STOCK MARKET PREDICTION

Submitted for the Summer Internship

On

Python and Machine Learning

(from 5th June, 2023 to 23rd July, 2023)

Organized by

**Centre of Excellence (COE) – Artificial Intelligence (AI),
AI Club IGDTUW, Anveshan Foundation, IGDTUW**

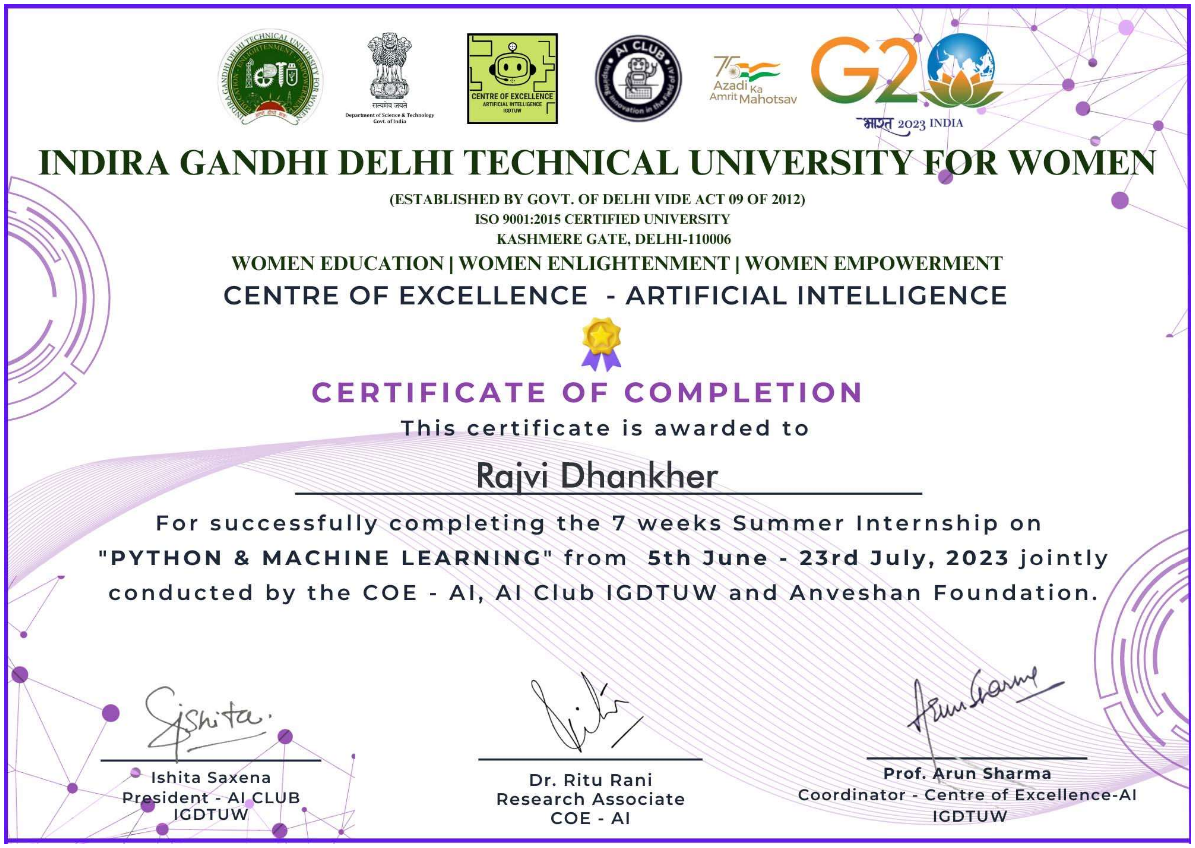
By

Rajvi Dhankher
14201012022
B. Tech CSE II
2022-2026

INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN

**(Established by Govt. of Delhi vide Act 9 of 2012)
Kashmere Gate, New Delhi, Delhi 110006**

CERTIFICATE



INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN

(ESTABLISHED BY GOVT. OF DELHI VIDE ACT 09 OF 2012)
ISO 9001:2015 CERTIFIED UNIVERSITY
KASHMERE GATE, DELHI-110006

WOMEN EDUCATION | WOMEN ENLIGHTENMENT | WOMEN EMPOWERMENT

CENTRE OF EXCELLENCE - ARTIFICIAL INTELLIGENCE

CERTIFICATE OF COMPLETION

This certificate is awarded to

Rajvi Dhankher

For successfully completing the 7 weeks Summer Internship on "**PYTHON & MACHINE LEARNING**" from **5th June - 23rd July, 2023** jointly conducted by the COE - AI, AI Club IGDTUW and Anveshan Foundation.

Ishita.

Ishita Saxena
President - AI CLUB
IGDTUW

Dr. Ritu Rani
Research Associate
COE - AI

Prof. Arun Sharma
Coordinator - Centre of Excellence-AI
IGDTUW

DECLARATION

I hereby declare that this project work titled Stock Market Prediction has been prepared by me during the year 2023 under the guidance of Dr Ritu Rani, Research Associate, COE-AI, IGDTUW.

I also declare that this project is the outcome of my own effort, that it has not been submitted to any other university for the award of any degree.

Rajvi Dhankher

B.Tech. CSE
Roll No. 14201012022

ACKNOWLEDGEMENT

I am deeply indebted to our supervisor of Dr Ritu Rani, Research Associate, COE-AI, IGDTUW for her advice, support, and guidance throughout the duration of the project. As our supervisor, she constantly guided me to remain focused on achieving our goal. Despite her busy schedules, she helped us at each step, discussed the problems that we faced and gave her valuable suggestions to overcome the obstacles we encountered during the implementation phase.

