A Dissertation on

INVESTMENT PREDICTION

Submitted in partial fulfillment of the

requirement for the award of the degree

of

MASTERS OF SCIENCE

in

Computer Science (BIG DATA ANALYTICS)

Submitted by

Rishiraj Singh Chauhan (2021MSBDA034)

Under the Guidance of

Avnish Yadav
Junior Data Scientist (iNeuron.ai. Pvt. Ltd)

&

Dr. Pritpal Singh
Assistant Professor
Department of Data Science & Analytics

Department of Data Science and Analytics

School of Mathematics, Statistics, and Computational Science

CENTRAL UNIVERSITY OF RAJASTHAN

August 2023

DECLARATION

I certify that

a. The work contained in the dissertation has been done by myself under the supervision of my supervisor.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

d. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the dissertation and giving their details in the references.

e. Whenever I have quoted written materials from other sources and due credit is given to the sources by citing them.

Date:19`August`2023
Place: Central University of Rajasthan

Student Name: Rishiraj Singh
Chauhan
Regd. No.: 2021MSBDA034

iNeuron Intelligence Pvt Ltd

17th Floor Tower A, Brigade Signature Towers, Sannatammanahalli,
Bengaluru, Karnataka - 562129.

Provisional Experience Letter

DATE: 18th August 2023

TO WHOM IT MAY CONCERN

This is to certify that **Mr. Rishiraj Singh Chauhan** is currently engaged in a valuable and enriching internship program under my guidance from 18th April 2023 in **Investment Prediction** at INEURON INTELLIGENCE PRIVATE LIMITED. During his internship programme with us, He is demonstrating exceptional skills with a self- motivatedattitude to learn new things and implement them end to end with all of our mentioned industrial standards. Also his performance is excellent.

Our wishes are always with his for future endeavours.

Regards,

Avnish Yadav
Junior Data Scientist at iNeuron.ai

DEPARTMENT OF DATA SCIENCE AND ANALYTICS
CENTRAL UNIVERSITY OF RAJASTHAN, INDIA

Date:19-08-2023

CERTIFICATE

This is to certify that the project titled “**INVESTMENT PREDICTION”** is a record of the bonafide work done by **RISHIRAJ SINGH CHAUHAN** (2021MSBDA034) submitted in partial fulfillment of the requirements for the award of the Degree of Master of Science (M.Sc.) in Computer Science (Big Data Analytics) of Central University of Rajasthan, during the academic year 2022-23.

Dr. Pritpal Singh
Supervisor,
DEPARTMENT OF DATA SCIENCE AND ANALYTICS
Central University of Rajasthan

Dr. Vidyottama Jain
*HOD,(*DEPARTMENT OF DATA SCIENCE AND ANALYTICS*)*
Central University of Rajasthan

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor D.C. Sharma, Dean/Director of the (Central University of Rajasthan). He motivated us and created a wonderful learning environment that has improved our academic path. His academic excellence has made a significant impact on our educational experiences.

I want to express my gratitude to Dr. Vidyottama Jain, head of (Department of data science & Analytics). Her advice and suggestions have played a significant role in how my project has turned out.

I want to express my sincere gratitude to my project mentor, Dr. Pritpal Singh (Assistant Professor, Department of Data Science & Analytics), for his dedication, support, and insightful advice throughout the course of the year, all of which were important to the completion of my project. Your assistance has been highly valued, and your guidance has been a precious gift.

I would like to convey my deepest appreciation to my family and friends for their love, understanding, & constant encouragement. This success is a result of both my own efforts and their faith in Me.

In conclusion, the accomplishment of the completion of this project is a team effort, and I really appreciate the amazing contributions provided by each and every person involved.

Rishiraj Singh
Chauhan
(2021MSBDA034)
Semester IV
Department of Data Science & Analytics
Central University of Rajasthan

ABSTRACT

Predicting investment outcomes has become important in today's rapidly shifting financial environment, which is characterized by complex dynamics and increased uncertainty. The important area of investment forecasting is the focus of this research, which addresses the urgent demand for precise predictions to enable investors to make wise judgments. The main goal of the project is to use sophisticated analytical methods to create a prediction model that gives investors information about upcoming market movements and enables them to make wise decisions.

