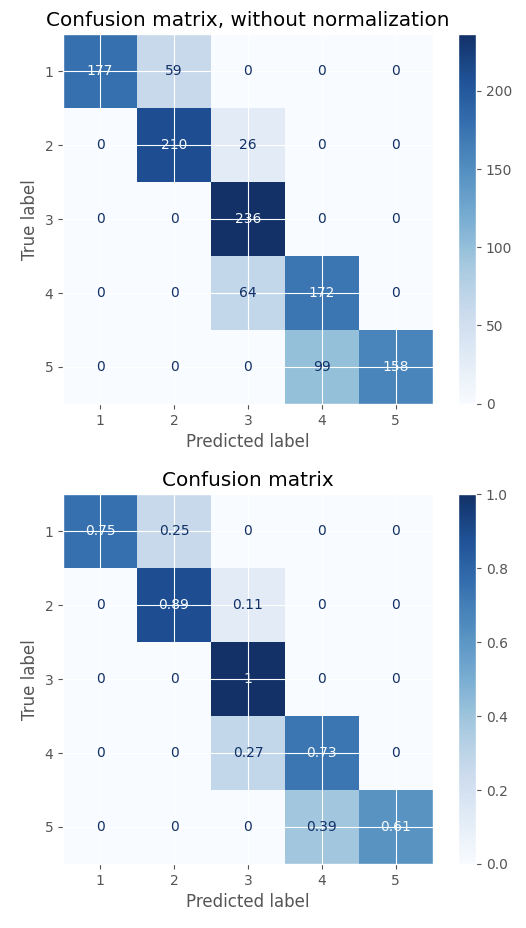


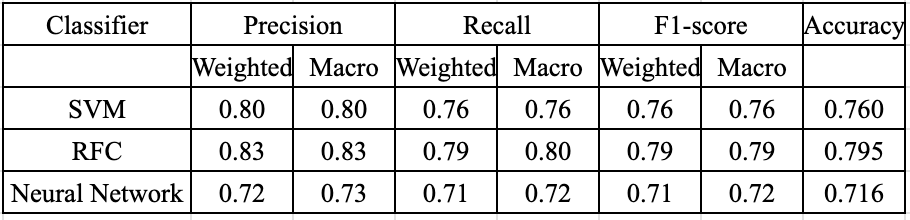


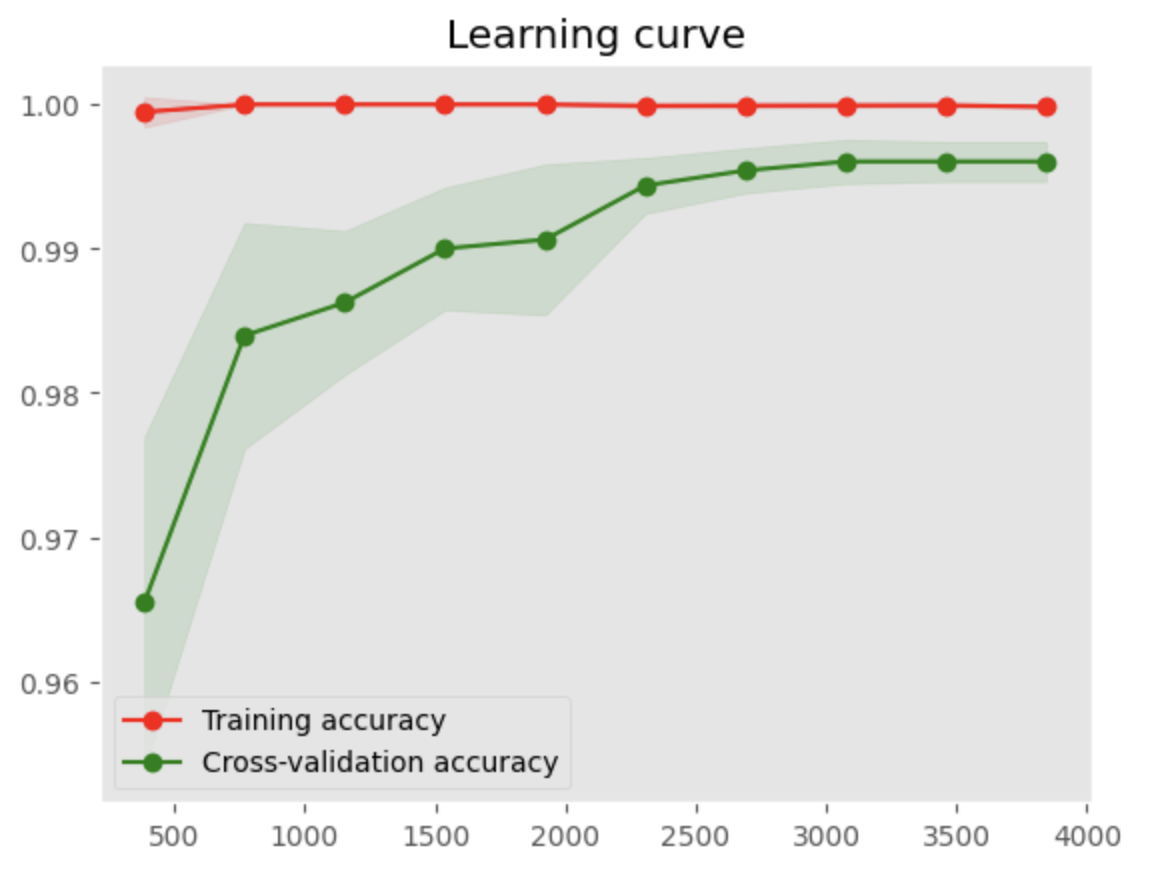
As written previously, the accuracy for class 3 (flat day) came out highest while other classes with fewer samples got lower accuracy even after normalizing with SMOTE. Also, the prediction on direction was very accurate. For example, if the model predicted a class 5 (huge increase) day, then it was right 55% of the time, while 42% came out to be a normal increase. The remaining errors were in the flat day class, but none of the errors were in normal down or huge down day.

