Stock Prediction using classification techniques

Jatin Dahiya
19ucs033@lnmiit.ac.in

Ram Ahuja
19ucs017@lnmiit.ac.in

Aditya Khandelwal
19ucs163@lnmiit.ac.in

***Abstract*—This research paper defines the use of classification techniques to determine whether to buy or sell stock. Using technical indicators, we wish to train a model that would help us maximize our stock predictions and in turn maximize our profits. Many approaches have been developed in the field of neural networks but when the data variables are low on the given company classification techniques such as SVM shine in those regions.**

I. INTRODUCTION

We all are captivated by the unpredictability of the stock market and therefore for a person to invest in a company is a very hard choice. A person may see the history of stock of that company but in the end, he or she may not be able to deduce whether to invest in the company or not. So we used classification algorithms to train the data on the technical indicators deduced from the given data and predict whether to buy or sell the shares of the company.
We need a profitable stock trading strategy to maximize our profits. One may use techniques like reinforcement learning to train a model to get us the optimal output, but to get the optimal output we need a lot of variable output such as sentimental analysis, a larger dataset, and training of a Reinforced learning agent which requires a very large time depending on the size of the dataset.

In this project, we have taken the stock of 'APPLE' and our aim is to use classification techniques like SVM, Logistic Regression and KNN that can help us determine whether we should buy or sell a stock. We compare the results of these different classification techniques and find the one that gives us the most optimal output.

Our results indicate that linear SVM and logistic regression give us the optimal output than KNN and rbf SVM.

II. LITERATURE SURVEY

Classification techniques and their use in the prediction of stocks before the use of reinforcement learning have given us good results. SVM has been used efficiently on the data with high dimensional input and is often superior to mainstream algorithms. SVM has many kernels like linear, polygonal and rbf. Choosing these kernels is a big task as polygonal SVM can be compared with logistic regression and rbf SVM can be compared with KNN.
As the data for the stocks generally follow an upward trend it is safe to assume that linear SVM and the logistic regression will result in better performance than then other algorithms.
Given the high time complexity of SVM, we can use the feature scaling to train them faster and to converge the loss function really faster
Reinforcement learning can also be used in stock prediction but given the limited data, it is easy and better to use classification instead of neural networks. Given more complex data about the company and the stocks, neural networks are a great approach than the classification techniques.

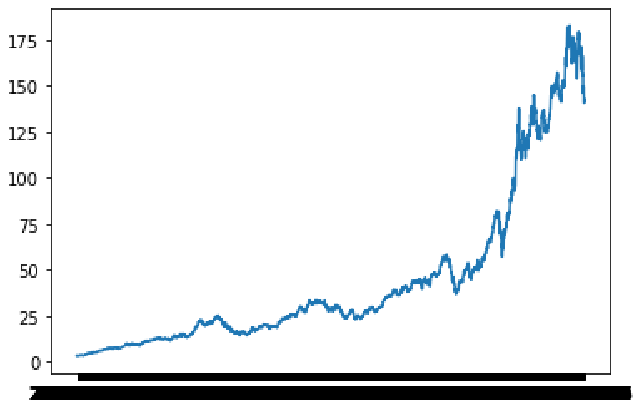
III. PROBLEM STATEMENT

The main objective of this project is to analyze the raw data and find the technical indicators to train the model on different models to find the model that would give us high profits by predicting whether to sell or buy a stock.

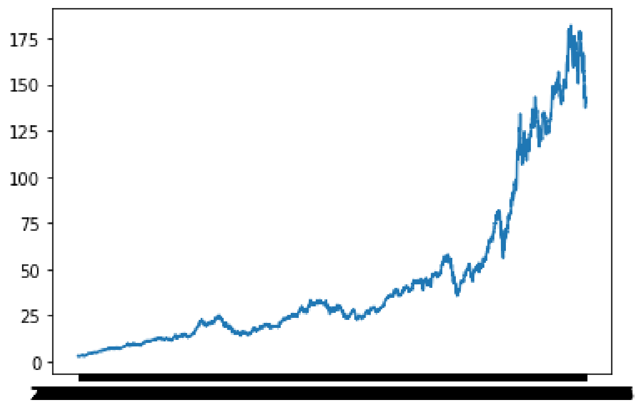
IV. PROPOSED APPROACH

Given a dataset with values such as high, low, open, and close we can derive many values like mean, standard deviation, Exponentially weighted moving average, relative strength index, Williams %R, Parabolic stop and reverse, and Average Directional Index.