Rajvi Dhankher

INDEX

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Accurate and reliable prediction of stock prices has always played an increasingly important role in the stock market with investors, analyst firms, financial institutions and even regulatory boards keeping a close eye on the risks and returns as well as their volatility. Prediction of stock prices is thus one of the most researched topics methods having evolved from economists deploying mathematical, statistical and econometric models to artificial intelligence and machine learning techniques leading to creation of robust models. This report specifically explores the different machine learning techniques that are used in the prediction of stock prices. It draws a detailed analysis of 3 major machine learning approaches to prediction: Logistic Regression, SVC and XGB Classifier and explores the challenges entailed along with the future scope of work in the domain.

Chapter 1
INTRODUCTION

Predicting stock market movements through machine learning involves employing various algorithms such as Logistic Regression, Support Vector Classifier (SVC) and XGBoost Classifier.

A. Logistic Regression

Logistic Regression is a classification technique commonly used to predict the probability of an instance belonging to a particular class rather than performing traditional regression. It models the relationship between one or more independent variables and the probability of the binary outcome using the logistic (sigmoid) function.

This function ensures that the output remains between 0 and 1, mapping the linear combination of input features to a probability score. The decision boundary is set by a threshold, often 0.5, to classify instances into one of the two classes and to separate the classes. It is a simple and interpretable algorithm, suitable for situations where the relationship between features and outcome is roughly linear.

Key features include:

- Binary Classification:

Logistic regression is primarily used for binary classification problems, where the goal is to categorize instances into one of two classes.

- Probability Estimation:

Unlike other classification algorithms that directly predict class labels, logistic regression outputs the probability of an instance belonging to a certain class using the logistic (sigmoid) function.

- Sigmoid Function:

It maps any input value to one be- tween 0 and 1, which can be interpreted as the probability of the positive class. This ensures that the output remains within a certain valid range for probabilities.

- Regularization:

It can be regularized using techniques like L1 (Lasso) and L2 (Ridge) to prevent over-fitting.

- Interpretable:

Logistic regression models are interpretable and can provide insights into the significance of each feature's contribution to the classification decision.

- Assumption of Independence:

Logistic regression assumes that the observations are independent of each other and that there are no multi-collinear features.

- Maximum Likelihood Estimation:

The logistic regression model is trained using the maximum likelihood estimation method, which seeks the coefficients that maximize the likelihood of observing some particular data under the model.

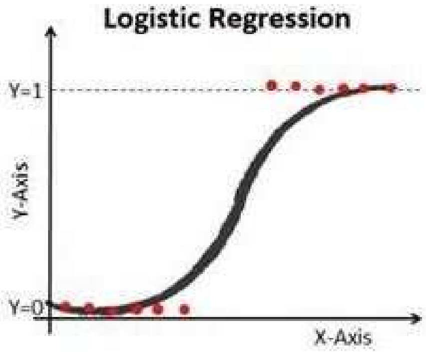


Fig 1. Logistic Regression

Overall, logistic regression is a foundational algorithm in machine learning and statistics, valued for its simplicity, interpretability, and effectiveness in scenarios where the relationship between features and outcome is relatively linear.

B. Support Vector Classifier

Support Vector Classification (SVC) is a machine learning algorithm used for binary classification tasks. It is a type of supervised learning algorithm that belongs to the family of Support Vector Machines (SVMs). It is a type of machine learning algorithm that can be used for financial market analysis. SVC can be used for regression, which is to predict the future values of financial variables, such as stock prices or exchange rates.

- Margin Maximization:

SVC aims to find the hyperplane that best separates the data points belonging to different classes while maximizing the margin between the two classes. The margin is the distance between the hyper- plane and the nearest data points from each class, and SVC seeks to find the hyperplane that maximizes this margin.

- Support Vectors:

In SVC, the data points that are closest to the hyperplane and have the most influence on its position are called “support vectors.” These are the data points that define the margin and play a crucial role in determining the decision boundary.

- Binary Classification:

SVC is primarily designed for binary classification tasks, where the goal is to separate data points into two distinct classes. However, it can also be extended to multi-class classification by using various techniques.

- Robust to Outliers:

SVC is generally robust to outliers because it focuses on the support vectors, which are the data points closest to the decision boundary. Outliers thatare far from the decision boundary have minimal impact on the model.

- Interpretability:

SVC provides a clear and intuitive geo- metric interpretation of the decision boundary, making it easy to understand how the algorithm makes predictions.

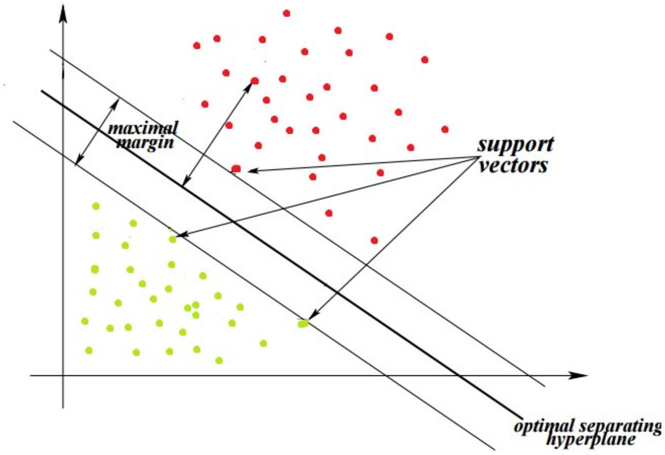


Fig 2. Support Vector Classification (SVC)

SVC is a powerful and versatile classification algorithm that can be effective in a wide range of applications. However, it may require careful tuning of hyperparameters and handling of unbalanced datasets. When used in combination with appropriate preprocessing and feature engineering techniques, SVC can deliver strong classification performance.