4.2 Random Forest Classifier

|  | Precision | Recall | F1-score |
| --- | --- | --- | --- |
| Class 1 | 1.00 | 0.75 | 0.86 |
| Class 2 | 0.78 | 0.89 | 0.83 |
| Class 3 | 0.72 | 1.00 | 0.84 |
| Class 4 | 0.63 | 0.73 | 0.68 |
| Class 5 | 1.00 | 0.61 | 0.76 |



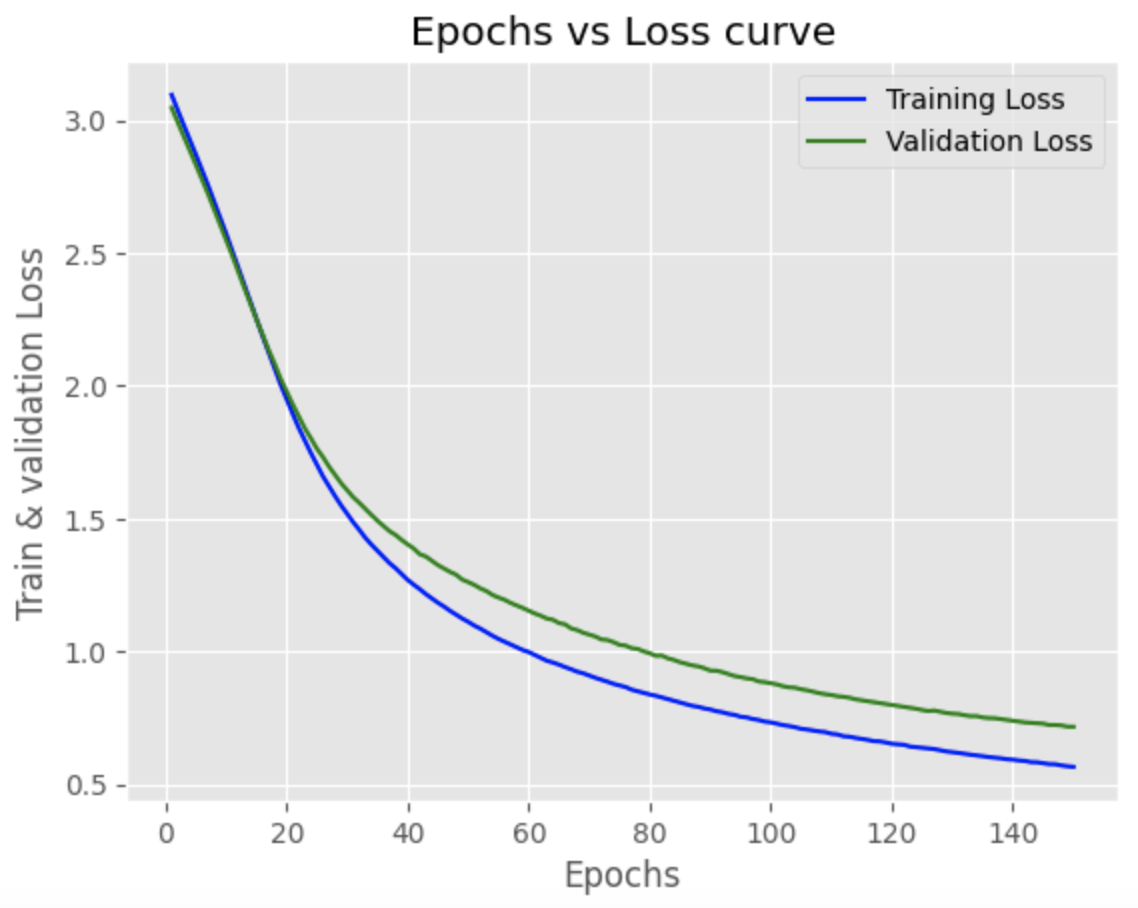
For the Random Forest Classifier, it is evident that it does a better job classifying the Class 0 as compared to SVM. However, as we look at the extreme ends on the diagonal of the confusion matrix for the extreme uo and down days i.e. Class 1 and Class 5 the number of missed detections is relatively high. RFC does a good job in predicting a normal high or low day better than the SVM. Also, there is no overfitting on the dataset since the training accuracy and cross validation accuracy have very low difference after ~3500 epochs. 