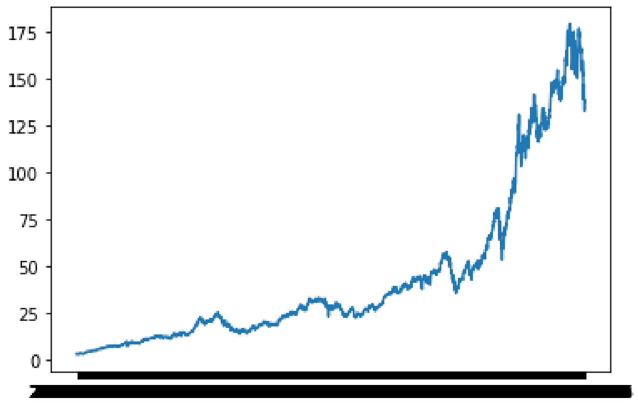
High-



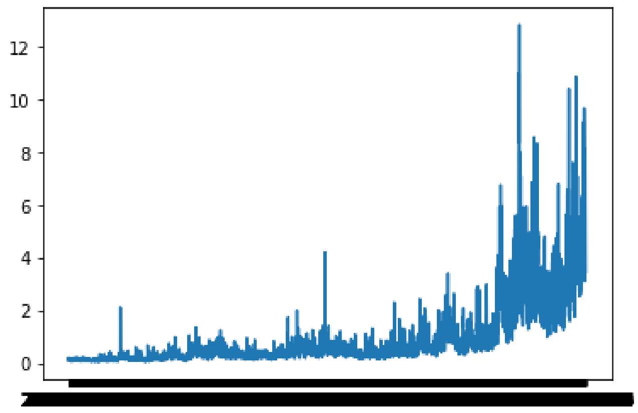
Close-



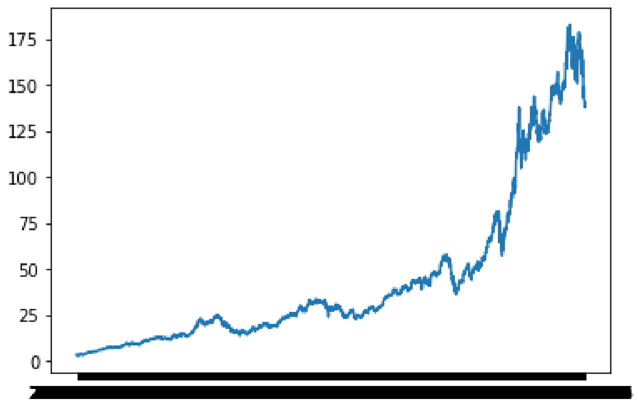
Low-



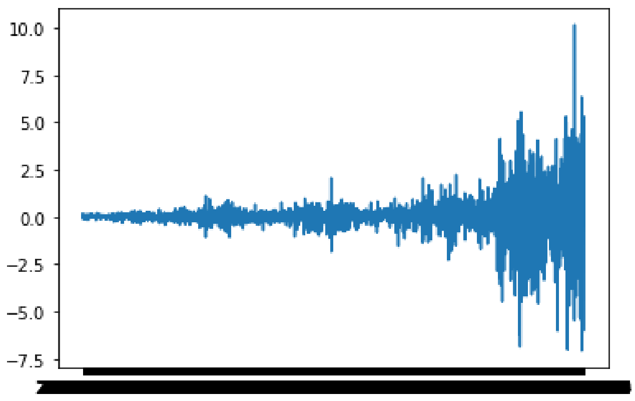
High-low-



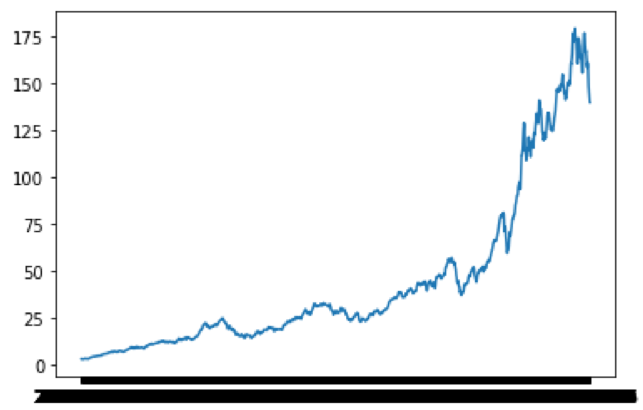
Open-



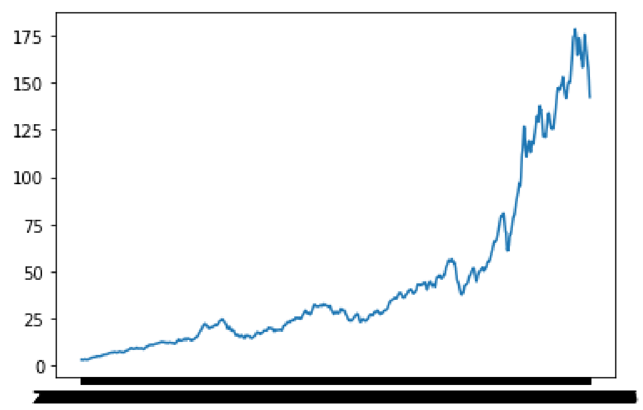
Open-Close-



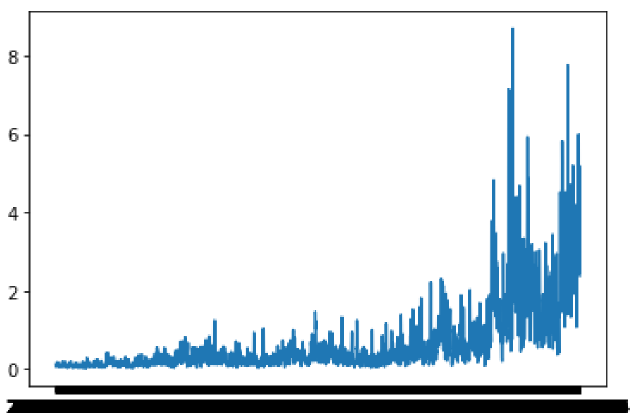
Mean_5-



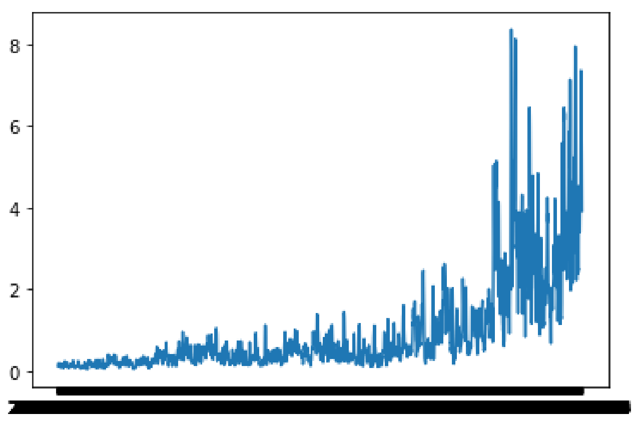
Mean-10-



standard deviation_5-



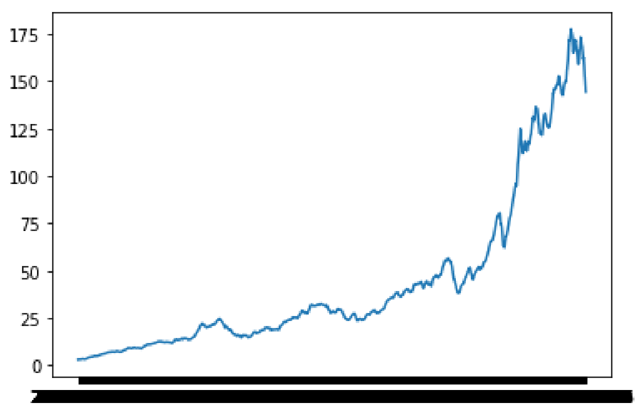
standard deviation_10-



