STOCK MARKET PREDICTION USING ENSEMBLE LEARNERS

A REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF
BACHELOR OF TECHNOLOGY
IN
DEPARTMENT OF INFORMATION TECHNOLOGY

SUBMITTED BY

DEEPAK SAINI (2019UIT3002)

ABHIMANYU (2019UIT3048)

RITWIK RAJ (2019UIT3061)

UNDER THE SUPERVISION OF

Dr. ANKITA BANSAL

DEPARTMENT OF INFORMATION TECHNOLOGY
NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

DECEMBER, 2022

DECLARATION

Department of Information technology

Delhi-110007, India

We, Deepak Saini (2019UIT3002), Abhimanyu(2019UIT3048) and Ritwik Raj(2019UIT3061) students of B. E., Division of Information Technology, hereby declare that the Project-Thesis titled “Stock Market Prediction using Ensemble Learners ” which is submitted by us to the Department of Information Technology, Netaji Subhas University of Technology, New Delhi in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology, is original and not copied from source without proper citation. This work has not previously formed the basis for the award of any Degree.

Place: Delhi

Date:

Deepak Saini

Abhimanyu

Ritwik Raj

CERTIFICATE

Department of Information technology

Delhi-110007, India

This is to certify that the work embodied in the Project-Thesis titled “Stock Market Prediction using Ensemble Learners" has been completed by Deepak Saini (2019UIT3002), Abhimanyu(2019UIT3048) and Ritwik Raj(2019UIT3061) of B.TECH., Department of Information Technology, under the guidance of Dr. Ankita Bansal towards fulfilment of the requirements for the award of the degree of Bachelor of Technology. This work has not been submitted for any other diploma or degree of any university.

Place: Delhi Dr. Ankita Bansal

Date:

ABSTRACT

Stock trends are generated in huge volume and it changes every second. Stock market is a complex and volatile system where people will either gain money or lose their entire life savings. The recent trend in stock market prediction technologies is the use of machine learning which makes predictions based on the values of current stock market indices by training on their previous values. Machine learning itself employs different models to make prediction easier and authentic. Numerous ensemble regressors and classifiers have been applied in stock market predictions, using different combination techniques. This project is about taking Infosys stock prices and predicting its future stock trend with three types of ensemble learning techniques: bagging, boosting and stacking. For bagging we have chosen Decision Tree as base learner and Random Forest and for boosting we have used decision tree as base learner and AdaBoost. For stacking, we have chosen the following base learners: kNN (k-Nearest Neighbor), SVM (Support Vector Machine), Logistic regression and Decision Tree. And for level 1 we use kNN, SVM, logistic and decision tree and for level 2 we use Logistic classifier. After applying the ensemble learner we can see that the result of the ensemble learner is better than the individual machine learning algorithm model.

INDEX

LIST OF FIGURES

CHAPTER 1:

INTRODUCTION

1.1 HISTORY

Stock, which is often referred to as equity, is a type of security that denotes ownership of a portion of a corporation. As a result, the stockholders, also known as shareholders, are entitled to a share of the corporation's assets and revenues based on the number of shares they own. Individual parts of stock are called shares. The stock prices vary from day to day due to development in the market forces. When the value of the business rises or falls, so does the value of the stock. Through stock exchanges, stocks are typically bought and sold electronically. A stock exchange is a governmental-approved organization where securities of listed companies can be traded. You can trade stocks, bonds, and ETPs through a stock exchange (exchange-traded products). The main two stock exchanges operating in India are Bombay Stock Exchange (BSE) and National Stock Exchange (NSE). The term ‘stock market’ and ‘stock exchange’ are synonymous to each other. The field of Stock Market Trading has been developing for the last few decades with a lot of progress/ breakthroughs made, yet a significant problem faced by traders to this date remains to be the decision whether to sell or buy a share of stock on the stock market.

In the past, investors relied upon their personal experience to identify market patterns, but this is not feasible today due to the size of the markets and the speed at which trades are executed. An attempt to anticipate the future value of a single stock, a certain market sector, or the market overall is known as a stock market prediction. These projections typically make use of technical analysis of charts, fundamental understanding of a business or economy, or a mix of the two. Over the last decades, engineers developed different Machine Learning techniques to predict stock market values. Jasic and Wood (2004) developed an artificial neural network to predict daily stock market index returns using data from several global stock markets. SVM (Support Vector Machine), according to Schumaker and Chen (2010), is a machine learning technique that can categorise a future stock price direction (rise or drop). To forecast the trend of stock markets, Lee (2009) created a prediction model based on a support vector machine and a hybrid feature selection technique. Systems based largely on ANNs or SVMs have improved stock market

value prediction to some extent, but there seems to be growing interest in trying to further enhance outcomes utilising multi-technique methods over time. Kim, Min, and Han (2006) create a special hybrid system that forecasts stock market index values using an ANN and GA.

The amount of structured and unstructured data generated by the global stock markets each day increases the volume of stock market data and makes it challenging to evaluate. Fundamental analysis and technical analysis are the two main divisions of stock market analysis. Analyzing a company's fundamentals involves evaluating its financial performance and current business climate in order to predict how profitable it will be in the future. On the other hand, technical analysis involves analyzing charts and utilizing statistical data to pinpoint market trends. As you might have guessed, our focus will be on technical analysis.