C. XGBoost Classifier

The XGB Classifier (extreme gradient boosting classifier) is a machine learning algorithm renowned for its performance in a wide range of classification tasks, its exceptional performance in various predictive modelling tasks and is built onthe principles of gradient boosting. It is specifically designed for and is applied to structured (can handle both numerical and categorical features) and tabular data. It implements gradient boosted decision trees designed for both speed and performance. XGB accommodates complex data relationships through its ability to handle missing values and its incorporation of regularization techniques that prevent model complexity from spiraling out of control. Furthermore, its versatility is evident as it caters to both classification and regression tasks, proving its efficacy across diverse domains. With optimized performance, scalability, and speed, the XGBoost classifier continues to be a top choicefor practitioners seeking state- of-the-art results in machinelearning applications. Key features include:

- Gradient Boosting Framework:

It builds a strong, robust and accurate predictive model by combining the predictions of multiple decision trees (ensemble modelling)

- Regularization Techniques:

It incorporates regularization(both L1 and L2) terms in complex models to help preventover-fitting by penalizing large coefficients and promotingsimpler models.

- Handling Missing Values:

It can effectively handle missing data during training and prediction. It has built in mechanisms to learn how to best treat missing values based on the available data.

- Feature Importance:

It helps us to identify which features contribute most to the model’s predictions. This is crucialfor feature selection and interpretation.

- Early Stopping:

It supports early stopping during training, which means that the process can be halted when the model’s performance on a validation dataset stops improving. This helps prevent over-fitting and saves computational resources.

- Cross Validation:

This allows us to estimate the model’s performance more reliably.

- Custom Objectives and Evaluation Metrics:

While it comes with some default objectives and evaluation metrics for classification, it allows users to define their own custom objectives and metrics to suit specific problem requirements.

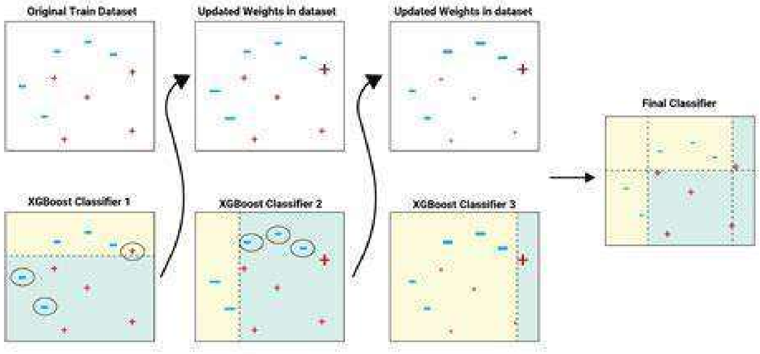


Fig 3. XGBoost Classifier

Due to its efficiency, performance, versatility, and ability to handle a variety of data types and complexities, it has become an important part of real-world applications across various industries.

Logistic Regression models the probability of a stock's price direction, relying on historical price data and other relevant indicators. However, it may struggle with capturing intricate non-linear patterns present in the stock markets. It provides interpretability. SVC, on the other hand, is capable of handling nonlinear relationships. It identifies optimal hyperplanes that best separate different classes of stock price movements. This flexibility suits markets with more complex trends and patterns. Proper kernel selection and regularization are vital for the prevention of overfitting. It adapts to non-linear trends. XGBoost Classifier, built on gradient boosting principles, excels in capturing complex dependencies between features and target variables. It can handle diverse data types, effectively incorporating indicators. Feature importance analysis aids in understanding market drivers. Hyperparameter tuning is necessary to prevent overfitting and ensure optimal performance.

In all cases, effective feature engineering remains extremely important. Extracting meaningful features from raw data enhances the model's accuracy. However, the inherent volatility and susceptibility of the stock market to external shocks poses a challenge. Employing hybrid approaches could potentially lead to more robust predictions, though the unpredictable nature of financial markets necessitates constant adaptation and evaluation of these models.

Chapter – 2
LITERATURE REVIEW

Determining the best algorithm for stock market prediction depends on various factors such as the quality and quantity of data, feature engineering, market conditions and the specific objectives of the prediction task. Each algorithm, be it logistic regression, Support Vector Classifier (SVC), and XGBoost Classifier comes with its own strengths and limitations.

Logistic Regression	**SVC**	**XGBoost Classifier**
Simple and Interpretable	Can handle high dimensional and non-linear data	High predictive accuracy
Efficient with small data sets	Classification	Handling non-linearity, outliers and anomalies
Low risk of over fitting	Deals well with outliers and noise	Feature importance
Fast training and prediction	Classification insights	Ensemble learning
Probabilistic output	Feature importance	Handling missing values
Works for linearly separable data	Outlier detection	Parallel and scalable
Good for feature selection	Model evaluation	Gradient boosting optimization

Table 1. Advantages of Logistic Regression, SVC, XGBoost Classifier

Logistic Regression	**SVC**	**XGBoost Classifier**
Linear assumption, independence	Sensitivity to hyper-parameters	Complexity
Limited to binary classification,	Computationally intensive	Hyper-parameter tuning
Sensitive to outliers	Limited interpretability	Resource intensive
Complex relationships	Over-fitting	Over-fitting risk

Table 2. Disadvantages of Logistic Regression, SVC, XGBoost Classifier

Logistic Regression:

Logistic regression is suitable when the prediction problem is binary, like whether the stock will go up or down. Itssimplicity and interpretability make it a good starting point.It can help identify the most influential features and theirrelationship with the outcome, which can be valuable in financial analysis. However, its linearity assumption might not capture complex relationships in highly dynamic markets. This is where other ensemble methods like SVC and XGB come into play.