The project's technique is based on a solid data-driven methodology. The groundwork for analysis is laid by compiling a comprehensive dataset that includes historical market data, economic indicators. The prediction model is painstakingly built to capture complex temporal relationships and hidden patterns that affect market behavior. It makes use of machine learning technologies.

The results of the prediction model demonstrate a noteworthy degree of accuracy in predicting investment performance. Because of the model's innate capacity to comprehend intricate market dynamics, investors are given a discriminating instrument to recognize and comprehend current market patterns, thereby reducing risks and maximizing returns on investments. The underlying value of this research comes in its ability to transform and advance investing techniques, enabling investors to make decisions that are more secure and fruitful.

The success of this project is largely due to crucial tools and technologies that enable effective execution. The basic framework for many tasks, such as data preprocessing, algorithm execution, and visualization, is provided by Python, a flexible programming language. Effective data manipulation, analysis, and model training are made possible by essential libraries like pandas, NumPy, and scikit-learn. The use of Jupyter Notebook encourages teamwork by giving the research team a shared workspace where they may iteratively improve, implement, and document their approaches.

In conclusion, this effort emphasizes how important accurate investment forecasting is in the modern financial environment. The project successfully produces a predictive model ready to alter investment decision-making by fusing the capabilities of machine learning and advanced data analysis. The project's ability to provide concrete insights for investors is strengthened by the seamless interaction of approaches and instruments, opening-up an ocean of more informed, strategic, and successful financial decisions.

LIST OF TABLES

LIST OF FIGURES

Contents

CHAPTER 1 INTRODUCTION

1.1 Introduction to work done/ Motivation (Overview, Applications &Advantages)

The stock market has always been closely watched by investors due to its high risk and high potential reward, and stock forecasting has always been a research topic of great interest to researchers. Additionally, the stock market is a crucial component of any nation's financial system. It serves as a reflection of how well the national economy is doing and has a significant influence on how well it is doing. Although the issue of predictability of stocks has always been controversial, the study of stock forecasts still helps us understand the laws of some market changes and development. With the advancement of science and technology, a large amount of financial data has been retained providing a solid data foundation for the analysis of the stock market; at the same time, the continuous development and updating of algorithms has provided a powerful tool for people to analyze the stock market.

As an important part of a country's economy, the stock market provides a financing and investment environment for the country's companies and investors. Predicting the future performance of the stock market can not only provide investors with investment advice, but also help companies formulate financing plans, there by promoting the healthy development of the economy. In addition, using the portfolio theory along with a stable investment portfolio built on forecast results can help investors increase their investment returns. As a result, it is a very important problem for research on investment portfolio methods and stock market forecasting.

Investors usually adjust the allocation of investment assets to reduce their own decision-making risks, this makes it very important for investors to predict the price of stocks or other financial assets. Machine learning algorithms can help in prediction of any stock's price. This study demonstrates how different classification algorithms can forecast the value of any stock. Different classification algorithms such as Logistic Regression, Support Vector Classifier, XG Boost, and KNN have been tested and compared to predict a better outcome of the model.

The motivation for this project is that the predictive models can offer a systematic and data-driven approach to address the challenges posed by modern investment scenarios. By applying sophisticated algorithms and data analysis techniques. These models have the potential to identify hidden patterns, correlations, and trends within large and diverse datasets. This allows investors to make more informed decisions that go beyond conventional financial metrics.

Traditional investment decisions frequently depend on historical data, financial indicators, and market trends. The ability to correctly forecast investment outcomes has consequently become crucial.

Models for predicting future investments play a key role in algorithmic trading techniques. This is especially useful when the market is unpredictable means market upswings or downturns. Investment forecasts offer perceptions into long-term patterns, assisting investors to make smart choices for retirement planning, paying for higher education, and other financial goals. Investors can use strategies for hedging to balance probable losses in one investment with gains in another, minimizing overall risk exposure, by correctly predicting market developments.

Making Informed Decisions: By providing investors with data-driven insights, investment prediction enables them to make decisions that are well-informed and are based on a thorough understanding of market dynamics.