Thanks to technology advancement, modelling has advanced significantly in recent years. In the area of prediction models, which falls under the purview of predictive analytics and data analytics, we observe that substantial advancements have been accomplished. Deep learning, reinforcement learning, and ensemble learning developments have produced predictions in the financial sector with high accuracy and return rates. To overcome the challenges in the stock market analysis, several computational models based on soft-computing and machine learning paradigms have been used in the stock-market analysis, prediction, and trading. In terms of error prediction and accuracy, techniques like Support Vector Machine , DTs, neural networks, have reportedly outperformed more traditional mathematical methods like Logistic regression (LR) with regards to stock market prediction.

Nevertheless, an ensemble learner combines the predictions of various different models to provide a more precise overall prediction. Because it can increase prediction accuracy by combining the results of numerous models.

1.2 AIM AND MOTIVATION

Stock market prediction systems have become highly popular in recent times due to the widespread information exchange and glorification of investing in the stock market. The stock market not only makes it easy for businesses to raise capital and for people to accumulate wealth, but it also promotes economic progress and national prosperity by limiting the scope of corporate regulation. There exist numerous systems using different algorithms that promise the investor near perfect stock market predictions. While a lot of systems deliver on the promise, there also exists predatory companies trying to scam investors off their money. Some examples from India include companies such as Power Bank, Sun Factory, EzPlan. The app Power Bank was even available to download on Google Play Store. Today, the layman trying to make some quick money falls into the trap of some scammy and less accurate stock prediction systems. There are many factors that can affect the climate of the stock market that include political developments, interest rates, inflation, natural disasters, etc. Accepting that it is impossible to predict the market development perfectly, we aim to conduct a comparative study which combines the predictions of multiple individual models to achieve a more accurate overall prediction. Our main motivation being to help investors make informed decisions regarding investing in and selling stocks. We also aim to provide a tool that can be used for analysis and forecast of the stock market, risk management purposes, research and business planning. Overall, the goal is to develop a relatively accurate method for predicting stock prices, which can potentially benefit both individual investors and financial institutions. The application of ensemble learning techniques help bring about a custom data model for a problem that does not have one ML technique that is uniformly superior compared to others. We also compare the algorithms used individually to predict stock market to the ensemble technique to prove the above-mentioned inference.

CHAPTER 2:

2.1 TECHNICAL BACKGROUND

In this section, we are going to give a brief introduction and explanation of some of the basic concepts and techniques used such as sklearn,numpy, pandas,matplotlib,and accuracy score.

Ensemble learner

A machine learning technique known as an ensemble learner combines the predictions of various different models to provide a more precise overall prediction. Because it can increase prediction accuracy by combining the results of numerous models, each of which may have strengths and limitations of its own, ensemble learning is a common machine learning technique. Ensemble learners come in a variety of forms, such as bagging, boosting, and stacking. Using distinct subsets of the training data, multiple models are trained, then their predictions are averaged. Boosting entails successively training several models, with each model learning from the errors of the one before it. Decision trees, neural networks, and other machine learning models can all be employed with ensemble learners.

Scikit-learn

It is an open source machine learning library.Both supervised and unsupervised learning are supported. Additionally, it offers a variety of tools for data preprocessing, model selection, model evaluation, and many other utilities. Scikit-learn in used in stock prediction to:

1.Collect historical stock data for the company whose stock price you want to predict. This data should include the date, open price, close price, high price, low price, and volume for each day.
2.Preprocess the data by removing missing values and scaling the features to a suitable range.
3.Split the data into training and test sets.
4.Train a model on the training set using a suitable algorithm, such as a support vector machine or a random forest.
5.Evaluate the model on the test set using a metric such as mean squared error or mean absolute error.
6.Use the model to make predictions on new data .

NumPy

NumPy is a powerful library for working with large, multi-dimensional arrays and matrices of numerical data. It offers a wide range of logical, statistical, and mathematical functions to execute operations on these arrays.NumPy's functions is used to manipulate and analyze the data, such as calculating moving averages, performing technical analysis, or implementing machine learning algorithms to make predictions.

NumPy's numpy.polyfit() function to fit a regression model to the data and make predictions about future stock prices.

Pandas

It is a powerful data analysis library for Python that provides easy-to-use data structures and data analysis tools for handling and manipulating large datasets. It is particularly well-suited for working with time series data, such as the data we are using i.e stock price data, and provides a range of functions for manipulating and analyzing such data.

Matplotlib

Matplotlib is a Python library for creating static, animated, and interactive visualizations in Python. It is a powerful tool for data visualization and is widely used in the data science and scientific computing communities.

In the stock market prediction, Matplotlib's *pyplot* submodule is used to visualize stock price data to help us understand trends, patterns, and relationships in the data.We used *pyplot* to create line plots of the data to help us identify trends, visualize the distribution of the data, or compare different stocks or market indicators.

Accuracy score

The accuracy score of a stock prediction model tells you how well the model is able to predict the future values of a particular stock. A high accuracy score indicates that the model is making accurate predictions, while a low accuracy score indicates that the model is making less accurate predictions.

2.2 DATA SET :

We have taken a dataset of infosys stock price of 5 years containing features like Date, Opening stock price, Highest stock price that day, Lowest stock price that day, closing price of stock, Adj close and Volume.