Exponentially weighted moving average(EWMA)-

The moving average is designed such that older observations are given lower weights. the weights fall exponentially as the data point gets older hence more exponentially weighted.

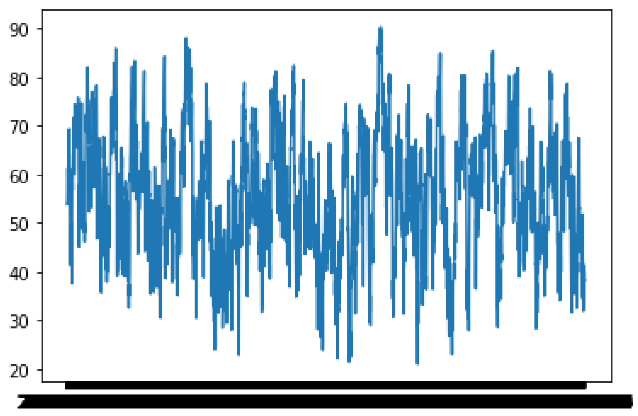
EWMA(t) = a * x(t) + (1-a) * EWMA(t-1)



Relative Strength index-

It is a momentum oscillator that measures the momentum of price movements in the stocks.It oscillates between 0 and 100. Traditionally the value is considered overbought when above 70 and oversold when below 30