Increased Returns: Accurate investment forecasts make it easier to spot high chances to succeed, which boosts investment returns and improves portfolio efficiency.

Risk reduction: By foreseeing market changes, investors can take precautions to cut losses and protect investment under challenging market conditions.

Competitive Advantage: Using predictive models gives investors the opportunity to capitalize on new possibilities and remain ahead of market trends.

1.2 Project Statement / Objectives of the Project

The goal of this project is to create a reliable investment forecasting model that makes use of modern machine learning methods to provide precise predictions of market trends and asset performance. The goal of the research is to develop a prediction framework that considers a broad range of elements, including previous market information. The model attempts to find hidden patterns and connections that influence investment outcomes by analyzing these many inputs. Our objective is to provide a simple user experience that enables investors to input real-time data and receive prompt recommendations for making wise investment decisions.

1.3 Organization of Report

Chapter 1 Introduction
The project's context, goals, and relevance will all be covered in this chapter. There will be a discussion of the project's issue statement and explanation, as well as an overview of the report's format.

Chap. 2: Background Information
2.1 Conceptual Overview (Used Theories/Concepts)
This section will cover the key concepts and theories behind the project. The conceptual frameworks on which the project depends will be made clear, along with the concepts needed to understand the following chapters. Any specialized terminology or theoretical frameworks will be thoroughly explained.

2.2 Technologies Used
This section will look over the various technologies used in the project. This category may include any technological components like as hardware, software, frameworks, programming languages, and others that are crucial to the project's execution. We'll go into further depth regarding the selection criteria for these technologies.

Chapter 3 Methodology
3.1 Detailed Methodology to Be Adopted
This section will provide a detailed explanation of the method that is used to achieve the project's objectives. It will provide strategies, tactics, and procedures used. If any existing methods are changed, they will be described and justified in this section.

3.2 Block and Circuit Diagrams
Detailed circuit configurations and block diagrams are provided in this portion. These graphic representations of the project's design and structure will help the reader comprehend the relationships and flow between its numerous components.

Chapter 4 Implementation

Chapter 4.1 Modules,
The several components and modules that make up the project's execution are described in this section. The functions of each module will be discussed, along with how and why they benefit the project overall. The implementation problems, challenges, and solutions for these modules will all be discussed.

4.2 Model
This section will present the project's prototype as it has been created. In order to provide readers a better idea of the finished product, it will also include information about the hardware and software components of the prototype.

Chapter 5 Results and Analysis
The project's implementation results will be presented in this chapter. In regard to the project's goals, it will discuss and analyze any data, measurements, or results obtained. Any differences or unexpected results will

be discussed, along with any impacts they may have affect the result.

Chapter 6 Conclusions and Future Directions

6.1 Results

Here, the main conclusions from the project's execution will be presented. It will look at how well the objectives were met as well as the importance of the project's findings. All challenges encountered will be discussed, along with the lessons learned.

6.2 Future Scope of Work

On the basis of the existing project, this section will identify prospective directions for future workand development. It can contain topics that weren't covered, improvements to the current prototype, or brand-new uses that could result from the project's discoveries. It will be addressed ifpursuing these future routes will have any advantages or disadvantages.

CHAPTER 2
Back-Ground Study

2.1 Conceptual Overview (Concepts/Theory Used)

The project's goal of predicting the price of stocks involves several key concepts and theoriesfrom both finance and data science domains.

A conceptual summary is given below:

Data Collection:

For an investment prediction project, historical data on the investment, including the price, volume, and other relevant details are collected. You can get this information from companies that offer financial data, like Bloomberg or Yahoo Finance. The data must be accurate, well-organized, and cover a sufficient amount of time to identify the pertinent trends. Additionally, the data ought to be an accurate reflection of the state of the market.

Data cleaning:

The first step is to clean the data to remove any errors or inconsistencies. This may involve removing duplicate data, correcting typos, and filling in missing values.

Data exploration:

Once the data has been cleaned, it is important to explore the data to get a better understanding of it. This may involve plotting the data, calculating summary statistics, and looking for patterns.