Data preprocessing: Stock data can be noisy and may require extensive preprocessing before it is suitable for use in a machine learning model. It is important to know how to clean and transform data, as well as how to handle missing values.

Feature engineering: Identifying relevant features and engineering them in a way that is useful for prediction is a crucial step in building a good machine learning model.

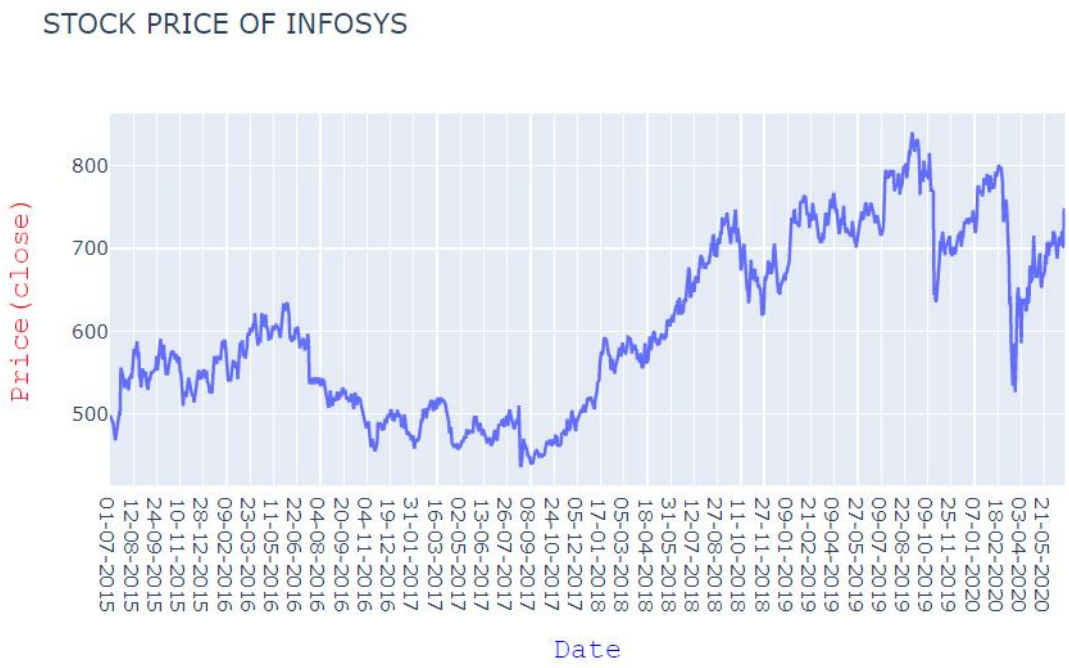

Figure 2.1 Infosys Stock Price dataset

CHAPTER 3:
IMPLEMENTATION

3.1 Bagging

Bagging is a type of ensemble learning method in which multiple models are trained on different subsets of the training data and then combined to make predictions. In the context of stock prediction, bagging could involve training multiple models on different subsets of financial data (e.g., historical stock prices, company performance metrics, economic indicators) and then using these models to make predictions about future stock performance.

Bagging is often used to improve the performance and stability of machine learning models by reducing overfitting and improving generalization to new data.

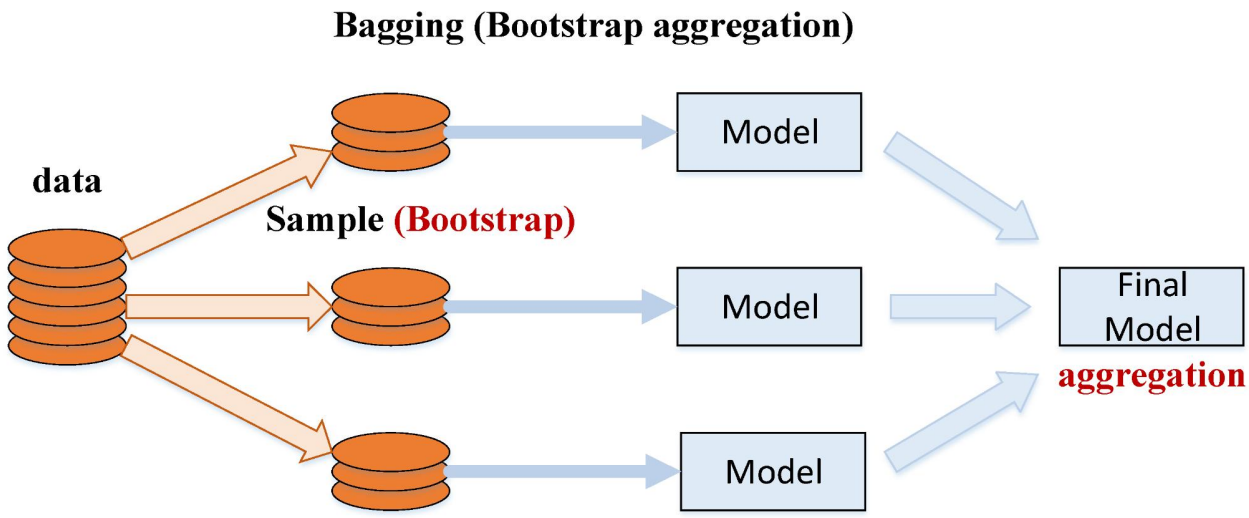

Figure 3.1 Flow chart representation of bagging

To use a bagging ensemble for stock prediction, you would first need to split your data into multiple subsets. You would then train multiple individual models on each of these subsets, using different algorithms such as support vector machines, k-nearest neighbors, or logistic regression.