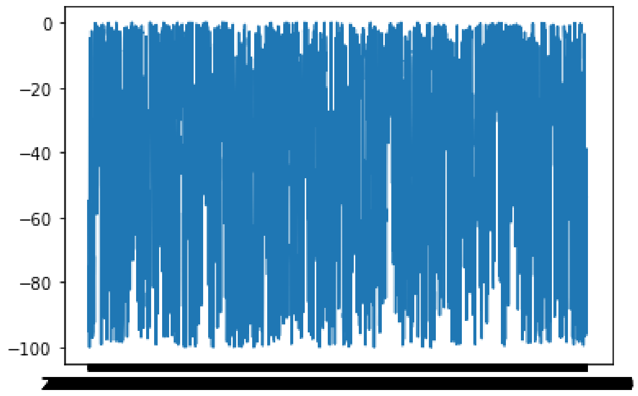
RSI = [100 - (100/{1+ RS})]



Williams %R-

It is a momentum indicator. Values from 0 to -20 are considered overbought. Values from -80 to -100 are considered oversold

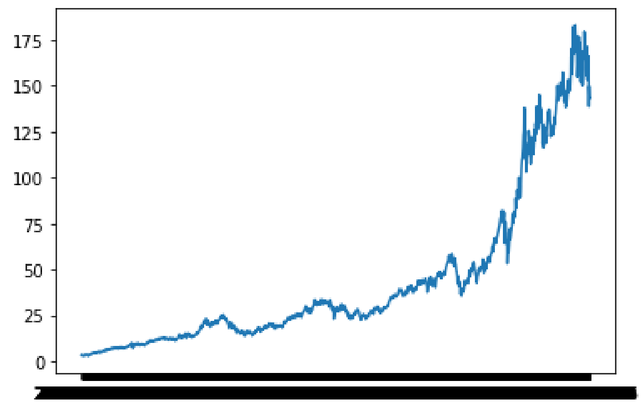
Williams %R = Highest high -Close/Highest high-Lowest low



Parabolic stop and reverse-

The Parabolic SAR trading system uses the parabolic level to identify the trend direction of the stock.