Key advantages include:

- Simple and Interpretable:

It is straightforward to under- stand and interpret. It provides coefficients for each feature, which show the direction, strength of their influenceon the prediction and the magnitude of their impact onthe probability of a particular outcome.

- Efficient with small data-sets:

It works well with small data-sets and doesn’t require a large number of computational resources.

- Low risk of over-fitting
- Fast Training and Prediction:

Training Logistic Regression models is relatively quick, making it suitable for real-time or time-sensitive applications.

- Probabilistic Output:

Instead of a simple class prediction, Logistic Regression provides a probability score. This canbe useful when ranking or prioritizing predictions.

- Works for Linearly Separable Data:

When the classescan be separated by a straight line (in 2D), Logistic Regression can work well.

- Good for Feature Selection:

You can assess the im- portance of features by examining their corresponding coefficients. Can handle binary and multi-class problems: While originally designed for binary classification, it can beextended for multi-class tasks using various techniques.

Limitations include:

- Linear Assumption:

Logistic Regression assumes a linear relationship between features and the log-odds of the outcome. It might struggle when the relationship is more complex.

- Limited to Binary Classification:

Logistic Regression is designed for binary classification tasks and might not be the best choice for multi-class problems. It is designed for binary classification problems, predicting a probability of one of two outcomes. While this can be useful for certain stock market predictions, it's not well-suited for forecasting continuous variables like stock prices.

- Sensitive to Outliers:

Outliers can disproportionately in- fluence the coefficients and predictions of the model.

- Complex Relationships:

If the relationships between features and the outcome are nonlinear or involve interactions, Logistic Regression might not perform well.

- Assumption of Independence:

It assumes that features are independent of each other, which might not hold true in many real-world situations.

- Imbalanced Classes:

Logistic Regression can struggle when dealing with imbalanced classes, where one class has significantly fewer samples than the other.

- Data Imbalance:

Imbalanced data-sets, where one class significantly outweighs the other, can affect the model's performance and lead to biased predictions. Logistic regression might struggle to handle such imbalances effectively.

- Over-fitting and Under-fitting:

Achieving the right balance between model complexity and generalization can be challenging. Logistic regression might under-fit when relationships are complex and over-fit when the model becomes too intricate.

- Limited Learning Capacity:

Logistic regression is a simple algorithm with limited learning capacity. It might struggle to capture intricate and nuanced patterns that more complex algorithms like gradient boosting or deep learning models can better handle.

Support Vector Classifier (SVC):

SVC is useful when there is a need to capture non-linear relationships between features and stock price movements. It can handle both binary and multi-class classification tasks, making it versatile for more complex predictions. However, SVC's performance depends on the selection of the kernel function and hyper-parameters, which can require careful tuning.

Key advantages include:

- Can handle high-dimensional and nonlinear data.
- Classification:

Can provide a clear margin of separation between different classes, which can reduce the risk of the wrong type of classification.

- Deals well with outliers and noise:

SVC can be robust to outliers and noise, which can improve the accuracy and stability of the model.

- Classification Insights:

SVC is a robust classification algorithm that can handle various types of data distributions and can provide insights into how well you can classify data into different categories.

- Feature Importance:

SVC can help you identify the importance of different features in making classification decisions. This information can be valuable for feature selection, which is often a part of EDA, as it helps in narrowing down the focus on the most relevant features.

- Outlier Detection:

While SVC is not typically used for outlier detection, it can indirectly assist in identifying potential outliers in your dataset by revealing data points that lie near the decision boundary.

Outliers can be crucial to identify during EDA, as they can impact the distribution and summary statistics of your data.

- Model Evaluation:

Once you've built an SVC model, you can use various evaluation metrics (e.g., accuracy, precision, recall, F1-score) to assess the model's performance. Understanding these metrics can provide valuable insights into the quality of your data and the effectiveness of the classification task, which can be part of the EDA process.

Limitations include:

- Sensitivity to Hyper-parameters:

SVC has hyper- parameters that need careful tuning for optimal performance. The choice of the kernel (linear, polynomial, RBF, etc.), regularization parameter (C), and other hyper-parameters can significantly affect the model's results. Finding the right set of hyper-parameters can be time- consuming and may require expertise.

- Computationally Intensive:

SVC can be computationally expensive, especially when dealing with large data-sets or high-dimensional feature spaces. Training an SVC model can take a considerable amount of time and resources, which can be a limitation in time-sensitive or resource- constrained situations.

- Limited Interpretability:

While SVC is a powerful classification algorithm, the decision boundary it creates can be complex and less interpretable compared to simpler models like logistic regression. Understanding the relationship between features and predictions can be more challenging with SVC.

- Overfitting:

SVC can suffer from overfitting, which is when the model fits too well to the training data and fails to generalize to new data.

XGBoost Classifier:

XGBoost is often favored for stock market prediction due to its ability to handle intricate relationships in the data and its robustness against over-fitting. Its ensemble of decision trees can capture non-linearity's and interactions in the data, which is crucial in the stock market where factors influencing prices are interconnected. XGBoost's ability to incorporate multiple features and its feature importance analysis provide insights into the predictive process.

Key advantages include:

- High Predictive Accuracy:

XGBoost is known for its predictive accuracy. It performs well even on complex data-sets, which is valuable in capturing the intricate patterns and relationships within stock market data.