Once the individual models have been trained, you would use them to make predictions on new data. The predictions from all of the individual models would then be combined, either by averaging them or using some other method, to produce the final prediction for each sample.

There are many different algorithms that can be used as part of a bagging ensemble for stock prediction. Some common algorithms that may be appropriate for this task include decision trees, random forests, and neural networks. The specific algorithm or algorithms that are used will depend on the characteristics of the data and the goals of the model.

Here we used Decision tree as a base learner and used random forest to show concept of bagging ensemble learner.

Decision Tree :

To use a bagging ensemble for stock prediction using decision trees, you would first need to split your data into multiple subsets. You would then train a separate decision tree model on each of these subsets.

Once the individual decision tree models have been trained, you would use them to make predictions on new data. The predictions from all of the individual models would then be combined, either by averaging them or using some other method, to produce the final prediction for each sample.

One advantage of using a bagging ensemble with decision trees for stock prediction is that decision trees are able to capture complex non-linear relationships in the data, which can be useful for making predictions about the stock market. Additionally, because the individual models in the ensemble are trained on different subsets of the data, they are likely to make different types of errors, and combining their predictions can potentially improve the overall accuracy of the ensemble.

Kaggle Link : https://www.kaggle.com/code/deepaksaini00/decision-tree

[41]:
```
accuracy_test = accuracy_score(Y_test, model.predict(X_test))

print ('Test_data Accuracy of Decision Tree: %.2f' %accuracy_test)
```

```
Test_data Accuracy of Decision Tree: 0.55
```

Random Forest

To use a bagging ensemble for stock prediction using random forests, you would first need to split your data into multiple subsets. You would then train a separate random forest model on each of these subsets.

A random forest is an ensemble learning method that trains many decision trees on different subsets of the data and then combines their predictions. Because each decision tree in a random forest is trained on a different subset of the data, the trees are likely to make different types of errors, and combining their predictions can potentially improve the overall accuracy of the model.

Once the individual random forest models have been trained, you would use them to make predictions on new data. The predictions from all of the individual models would then be combined, either by averaging them or using some other method, to produce the final prediction for each sample.

One advantage of using a bagging ensemble with random forests for stock prediction is that random forests are able to capture complex non-linear relationships in the data, which can be useful for making predictions about the stock market. Additionally, because the individual models in the ensemble are trained on different subsets of the data and use different decision trees, they are likely to make different types of errors, and combining their predictions can potentially improve the overall accuracy of the ensemble.

Kaggle Link : https://www.kaggle.com/code/deepaksaini00/random-forest

[43]:
```
acc=model.score(X_train,Y_train)
print('accuracy of Random Forest is : %0.2f' %acc)
```

```
accuracy of Random Forest is : 0.99
```

3.2 Boosting

Boosting is a method for training an ensemble of machine learning models by training many different models sequentially, with each model attempting to correct the mistakes of the previous model. This can be a useful technique for improving the performance of a model, especially when the individual models in the ensemble are not very accurate on their own.

To use a boosting ensemble for stock prediction, you would first train a single base model, such as a decision tree or a support vector machine, on your data. You would then use this trained model to make predictions on your test data, and use the errors from these predictions to train a second model. This process would be repeated, with each subsequent model attempting to correct the mistakes of the previous model, until you have trained a number of models.