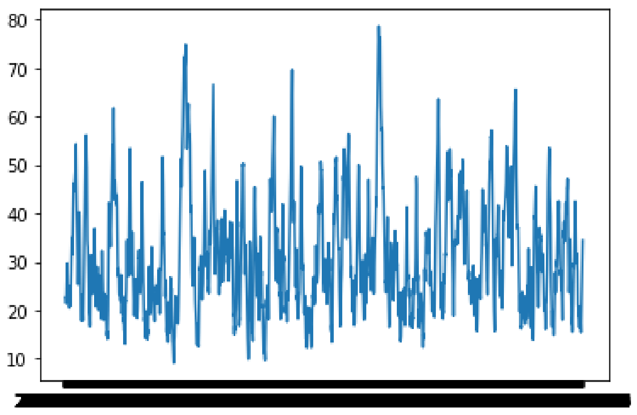
Uptrend: PSAR = Prior PSAR + Prior AF (Prior EP - Prior PSAR) Downtrend: PSAR = Prior PSAR - Prior AF (Prior PSAR - Prior EP)



Average Directional Index-

It can be used to determine the overall strength of a trend. The Average Directional Index indicator is an average of expanding price range values

$c(x, y, f($



From these, we can calculate whether to buy or sell a stock. The metric used for measuring this would be

1. We could compare the closing price of the previous day to the closing price of the current day. If the price of the previous day > the price of the current day sell the stock and if the price of the previous day < price of the current day buy the stock.
2. We could compare the mean of the closing price of the last 5 or 10 days to the closing price of the current day. If the price of the previous day > the price of the current day sell the stock and if the price of the previous day < price of the current day buy the stock.

Using these metrics we could calculate values for each day in the dataset and train these data on different classification techniques such as SVM, KNN, and Logistic regression to determine the algorithm which would give us the output of 0 and 1 where 1 signifies ‘SELL’ and 0 represents ‘BUY’ of the stock.

SVM-

Support vector machine works by finding a hyperplane in the N dimension space. There are many possible hyperplanes that could be chosen to separate two classes of the data points. Our objective is to find a plane that has the maximum margin. Maximizing the margin distance provides some assurance so that future data points can be classified with more accuracy and confidence.

In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane.

The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = (1 - y * f(x))_+$$

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

when the predicted value and the actual value are of the same sign the cost is 0. If the values are not the same, we then calculate the loss value. We also add a regularization parameter for the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions look as below.

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Now that we have the loss function, we take partial derivatives with respect to the weights to find the gradients. Using the gradients, we can update our weights.

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 \ = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} \left(1 - y_i \langle x_i, w \rangle\right)_+ \ = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

When there is no misclassification, i.e our model correctly predicts the class of our data point, we only have to update the gradient from the regularization parameter.

$$w = w - \alpha \cdot (2\lambda w)$$

When there is a misclassification, i.e our model makes a mistake on the prediction of the class of our data point, we include the loss along with the regularization parameter to perform a gradient update.

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$$

Logistic regression-

It is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, based on a given dataset of independent variables. The outcome of the model is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied to the odds—that is, the probability of success divided by the probability of failure. Also is commonly known as the log odds or the natural logarithm of odds, and this logistic function is represented by the following formulas:

Logit(pi) = 1/(1+ exp(-pi))

ln(pi/(1-pi)) = Beta_0 + Beta_1*X_1 + … + B_k*K_k

In this logistic regression equation, logit(pi) is the dependent or response variable and x is the independent variable. The beta parameter is commonly estimated via maximum likelihood estimation (MLE). This method tests different values of beta through multiple iterations to optimize for the best fit of log odds. All of these iterations produce the log-likelihood function, and logistic regression seeks to maximize this function to find the best parameter estimate. Once the optimal coefficient is found, the conditional probabilities for each observation can be calculated, logged, and summed together to yield a predicted probability. For binary classification, a probability less than .5 will predict 0 while a probability greater than 0.5 will predict 1.