- Handling Non-Linearity:

Stock market data often exhibits non-linear relationships and patterns. XGBoost is capable of modeling non-linear dependencies between features and target variables, making it well-suited for capturing the complex dynamics of financial markets.

- Feature Importance:

XGBoost provides insights into feature importance, allowing you to identify which input variables contribute the most to making accurate predictions. This is crucial in stock market prediction, where understanding which factors influence market movements can be highly valuable.

- Ensemble Learning:

XGBoost uses an ensemble of decision trees, which allows it to learn from multiple weak models and combine their strengths to produce a stronger overall prediction. This ensemble approach can help mitigate over-fitting and improve generalization to unseen data.

- Handling Missing Values:

XGBoost can handle missing values in the dataset, reducing the need for extensive pre-processing of the data. In financial data-sets, missing values are common due to irregular trading times and incomplete data.

- Parallel and Scalable:

XGBoost is designed to efficiently utilize multi-core processors and can be parallelized, making it suitable for handling large data-sets. This scalability is important in financial markets where data can be voluminous and high-frequency.

- Outliers and Anomalies Handling:

XGBoost is robust to outliers and anomalies in the data. In financial markets, sudden price movements or unexpected events can lead to outliers, and XGBoost's resilience to such data points is advantageous.

- Gradient Boosting Optimization:

XGBoost employs a gradient boosting framework that optimizes the model's performance in an iterative manner. This allows the algorithm to focus on correcting the errors made by previous iterations, leading to gradual improvement in predictive accuracy.

Limitations include:

- Complexity:

XGBoost is a complex ensemble method that consists of many decision trees. This complexity can make it challenging to interpret the model's behaviour, understand the relationships between features and pre- dictions, and gain insights from the EDA phase.

- Hyper-parameter Tuning:

To achieve optimal performance, XGBoost requires careful tuning of hyper- parameters such as learning rate, tree depth, number of estimators (trees), and regularization parameters. Finding the right set of hyper-parameters can be time-consuming and may require expertise.

- Resource Intensive:

Training XGBoost models, especially with a large number of trees and features, can be computationally intensive and require substantial computational resources. This can be a limitation in terms of time and hardware constraints.

- Over-fitting Risk:

XGBoost is prone to over-fitting, especially when the model complexity is not properly controlled. Over-fitting can lead to a model that performs well on the training data but poorly on unseen data, which can be detrimental in stock market prediction where generalization is crucial.

Despite all the advantages, it's important to note that stock market prediction is a challenging task due to its inherent volatility and unpredictability. In practice, many successful stock market prediction models use machine learning ensembles that combine multiple algorithms to harness their individual strengths. Additionally, the effectiveness of any algorithm depends on the quality and relevance of the features used for prediction. The dynamic and volatile nature of the stock market makes prediction a challenging task, and while these algorithms can provide insights, they may not consistently produce accurate predictions due to the inherent uncertainties involved.

Considering domain expertise and incorporating external data sources can further enhance the predictive capabilities of these algorithms in the context of stock market prediction. Rigorous testing, continuous validation, and consideration of fundamental and external factors are essential regardless of the chosen algorithm. Finally, experimentation and thorough validation are key to determining the most suitable approach for stock market prediction.

Chapter – 3

OBJECTIVE

Objective of this study is how to predict a signal that indicates the future performance of a stock (specifically Uniqlo’s stock) by using 3 major machine learning techniques namely, Logistic Regression, Support Vector Classifier (SVC), and XGB Classifier. We shall further test deployment machine learning technique which would result in the most accurate model.

Chapter – 4

METHODOLOGY AND IMPLEMENTATION

In this report, we are focusing on developing a model suitable to accurately predict stock prices.

For the first part, to do exploratory data analysis (EDA) on the chosen data-set (of the stock prices of Uniqlo from the year 2012-16) we chose a distribution plot. A distribution plot in Exploratory Data Analysis (EDA) is a graphical representation that shows the distribution of a dataset’s values. We chose this because it helped us understand the underlying characteristics of a dataset, including the central tendency, spread, how skewed the data is, and the potential outliers. Distribution plots are known to be crucial for gaining insights into the shape of the data and identifying patterns or anomalies.

For the second part, we divided the data-set into the training and testing parts. We then deployed the 3 machine learning techniques on the training data and subsequently tested it on the test data to see the accuracy of the models.

Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from xgboost import XGBClassifier
from sklearn import metrics

! pip install chart_studio

from chart_studio.plotly import iplot
import plotly.graph_objs as go
import warnings
```

```
warnings.filterwarnings('ignore')
%pylab inline

!pwd

from google.colab import files

uploaded = files.upload()
datatrain = pd.read_csv("/content/UniqloTrainingstocks1216.csv")

datatrain.head()

chart1 = go.Figure(data=[go.Candlestick(x=datatrain['Date'],
                open=datatrain['Open'],
                high=datatrain['High'],
                low=datatrain['Low'],
                close=datatrain['Close'])])

chart1.show()

#EDA
features = ['Open', 'High', 'Low', 'Close', 'Volume']

plt.subplots(figsize=(20,10))

for i, col in enumerate(features):
  plt.subplot(2,3,i+1)
  sns.distplot(datatrain[col])
plt.show()

# From the distribution plots we can see that the Volume data is left skewed and across OHLC, there are two major peaks

plt.subplots(figsize=(20,10))
```

```
for i, col in enumerate(features):
  plt.subplot(2,3,i+1)
  sns.boxplot(datatrain[col])
plt.show()