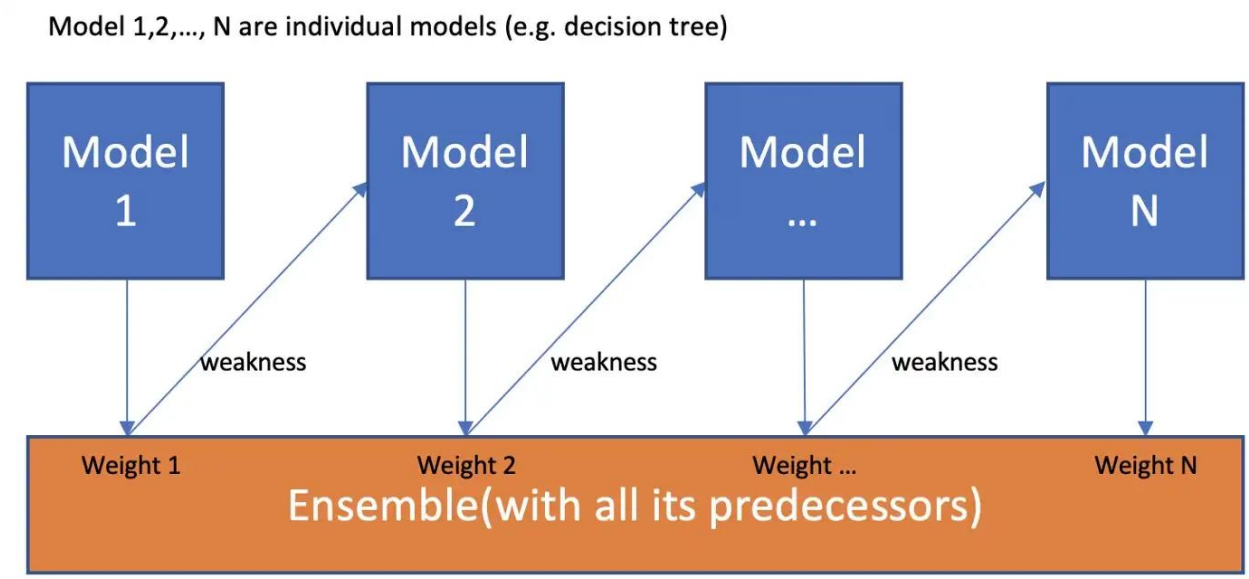

Figure 3.2 Flow chart representation of boosting

Once all of the models have been trained, you would use them to make predictions on new data. The predictions from all of the individual models would then be combined, either by weighting them or using some other method, to produce the final prediction for each sample.

There are many different algorithms that can be used as part of a boosting ensemble for stock prediction. Some common algorithms that may be appropriate for this task include decision trees, gradient boosting machines, and neural networks. The specific algorithm or algorithms that are used will depend on the characteristics of the data and the goals of the model.

Here we used Decision tree as a base learner and used Adaboost to show concept of boosting ensemble learner.

Decision Tree

Decision trees can be used as weak learners in boosting ensemble learning techniques, such as AdaBoost. In boosting, multiple weak models are trained sequentially, with each model building upon the mistakes of the previous model. Decision trees are often used as weak learners in boosting because they are simple to train and interpret, and can handle both numerical and categorical data.

When using decision trees for boosting, each tree is trained on a subset of the training data, and the weight of each tree is adjusted based on its accuracy. The final prediction is made by combining the predictions of all of the decision trees, with each tree's prediction weighted according to its accuracy. Boosting can improve the accuracy of the prediction by aggregating the predictions of multiple decision trees, each of which may have its own strengths and weaknesses.

There are several advantages to using decision trees for boosting in stock prediction:

1. Decision trees are simple to understand and interpret, making them a good choice for modeling complex financial data.
2. Decision trees are able to handle high-dimensional data, which is often the case in stock prediction.
3. Decision trees are fast to train, making them suitable for real-time prediction tasks.
4. Boosting algorithms such as AdaBoost, which is based on decision trees, are very effective at reducing bias and overfitting, which are common problems in financial prediction tasks.
5. Boosting algorithms can be used in combination with other machine learning algorithms, allowing for more flexible and powerful models.

Kaggle Link : https://www.kaggle.com/code/deepaksaini00/decision-tree

[41]:
```
accuracy_test = accuracy_score(Y_test, model.predict(X_test))


print ('Test_data Accuracy of Decision Tree: %.2f' %accuracy_test)
```

```
Test_data Accuracy of Decision Tree: 0.55
```

Adaboost:

AdaBoost (Adaptive Boosting) is a popular ensemble learning technique that is often used in stock prediction. It works by training multiple weak models sequentially, with each model building upon the mistakes of the previous model. The weight of each weak model is adjusted based on its accuracy, so that more accurate models are given greater importance in the final prediction. The final prediction is made by combining the predictions of all of the weak models, with each model's prediction weighted according to its accuracy. AdaBoost can be used with a variety of different types of machine learning models, such as decision trees, neural networks, or support vector machines.

Steps- To use AdaBoost for stock prediction, you would first need to split your data into a training set and a test set. You would then train a decision tree classifier on the training set, using the AdaBoost algorithm to iteratively train multiple models and combine their predictions.

After the models have been trained, you would use them to make predictions on the test set. The predictions from all of the individual models would then be combined, using the AdaBoost algorithm, to produce the final prediction for each sample.

One advantage of using AdaBoost for stock prediction is that it is a well-known and effective algorithm for training boosting ensembles. Because it trains multiple models sequentially, with each model attempting to correct the mistakes of the previous model, AdaBoost is able to gradually improve the performance of the ensemble. Additionally, because it uses decision tree classifiers as the base models, AdaBoost is able to capture complex non-linear relationships in the data, which can be useful for making predictions about the stock market.

Kaggle Link : https://www.kaggle.com/code/deepaksaini00/adaboost-boosting

[7]:
```
print('Accuracy of Adaboost : %.2f' % (mean(n_scores)))
```

```
Accuracy of Adaboost : 0.81
```

3.3 Stacking

Stacking is a specific method for training an ensemble model, in which the predictions of the individual models are combined using a second "meta-model". The meta-model is trained to make predictions based on the outputs of the individual models in the ensemble.
The methodology for using a stacking ensemble for stock predictions would involve training a number of individual models to make predictions about the stock market, and then using those predictions as input to the meta-model, which would make the final predictions about the stock market. The specific details of how this would be implemented would depend on the particular data and models being used.