Feature Engineering:

The term "feature engineering" refers to the process of choosing and creating important input variables for the predictive model. Financial indicators and relative strength indicators are examples of meaningful features that can be extracted from raw data using concepts from statistics and domain expertise. Creating features from the data is the next step. This involves transforming the data into a format that the prediction model can use. For instance, the volume data could be divided into bins or the price data could be normalized.

Model Selection:

Choosing a prediction model is the fourth step. There are numerous prediction models available, each with unique advantages and disadvantages. The specific data and the desired accuracy will determine the best model for a given investment.

Model training:

After choosing a prediction model, it must be put to use by being trained on the past data. In order for the model to recognize patterns in the data, the data must be fed to it.

Testing and Validation:

The model needs to be tested on new data after it has been trained to see if it can make generalizations. Cross-validation techniques and metrics like precision, accuracy, recall, and F1 score are used to evaluate the model's efficacy.

Continuous Learning:

By allowing the model to learn from new data and adjust to market changes, it can increase the prediction accuracy. It may reduce the chance of overfitting, a problem that can arise when a model is trained on a large amount of data. It may improve the model's resistance to market changes like new regulations or economic shocks.

2.2 Technologies Involved

The project aiming to predict the risk of thyroid disease involves a combination of medical knowledge and data science techniques. To achieve this, the following technologies, tools, and frameworks are likely to be used:

Python:
Python is a versatile programming language commonly used in data science and machine learning projects due to its rich libraries and frameworks for data manipulation, analysis, and modelling.

Libraries for Data Manipulation and Analysis:
Pandas: Used for cleaning, preprocessing, and data manipulation.
NumPy: Used for array operations and numerical computations.

Data Visualization Libraries:
Matplotlib: For creating static, interactive, and animated visualizations.
Seaborn: Built on top of Matplotlib, specialized for statistical visualizations.

Machine Learning Libraries: Scikit-Learn is a popular library for machine learning in Python. It provides a wide range of algorithms for classification tasks, which can be used to train models to predict the stock.

Jupyter Notebooks: Jupyter provides an interactive environment for developing and sharing code, visualizations, and explanations. It's a valuable tool for documenting the analysis process and results

Version Control: Platforms like Git and services like GitHub or GitLab can help in versioning code, collaborating with team members, and maintaining a record of changes.

The following Table Summarizes the Technologies Used in This Project

Table 2.1 Technologies Involved

Technology	Purpose
Programming Language (Python)	Implementation of algorithms and model
Data Processing (Pandas)	Preprocessing and Analysis of data
Machine Learning (Scikit-learn)	Model Development
Model Evaluation & Visualization (Matplotlib, Seaborn)	Visualization and Analysis
Data Collection (YFinance)	Collecting the Finance Data
Jupyter Notebook(EDA)	Provides an Interactive Environment
Version Control (Git, GitHub)	Code Management and Collaboration

CHAPTER 3 METHODOLOGY

3.1 Detailed Methodology that Will be Adopted:

Step 1. Data Collection:

Data Sources:

- Subscription-based financial data providers provide historical and current market data, such as stock prices, ratings, economic indicators, and company financials.
- Public Financial APIs: Although they come with some restrictions, public APIs like Alpha Vantage, Yahoo Finance, and Quandl provide free access to financial data.
- We use public financial APIs Yahoo Finance for this project

Data Type:

- We use Time-Series data for this project. Time series data include historical and real-time market information as well as stock prices, trading volumes, and other market indicators.

Data Collection Techniques

- APIs: Numerous news and financial sources provide APIs that enable programmatic data access.
- Web scraping: Software like Beautiful Soup or Scrapy can be used to extract data from websites.
- Manual file downloads are possible from financial platforms or websites.
- Financial data providers offer data that can be directly purchased or subscribed to.
- But for this project we use 'YFINANCE API'.