# From the boxplots we can see that the Volume data has outliers

splitted = datatrain['Date'].str.split('-', expand=True)

datatrain['Year'] = splitted[0].astype('int')
datatrain['Month'] = splitted[1].astype('int')
datatrain['Day'] = splitted[2].astype('int')

datatrain['is_quarter_end'] =
  np.where(datatrain['Month']%3==0,1,0)

data_grouped = datatrain.groupby('Year').mean()
plt.subplots(figsize=(20,10))

for i, col in enumerate(['Open', 'High', 'Low', 'Close']):
  plt.subplot(2,2,i+1)
  data_grouped[col].plot.bar()
plt.show()

# We can see from the quarterly data that Uniqlo's stock price
  increases by almost 2.5x from 2012 - 2015 and then reduced in
  2016

datatrain.groupby('is_quarter_end').mean()

# Prices and volumes traded are higher at the end of a quarter

#Modelling
datatrain['open-close'] = datatrain['Open'] - datatrain['Close']
datatrain['low-high'] = datatrain['Low'] - datatrain['High']
```

```
  datatrain['target'] = np.where(datatrain['Close'].shift(-1) >
   datatrain['Close'], 1, 0)


  features = datatrain[['open-close', 'low-high',
   'is_quarter_end']]

  target = datatrain['target']


  scaler = StandardScaler()

  features = scaler.fit_transform(features)


  X_train, X_test, Y_train, Y_test = train_test_split(

      features, target, test_size=0.1, random_state=2022)

  print(X_train.shape, X_test.shape)


  models = [LogisticRegression(), SVC(

    kernel='poly', probability=True), XGBClassifier()]


  for i in range(3):

    models[i].fit(X_train, Y_train)


    print(f'{models[i]} : ')

    print('Training Accuracy : ', metrics.roc_auc_score(

      Y_train, models[i].predict_proba(X_train)[:,1]))

    print('Validation Accuracy : ', metrics.roc_auc_score(

      Y_test, models[i].predict_proba(X_test)[:,1]))

    print()