4.3 Feed-Forward Neural Network

|  | Precision | Recall | F1-score |
| --- | --- | --- | --- |
| Class 1 | 0.88 | 0.88 | 0.88 |
| Class 2 | 0.65 | 0.78 | 0.72 |
| Class 3 | 0.68 | 0.63 | 0.66 |
| Class 4 | 0.59 | 0.62 | 0.60 |
| Class 5 | 0.84 | 0.71 | 0.76 |

The Feed-Forward Neural Network does a good job in predicting Classes 1, 2, and 5 i.e. It is able to predict the low days much better than the high days i.e. as we move from the center towards the extreme days we get a better accuracy.



The model is not overfitting since on the learning curve the cross-validation and training loss have a very minimal difference which indicates an overall good fit.

4.4 Comparison Table

**5. Possible Option Strategy**

Options are financial instruments that give the holder the right, but not the obligation, to buy or sell an underlying asset at a predetermined price (called the strike price) before a specific expiration date. They can be used to hedge a portfolio, generate income, or speculate on the direction of the market. In this section, we will discuss how to utilize options based on the predictions generated by our machine learning model.

The price of an option is influenced by several factors, including the time value and whether the option is in-the-money or out-of-the-money. Time value represents the portion of an option's price attributed to the time remaining until the option's expiration. Options with more time until expiration have a higher time value, as there is a greater chance that the option will move into a profitable position.

In-the-money options have a strike price that is favorable compared to the current market price of the underlying asset. For call options, this means the strike price is lower than the market price, while for put options, the strike price is higher. Out-of-the-money options, conversely, have a strike price that is unfavorable compared to the current market price.

Option prices can fluctuate significantly based on their time value and whether they are in or out of the money. In general, ITM options have a higher intrinsic value and less time value, while OTM options have a lower intrinsic value and a higher time value.

Based on the predictions from our machine learning model, traders can use options in the following ways. When the model predicts a flat day, do not invest or hold any option position as the time value of the option will simply decay, leading to a loss in the option's value. When the model predicts a normal up or down day, consider buying options with longer expiration dates and in-the-money strike prices. These options will have less price fluctuation, providing a more stable return on investment. When the model predicts a huge up or down day prediction, consider buying options with shorter expiration dates and out-of-the-money strike prices, are more sensitive to price fluctuations in the underlying asset, allowing traders to capitalize on the predicted extreme market movements and potentially achieve substantial returns.

The option trading strategy outlined above is based on the results of our machine learning model, which has demonstrated a strong ability to accurately predict the direction of the market. In most cases, the model's errors stem from inaccuracies in predicting the degree of market movement rather than the direction of the movement itself. By leveraging this key strength of the model, traders can effectively utilize options to capture potential profits while mitigating risk, as they can confidently rely on the model's directional predictions while adapting their strategies to account for varying degrees of market movement.

**6. Conclusion**

Our study found that the best performing model was the Random Forest Classifier, achieving an accuracy of 79.5%, compared to the Support Vector Machine at 76.0% and the Feed-Forward Neural Network at 71.6%. Future research could expand the scope of this study to predict different time frames, such as a week or a month. We are eager to test our model daily on the real market even after the completion of this project, to evaluate its performance in a real-world context. As discussed in section 4, the model excels in accurately predicting the direction of the market movement. If this holds during the test on the real market daily, this model could be of enormous value for the stock market investors.

**Citation**

1. Kenton, W. (2021, August 30). Stochastic Oscillator. Investopedia. https://www.investopedia.com/terms/s/stochasticoscillator.asp
2. Kenton, W. (2021, May 19). Moving Average Convergence Divergence (MACD). Investopedia. https://www.investopedia.com/terms/m/macd.asp
3. Chen, J. (2021, June 25). Relative Strength Index (RSI). Investopedia. https://www.investopedia.com/terms/r/rsi.asp
4. Block, J. (2021, August 31). Federal Funds Rate History: When Has the Fed Raised or Lowered Interest Rates? Forbes Advisor. https://www.forbes.com/advisor/investing/fed-funds-rate-history/
5. Brownlee, J. (2020, July 28). SMOTE for Imbalanced Classification with Python. Machine Learning Mastery. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/
6. Scikit-learn Developers. (n.d.). Support Vector Machines. Scikit-learn. https://scikit-learn.org/stable/modules/svm.html
7. Scikit-learn Developers. (n.d.). sklearn.ensemble.RandomForestClassifier. Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html