Step 2. Data Preprocessing:

Before developing predictive models, data preprocessing is a crucial step in investment prediction to ensure the accuracy, consistency, and relevance of the data. The essential preprocessing steps for investment prediction projects are described in this step-by-step manual. Make the data more accurate by managing missing numbers, outliers, and irregularities. To guarantee uniform scaling, normalize or standardize numerical properties.

a. Cleaning of Data:

- Handle missing values: Depending on the type of data, impute missing values using methods like mean, median, forward-fill, or backward-fill.
- Find outliers and deal with them: Determine whether to remove, transform, or treat separately any outliers that may skew predictions.

b. Data Transformation:

- Scale numerical features to ensure they have comparable ranges, preventing dominance by larger values, to normalize or standardize data.
- Log or power transformations: Improve the assumptions of some algorithms by transforming

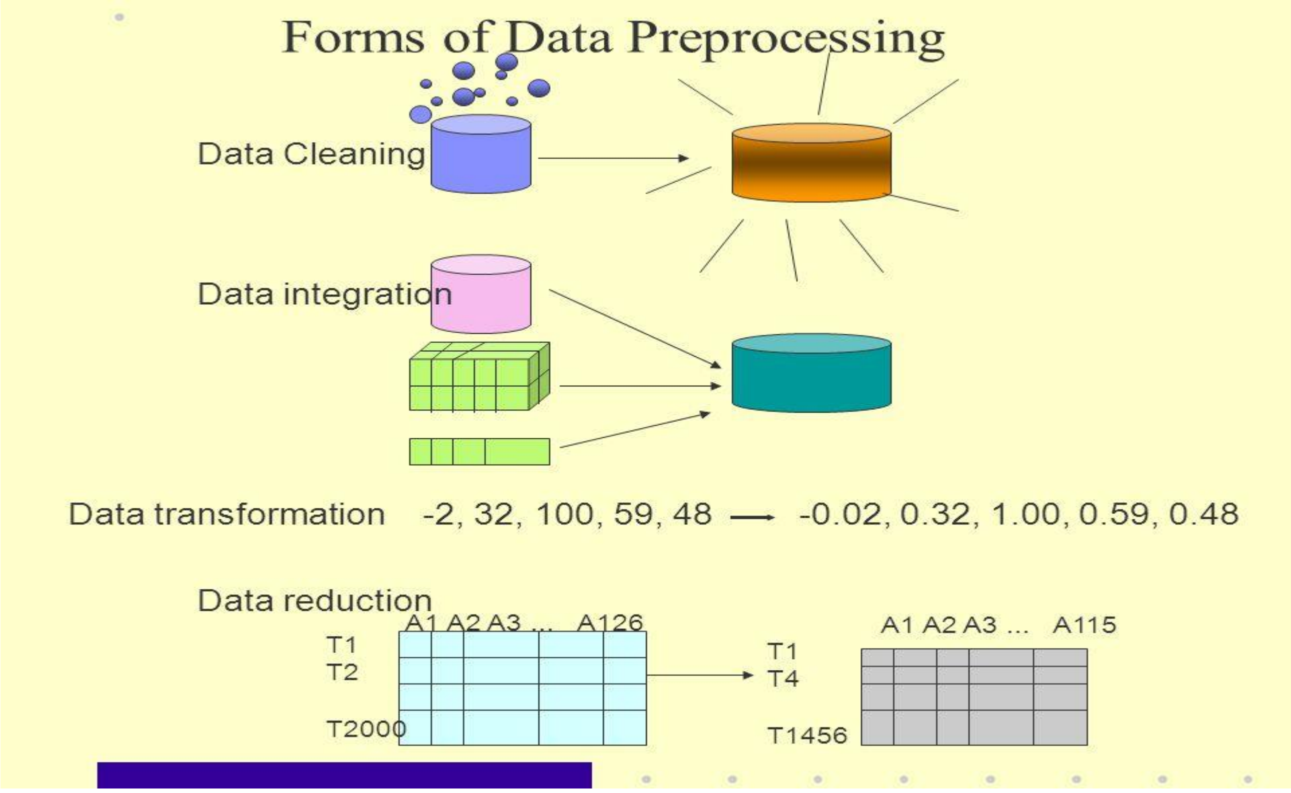
skewed distributions to resemble normality.

c. Feature Engineering:

- Construct technical indicators Create elements like ‘Open-Close’, ‘Low-High’, Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD).

d. Time Alignment:

- Align time series data to the same time intervals to create consistent time points for analysis.