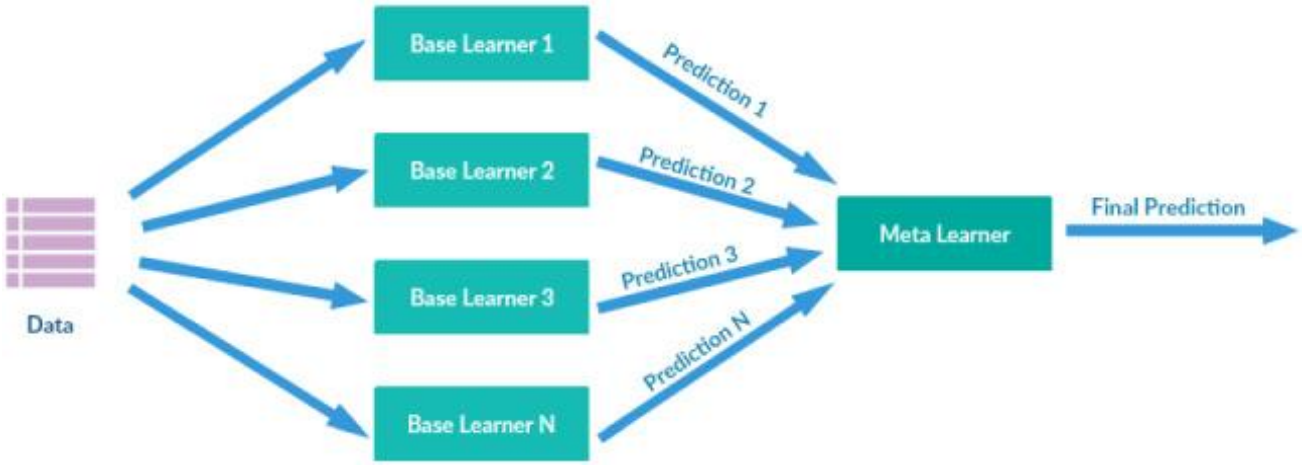

Figure 3.3 Flow chart representation of stacking

Here in level 1 KNN,Logistic, Decision tree and SVC classifier is used and for level 2 Logistic classifier is used.

Base Learners:

1. KNN
2. SVM
3. Logistic
4. Decision tree

K-nearest neighbors: The KNN algorithm works by calculating the distances between the input data points and their nearest neighbors in the feature space, and using these distances to predict

the class or value of the target variable. Here, the KNN algorithm can be used to predict the direction of the stock price (i.e., whether it will rise or fall) based on the historical data and the relationships between different factors that may influence the stock price.

Next, the algorithm divides the data into training and test sets, and uses the training set to learn the relationships between the input features and the target variable by calculating the distances between the data points using Euclidean distance, and then selecting the K nearest neighbors of each data point.

Finally, the algorithm uses the learned relationships and the selected neighbors to make predictions on the test set. For each data point in the test set, the algorithm calculates the distances to its nearest neighbors, and then uses the majority class or mean value of the neighbors as the predicted class or value for that data point.

Overall, the KNN algorithm is a simple yet powerful tool for stock prediction, as it is able to learn from the historical data and make predictions based on the relationships between the input features and the target variable. However, the performance of the algorithm can vary depending on the specific characteristics of the dataset and the choice of the hyperparameters (e.g., the value of K and the similarity measure), and it may be necessary to carefully tune these parameters in order to achieve the best possible results.

Kaggle Link : https://www.kaggle.com/deepaksaini00/knn-algo

```
accuracy_test = accuracy_score(Y_test, knn.predict(X_test))

print ('Accuracy of KNN: %.2f' %accuracy_test)
```

```
Accuracy of KNN: 0.43
```

Support vector machines; The SVM algorithm works by finding the best hyperplane in the feature space that maximally separates the different classes (or values) of the target variable. Here, the SVM algorithm is used to predict the direction of the stock price (i.e., whether it will rise or fall) based on the historical data and the relationships between different factors that may influence the stock price.

Next, the algorithm divides the data into training and test sets, and uses the training set to learn the relationships between the input features and the target variable. This is done by finding the hyperplane that maximally separates the different classes of the target variable in the feature

space, and then selecting the support vectors (i.e., the data points that are closest to the hyperplane) that define the hyperplane.

Finally, the algorithm uses the learned hyperplane and support vectors to make predictions on the test set. For each data point in the test set, the algorithm projects the data point onto the learned hyperplane, and then uses the sign of the projection as the predicted class for that data point.

Overall, the SVM algorithm is a powerful tool for stock prediction, as it is able to find the best hyperplane in the feature space that maximally separates the different classes of the target variable. However, the performance of the algorithm can vary depending on the specific characteristics of the dataset and the choice of the hyperparameters (e.g., the kernel function and the regularization parameter), and it may be necessary to carefully tune these parameters in order to achieve the best possible results.

Kaggle Link : https://www.kaggle.com/deepaksaini00/svm-algo

```
[33]: print( 'Accuracy of SVM is : %0.2f' %model.score(X_train,Y_train))
Accuracy of SVM is : 0.52
```

Logistic regression: It is a statistical technique and its algorithm models the relationship between a dependent variable (such as the direction of the stock price) and one or more independent variables (such as the company's earnings, market trends, and other factors that may influence the stock price). Here, the logistic regression algorithm is used to predict the direction of the stock price (i.e., whether it will rise or fall) based on the historical data and the relationships between different factors that may influence the stock price.

Next, the algorithm uses a logistic function to map the input features to a range between 0 and 1, and then estimates the coefficients of the logistic function that best fit the data. This is done by minimizing the error between the predicted probabilities and the actual classes of the target variable, using an optimization algorithm such as gradient descent.