# While XBGClassifier has the highest accuracy for training data,
the large difference between the accuracy for training and test data
shows it's prone to overfitting for which reason Logistic Regression
is the best method to be used here
```

Chapter – 5

RESULT DISCUSSION

We see that Uniqlo's share prices have seen fluctuations over the years.

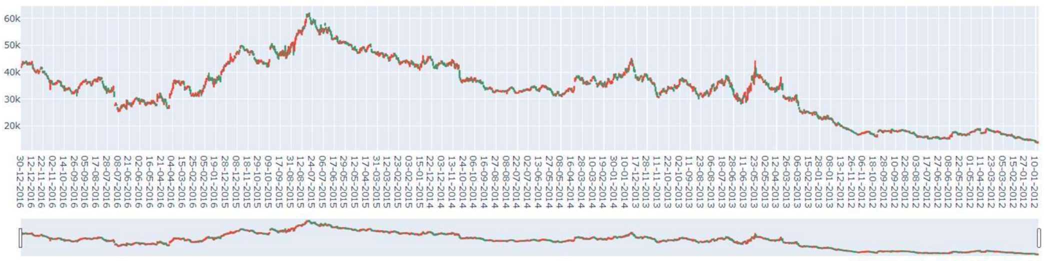


Fig 4. Uniqlo's stock price trend 2012 - 2016

From the distribution plots we can see that the Volume data is left skewed and across OHLC (open, high, low, close data points) and there are two major peaks around the center.

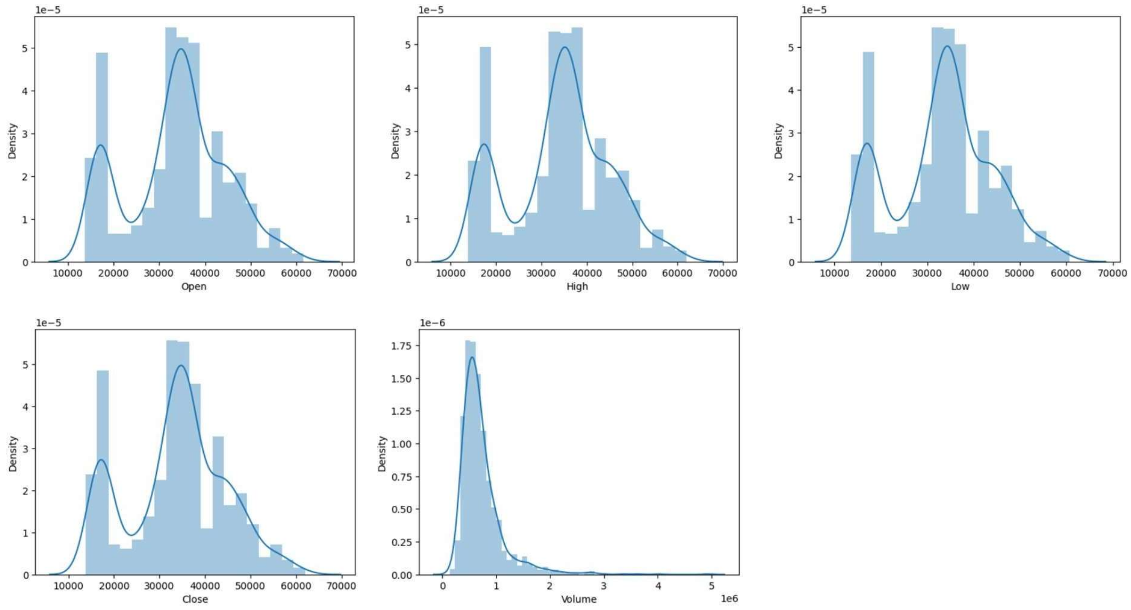


Fig 5. Distribution plots

From the box-plots, we see that the volume data has outliers (these are data points that significantly differ from the majority of the observations in a dataset, data points that are notably distant from the central tendency of the data distribution and can have a substantial impact on the summary statistics, visualizations, and overall analysis).

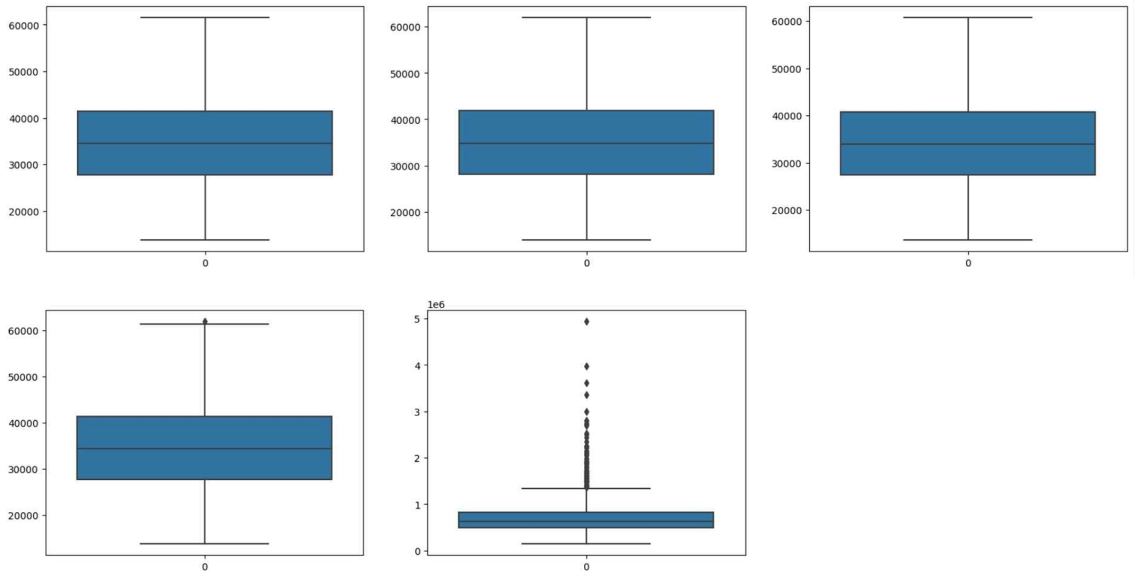


Fig 6. Box plots

From the quarterly data, we see that Uniqlo’s stock price increased by almost 2.5x times fromthe years 2012-15 and they then reduced in the year 2016. Prices and volumes traded are higher towards and at the end of a quarter.

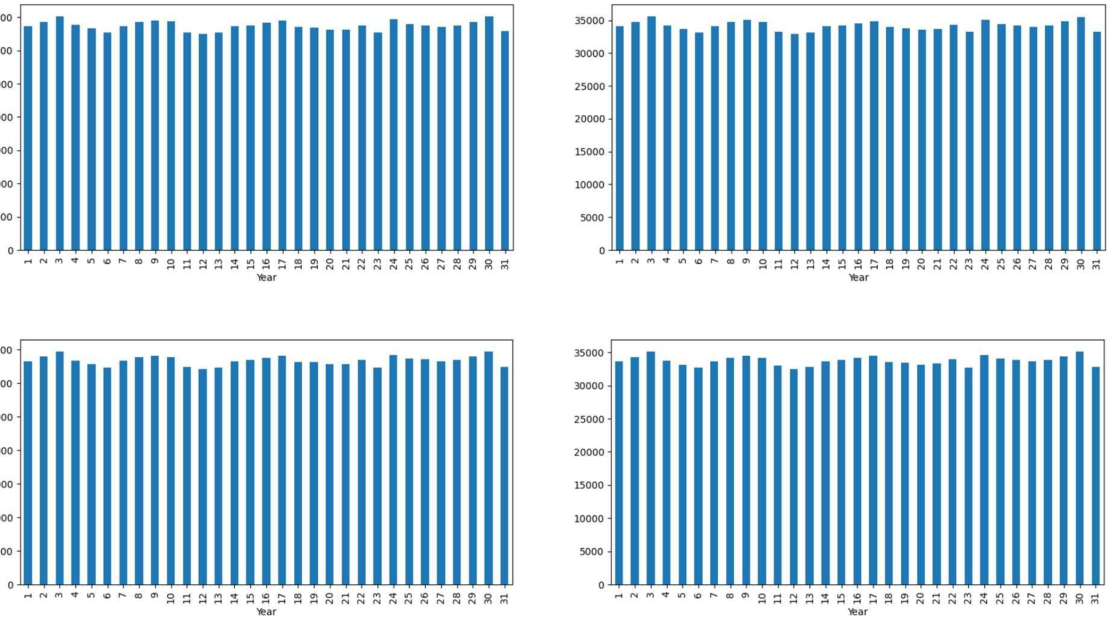
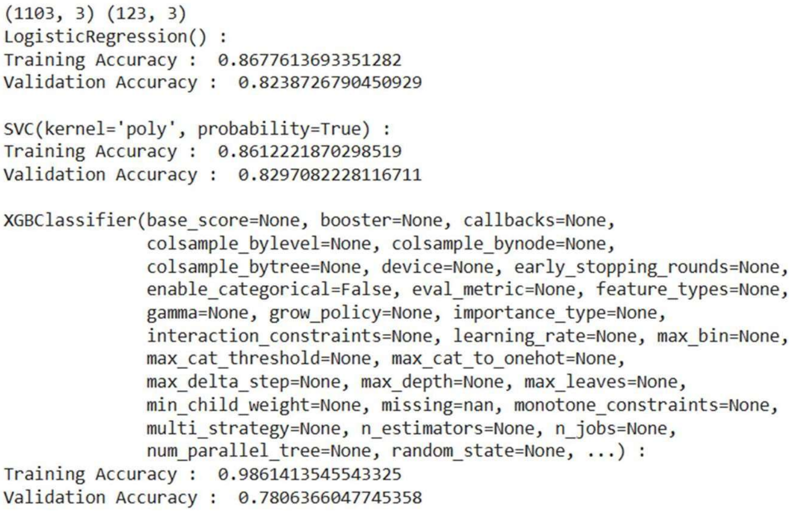


Fig 7. Bar charts for quarterly data

After doing the modelling part of our analysis, we saw that while XGB Classifier has the highest accuracy for the training data, there was a large difference in its accuracy for the testing and the training data. This large difference between the accuracy for training and testing data shows that it is prone to over-fitting for which reason, keeping all the advantages and limitations in mind, Logistic Regression would be the best method to use here (as it handles outliers better than SVC and the accuracy obtained from them is very similar).