Forms of Data Preprocessing

Data Cleaning

Data integration

Data transformation -2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

Data reduction

Figure 3.1 Data Preprocessing

Preprocessing data can be done using a variety of instruments and techniques, including:

- **Sampling:** Using this technique, a representative subset is chosen from a large population of data.
- **Transformation:** To create a consolidated input, raw data is modified.
- **Denoising:** To improve the quality of data, noise is removed.
- **Imputation:** The use of synthesized, statistically pertinent data to fill in missing values.
- **Normalization:** Information is organized to enable easier access.
- **Feature Extraction:** Relevant feature subsets are isolated based on how important they are in particular contexts.

Why is preprocessing of data necessary?

Data preprocessing is fundamental to many fields, including data science, data analysis, and the creation of artificial intelligence. Its importance lies in ensuring the accuracy, robustness, and dependability of results, which are essential for enterprise applications.

Real-world data is inevitably disorganized because it comes from numerous sources, procedures, and applications. Missing fields, incorrect manual entry, duplicate records, and inconsistent naming conventions can result from this. While operational data used by human users may address these problems, data used to train machine learning and AI models requires automated preprocessing.

When given data that highlights important aspects of problem-solving, efficient machine learning and deep learning algorithms flourish. Raw data is transformed into formats suitable for particular algorithms through feature engineering techniques like data wrangling, transformation, reduction, selection, and scaling. As a result, the time and computational requirements for training and inference are reduced.

To prevent reintroduction biases into the data, it is crucial to approach data preprocessing cautiously. For applications that affect individual decisions, like loan approvals, it is essential to identify and correct biases. Even if they are purposefully left out, factors like zip codes or educational backgrounds may be correlated with variables like gender, race, or religion, which could lead to biased results.

The following are the primary steps in data preprocessing:

Profiling of Data: Examining and analyzing data to compile statistics about its quality is known as data profiling. This entails determining pertinent attributes, formulating hypotheses regarding pertinent features, and picking the best preprocessing libraries in accordance with the issue at hand.

Data cleansing: This step aims to fix problems with the data's quality. It involves activities like purging inaccurate data, adding missing values, and making sure the raw data is appropriate for later feature engineering.

Data Reduction: Data redundancy and irrelevant information are common in data. Principal component analysis is one data reduction technique that simplifies data while preserving its essential characteristics.

Data Transformation: It involves organizing and structuring the data so that it is consistent with the goals of the analysis, AI, or ML task. This may entail combining variables, structuring unstructured data, or concentrating on particular ranges.

Data Enrichment: To achieve desired transformations, feature engineering libraries are applied to the data. As a result, a dataset that is well-suited for effective model training and computation is produced.

Data Validation: The dataset is divided into training and testing sets. The training set is used to train the model, while the testing set evaluates the model's accuracy and robustness.