KNN

The k-nearest neighbors (KNN) algorithm is a simple, machine learning algorithm that can be used to solve both classification and regression problems. It has a major drawback of becoming significantly slower as the size of that data in use grows.

It works by finding the distances between a vector and all the vectors in the data, selecting the specified number of examples (K) closest to the query, then voting for the most frequent label or averages the labels

V. RESULTS AND DISCUSSION

When using the Linear SVM we got an accuracy of 83.755%
With Precision and Recall as 85.826% and 85.602% respectively.
When using polygonal SVM we got an accuracy of 75.111%
With Precision and Recall as 75.779% and 82.722% respectively.

When using kernel SVM we got an accuracy of 53.800%
With Precision and Recall as 85.294% and 22.774% respectively.
When using KNN we got an accuracy of 70.789%
With Precision and Recall as 80.194% and 64.659% respectively.
When using Logistic regression we got an accuracy of 83.755% With Precision and Recall as 84.910% and 86.910% respectively.

VI. LIMITATIONS

1. SVM can take a long time to train.
2. The time complexity of SVM can be $O(n^2)$.
3. Finding the appropriate kernel for SVM is not an easy task.
4. If the number of observations is less than the number of features may lead to overfitting.
5. Tough to obtain complex relationships using regression more powerful algorithms may be used like neural networks.

VII. CONCLUSION AND FUTURE WORK

Linear SVM and logistic regression should be preferred over KNN and rbf SVM. These methods provide good results when the stock of the company is on the rise but due to their linear nature, they may not be efficient enough to get us the correct result.

Due to the oscillating nature of the stock market models like KNN were not able to perform well as the values were mixed together hence grouping was not good and was full of noise. Introducing a hyperplane may be fruitful so that the values are sufficiently separated for the algorithm to the cluster.

We conclude that a simple classification algorithm may be used to predict whether to buy or sell stock given the data is limited it may be used comparable to reinforced learning techniques. But when data like sentiment analysis are included it may be a bit hard for classification algorithms to do their job. In these cases, the Reinforced algorithms based on q learning may be used such as DDPG and TD3, or some policy-based algorithms such as PPO.

Acknowledgment

We would like to express our special thanks to our teacher Mr. Bharavi Mishra, who gave us the opportunity to do a wonderful project on machine learning. We came to know about so many new things.

References

https://www.geeksforgeeks.org/support-vector-machine-algorithm/#:~:text=Support%20Vector%20Machine(SVM)%20is,distinctly%20classifies%20the%20data%20points.

https://www.sciencedirect.com/topics/computer-science/logistic-regression#:~:text=Logistic%20regression%20is%20a%20process,%2Fno%2C%20and%20so%20on.

https://www.ibm.com/in-en/topics/knn

https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

https://towardsdatascience.com/understanding-random-forest-58381e0602d2

https://towardsdatascience.com/decision-tree-classifier-explained-in-real-life-picking-a-vacation-destination-6226b2b60575

https://www.investopedia.com/top-7-technical-analysis-tools-4773275

https://www.motilaloswal.com/blog-details/The-most-important-Technical-Indicator-tools-every-trader-should-know/1042

https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/technical-analysis/

https://www.ig.com/en/trading-strategies/10-trading-indicators-every-trader-should-know-190604

https://www.ijmttjournal.org/Volume-67/Issue-7/IJMTT-V67I7P515.pdf

https://www.akademiabaru.com/doc/ARBMSV14_N1_P35_41.pdf

https://www.ijbhtnet.com/journals/Vol_3_No_3_March_2013/4.pdf

https://blog.quantinsti.com/machine-learning-k-nearest-neighbors-knn-algorithm-python/

CODE-

https://github.com/jatindahiya027/Apple-Stock-prediction-using-classification_techniques