```
(1103, 3) (123, 3)
LogisticRegression() :
Training Accuracy :  0.8677613693351282
Validation Accuracy :  0.8238726790450929

SVC(kernel='poly', probability=True) :
Training Accuracy :  0.8612221870298519
Validation Accuracy :  0.8297082228116711

XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None,
              num_parallel_tree=None, random_state=None, ...) :
Training Accuracy :  0.9861413545543325
Validation Accuracy :  0.7806366047745358
```

Fig 7. Accuracy across Logistic Regression, SVC and XGB Classifier

Chapter – 6

CONCLUSION AND FUTURE SCOPE

From the study, we can conclude that Logistic Regression would be the best machine learning technique for use cases such as stock market prediction. In order to solve its limitation of over-fitting, we can deploy several methods. Some of these are as listed below:

- Using a sufficiently large data-set

The larger the data-set, the less likely the model will over-fit.

- Test-Train Split

Making sure that the class imbalance is equivalent between the training set and the testing set by using stratified sub-sampling.

- Parameter Fine-Tuning

Tuning the parameters of XGB Classifier to avoid over- fitting. Some of the parameters that can be changed include the ratio of features used (i.e., columns used), the ratio of the training instances used (i.e., rows used), and the maximum depth of a tree.

- Early Stopping

Using early stopping to prematurely stop the training of the classifier at an optimal epoch. It is important to note that trying to avoid over-fitting might lead to under-fitting, where we regularize too much and failto learn relevant information. Therefore, it is crucial to find a balance between avoiding over-fitting and under-fitting.

Future scope of this study can include adding other parameters such as the financial ratios, market multiples, etc. to increase accuracy of the model. The scope can also be expanded to include sentiment analysis of public and analyst firms' commentary as well as predicting corporate performance structure in its entirety. Additionally, other more advanced algorithms and concepts can be explored such as genetic algorithms for optimizing network architecture and fuzzy logic to account for uncertainty produced, helping the model become more robust.

REFERENCES

[1] Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis.

[2] Effect of public sentiment on stock market movement prediction during the COVID-19 outbreak.

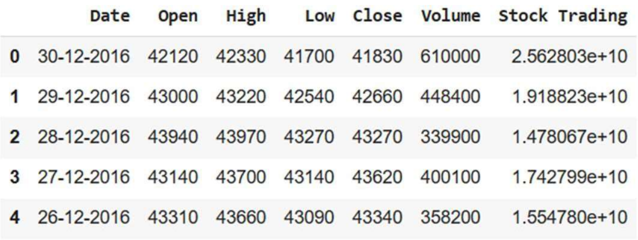
[3] Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis.

[4] Machine Learning Stock Market Prediction Studies: Review and Re- search Directions.

[5] Machine Learning Approaches in Stock Price Prediction: A Systematic Review.

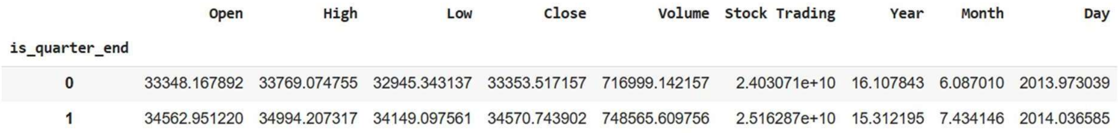
APPENDIX

Appendix 1: Uniqlo's stock data snippet



	Date	Open	High	Low	Close	Volume	Stock Trading
0	30-12-2016	42120	42330	41700	41830	610000	2.562803e+10
1	29-12-2016	43000	43220	42540	42660	448400	1.918823e+10
2	28-12-2016	43940	43970	43270	43270	339900	1.478067e+10
3	27-12-2016	43140	43700	43140	43620	400100	1.742799e+10
4	26-12-2016	43310	43660	43090	43340	358200	1.554780e+10

Appendix 2: Feature Engineering



	Open	High	Low	Close	Volume	Stock Trading	Year	Month	Day
is_quarter_end									
0	33348.167892	33769.074755	32945.343137	33353.517157	716999.142157	2.403071e+10	16.107843	6.087010	2013.973039
1	34562.951220	34994.207317	34149.097561	34570.743902	748565.609756	2.516287e+10	15.312195	7.434146	2014.036585