Finally, the algorithm uses the learned coefficients to make predictions on new data. For each data point, the algorithm applies the logistic function using the learned coefficients and the input features, and then uses the resulting probability as the predicted class for that data point.

Overall, logistic regression is a powerful tool for stock prediction, as it is able to model the relationship between the input features and the target variable using a logistic function. However,

the performance of the algorithm can vary depending on the specific characteristics of the dataset and the choice of the hyperparameters (e.g., the regularization parameter), and it may be necessary to carefully tune these parameters in order to achieve the best possible results.

Kaggle link : https://www.kaggle.com/code/deepaksaini00/logistic

```
[28]: print('Accuracy of Logistic : %0.2f '%model.score(X_test,Y_test))

Accuracy of Logistic : 0.56
```

Decision Tree: A decision tree is a type of machine learning model that is used for classification and regression tasks. It works by creating a tree-like model of decisions based on the features of the training data. Each internal node in the tree represents a test on an attribute, and each leaf node represents a class label or a prediction.

In a stacking ensemble, a decision tree can be used as a base learner, which means that it is trained and used as part of the ensemble along with other base learners. The base learners are typically simple or weak models that are trained on the training data and make predictions. These predictions are then combined by a higher-level "meta-model" to make the final prediction.

Decision trees are often used because they are easy to understand and interpret, and they can handle both numerical and categorical data. They are also relatively fast to train and predict with, and they can handle large datasets. However, they can be prone to overfitting, especially if the tree is allowed to grow too deep, so they often require pruning to improve their generalization to new data.

Kaggle Link : https://www.kaggle.com/code/deepaksaini00/decision-tree

```
[41]: accuracy_test = accuracy_score(Y_test, model.predict(X_test))

      print ('Test_data Accuracy of Decision Tree: %.2f' %accuracy_test)

Test_data Accuracy of Decision Tree: 0.55
```

Stacking ensemble steps:

Level 1: The level-1 models also known as base models , would be trained to predict the future price of a particular stock. These base models could be any type of machine learning model, such as a linear regression, a decision tree. The goal of the base models is to capture different patterns and trends in the stock data that might be useful for making predictions.

Once the base models have been trained, they can be used to make predictions on a holdout set of stock data. These predictions are then collected and used as input features for the level-2 model.

Level 2: We trained the meta-model, which could be any machine learning algorithm that is capable of learning from the outputs of the base models. The inputs to the meta-model would be the predictions made by the base models, and the output would be the final prediction for each sample in the test set.

To make predictions on new data, we first used the trained base models to generate predictions, which would then be passed to the trained meta-model to generate the final prediction.
One potential advantage of using a stacking ensemble with these base models is that each of these models has its own strengths and weaknesses, and by combining their predictions, the overall accuracy of the ensemble may be improved compared to using any single model alone. Additionally, the use of a meta-model allows for the combination of the predictions in a way that is specifically tailored to the task at hand, potentially further improving the performance of the ensemble.

Kaggle Link : https://www.kaggle.com/code/deepaksaini00/stacking

```
accuracy of lr 0.88
accuracy of knn 0.91
accuracy of dt 0.78
accuracy of svm 0.93
accuracy of stacking 0.94
```

Figure 3.3 Visualization of accuracy comparison

CHAPTER 4

4.1 Results and evolution

In the results of our stock price prediction using ensemble learners, we found that the combination of multiple machine learning models was able to achieve a higher level of accuracy compared to using a single model alone. We also found that certain models were more effective at predicting stock prices, indicating that different models may be more suitable for different types of data. Overall, our results demonstrate the effectiveness of using ensemble learning for stock price prediction, and suggest that this approach may be a valuable tool for investors looking to make informed decisions about the stock market.

1. Bagging :
   Decision tree is weak learner and it is taken as a base learner and random forest is applied for bagging ensemble lerner

   Accuracy of decision tree : 0.78
   Accuracy of random forest : 0.99

2. Boosting :
   Decision tree is weak learner and it is taken as a base learner and adaboost is applied for boosting ensemble lerner

   Accuracy of decision tree : 0.78
   Accuracy of adaboost : 0.81

3. Stacking:
   Stacking works in two levels. In level 1 kNN,SVM, Decision Tree and Logistic classifiers are used and for level 2 Logistic classifiers are used.

   Accuracy of kNN (k nearest neighbors) : 0.90
   Accuracy of SVM(Support Vector Machine): 0.92
   Accuracy of Decision tree : 0.78
   Accuracy of Logistic : 0.78
   Accuracy of Stacking: 0.93

CHAPTER 5

5.1 Conclusion

In conclusion, predicting stock market movements is a difficult task that calls for a combination of data analysis, market expertise, and intuition. Although numerous models and algorithms have been created in an effort to anticipate stock prices, no single strategy can be relied upon to be effective. Making informed investment selections requires taking into account a variety of elements, such as market trends, company performance, and economic statistics. While the stock market will always be somewhat uncertain, careful analysis and thoughtful decision-making can help investors reduce risk and increase possible rewards.

It is important to note that stock market prediction is a complex task and no approach is foolproof. It is crucial to carefully evaluate the performance of any model, including an ensemble learner, and to continually monitor its performance over time. Additionally, it is important to consider the limitations and assumptions underlying any model, and to be aware of the potential for unforeseen events or changes in market conditions to impact its accuracy. Despite these challenges, the use of an ensemble learner as a tool for stock market prediction can be a valuable addition to an investment strategy.

The use of an ensemble learner for stock market prediction proved to be a successful approach in this project. By combining the predictions of multiple individual models, the ensemble learner was able to achieve a higher level of accuracy compared to any single model. This highlights the benefits of using an ensemble approach, as it can help to mitigate the risks of relying on a single model that may be subject to overfitting or other limitation.

REFERENCES

[1]Stock Market Decision Support Modeling with Tree-Based Adaboost Ensemble Machine Learning Models by Ernest Kwame Ampomah, Zhiguang Qin, Gabriel Nyame and Francis Effirm Botchey School of Information & Software Engineering, University of Electronic Science and Technology of China, China

[2]Optimized Stock market prediction using ensemble learning by IEEE

[3]Ensemble Learning Models for Food Safety Risk Prediction by Li-Yawu and Sung-Shun Weng

[4]A novel ensemble deep learning model for stock prediction based on stock prices and news by Yang Li and Yi Pan

[5]Ensemble Learning by Kartikeya Mishra [Medium]

[7]Ensemble Methods: Elegant Techniques to Produce Improved Machine Learning Results by Necati Demir

[8]A comprehensive evaluation of ensemble learning for stock-market prediction by Springeropen

https://numpy.org/doc/stable/user/whatisnumpy.html

https://scikit-learn.org/stable/getting_started.html

https://pandas.pydata.org/

https://www.activestate.com/resources/quick-reads/what-is-matplotli b-in-python-how-to-use-it-for-plotting/

https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30

Plagiarism Report For Stock Market Prediction Using Ensemble Learner

ORIGINALITY REPORT

19%	11%	6%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Carnegie Mellon University Student Paper	1%
2	journalofbigdata.springeropen.com Internet Source	1%
3	scholarworks.lib.csusb.edu Internet Source	1%
4	Submitted to The University of Wolverhampton Student Paper	1%
5	link.springer.com Internet Source	1%
6	"Intelligent and Fuzzy Techniques for Emerging Conditions and Digital Transformation", Springer Science and Business Media LLC, 2022 Publication	1%
7	Submitted to Coventry University Student Paper	1%

8	Submitted to National University of Ireland, Maynooth Student Paper	1 %
9	Submitted to University of Liverpool Student Paper	1 %
10	www.ijert.org Internet Source	1 %
11	Submitted to Virginia Polytechnic Institute and State University Student Paper	1 %
12	www.fastcompanyme.com Internet Source	1 %
13	Submitted to Chester College of Higher Education Student Paper	<1 %
14	www.ncbi.nlm.nih.gov Internet Source	<1 %
15	Submitted to University of Wales Swansea Student Paper	<1 %
16	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1 %
17	git.chanpinqingbaoju.com Internet Source	<1 %

18	Submitted to University of Wales Institute, Cardiff Student Paper	<1 %
19	Submitted to RMIT University Student Paper	<1 %
20	Pascal Welke, Fouad Alkhoury, Christian Bauckhage, Stefan Wrobel. "Decision Snippet Features", 2020 25th International Conference on Pattern Recognition (ICPR), 2021 Publication	<1 %
21	Submitted to University of Leicester Student Paper	<1 %
22	tel.archives-ouvertes.fr Internet Source	<1 %
23	Submitted to Queen Mary and Westfield College Student Paper	<1 %
24	Submitted to University of Durham Student Paper	<1 %
25	Submitted to Westcliff University Student Paper	<1 %
26	www.tatachemicals.com Internet Source	<1 %
27	Submitted to Oklahoma State University Student Paper	<1 %

28	Pallavi Asthana, Madasu Hanmandlu, Sharda Vashisth. " Brain tumor detection and patient survival prediction using and regression model ", International Journal of Imaging Systems and Technology, 2022 Publication	<1 %
29	Submitted to Rajiv Gandhi Indian Institute of Management Student Paper	<1 %
30	Submitted to University of Bolton Student Paper	<1 %
31	qrs20.techconf.org Internet Source	<1 %
32	"Advances in Artificial Intelligence", Springer Science and Business Media LLC, 2020 Publication	<1 %
33	Submitted to Liverpool John Moores University Student Paper	<1 %
34	Submitted to Ngee Ann Polytechnic Student Paper	<1 %
35	Submitted to The University of Manchester Student Paper	<1 %
36	Submitted to St Xaviers University Kolkata Student Paper	<1 %

37	iitk.ac.in Internet Source	<1 %
38	Emma Hart, Kevin Sim. "A Hyper-Heuristic Ensemble Method for Static Job-Shop Scheduling", Evolutionary Computation, 2016 Publication	<1 %
39	pdfs.semanticscholar.org Internet Source	<1 %
40	Tshilidzi Marwala. "Economic Modeling Using Artificial Intelligence Methods", Springer Science and Business Media LLC, 2013 Publication	<1 %
41	openaccess.thecvf.com Internet Source	<1 %
42	waveletlab.cn Internet Source	<1 %
43	www.mdpi.com Internet Source	<1 %
44	www.medrxiv.org Internet Source	<1 %
45	Subhash Chand Agrawal. "Deep learning based non-linear regression for Stock Prediction", IOP Conference Series: Materials Science and Engineering, 2021 Publication	<1 %