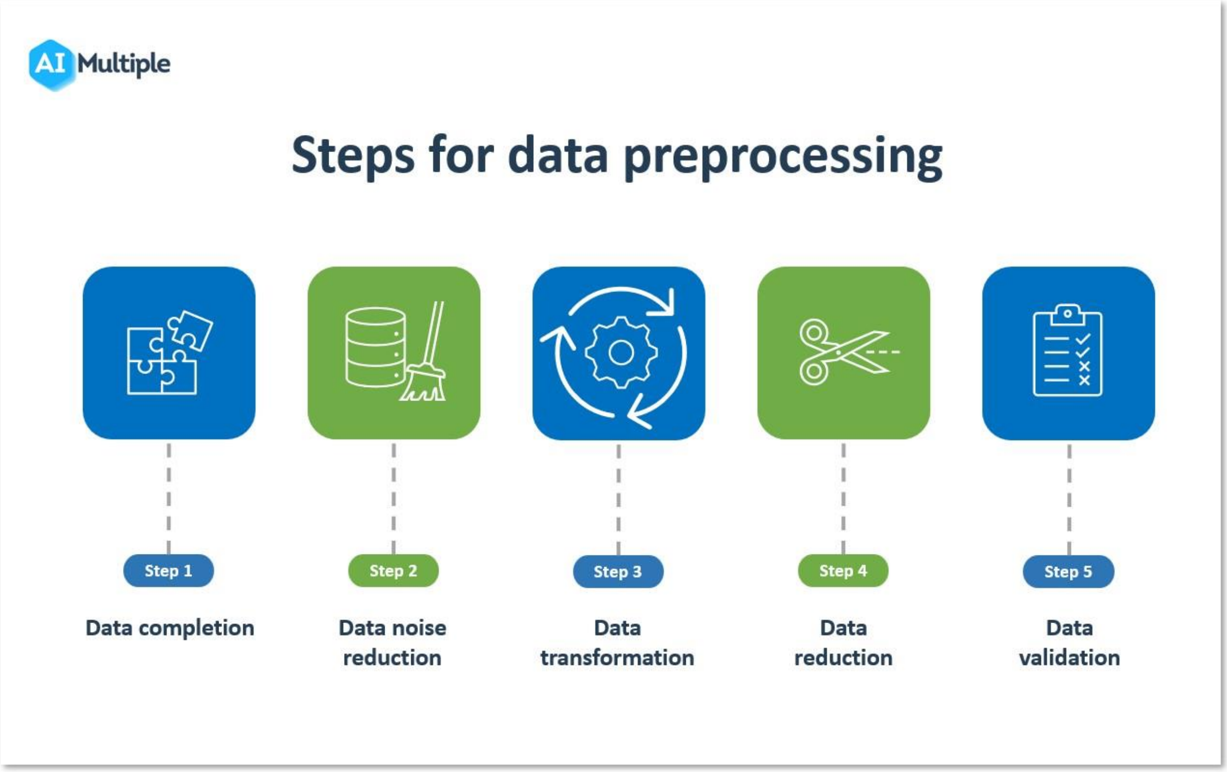


Figure 3.2 Steps for data preprocessing

Feature Engineering Techniques:

Feature Scaling or Normalization: Normalize data to ensure variables with varying scales are comparable. This is especially useful when some variables change linearly while others exhibit exponential changes.

Data Reduction: Combine variables or eliminate irrelevant ones to create a more efficient representation for AI or analytics models. Techniques like principal component analysis (PCA) help reduce dimensions in the training data.

Discretization: Group continuous numerical values into discrete intervals. For example, categorize income into representative ranges for specific loan applicant profiles.

Feature Encoding: Transform unstructured data (like text or images) into structured formats suitable for analysis. Techniques like Word2Vec translate text into numerical vectors, enabling algorithms to understand semantic relationships between words. Similarly, facial recognition algorithms convert raw pixel data into vectors representing facial features' spatial relationships.

By employing these data cleansing and feature engineering techniques, data scientists enhance data quality, optimize models' performance, and ensure that subsequent analysis or machine learning processes yield accurate and reliable results.

Step 3. Exploratory Data Analysis (EDA):

Exploratory Data Analysis (EDA) is a critical phase in an investment prediction project that involves visually and statistically analyzing the collected data. EDA helps uncover patterns, relationships, anomalies, and potential insights that inform subsequent steps in the project. Here's a comprehensive guide to conducting EDA for investment prediction:

- Use descriptive analysis to discover the distributions, relationships, and probable patterns of the data.
- Visualize data using plots and graphs to find connections between features and thyroid disease
- Exploratory data analysis (EDA), a method for data analysis, involves visually and statistically summarizing and analysing a dataset in order to obtain new insights while understanding its structure, patterns, and characteristics.
- EDA is often done at the start of a data analysis project to aid in hypothesis formation, trend identification, anomaly detection, and decision-making over how to move forward with more research

Step 4. Feature Selection:

- Identify relevant features through statistical tests, which can directly affect the result.
- Select a subset of features that contribute most to predicting stock value.
- Feature selection is a crucial step in data preprocessing and machine learning, involving the process of choosing a subset of the most relevant and informative features (variables) from the original dataset.
- The goal of feature selection is to improve model performance, reduce overfitting, enhance interpretability, and decrease computational complexity by working with a reduced set of features.

What is Feature Selection?

Feature selection is the key influence factor for building accurate machine learning models**.**
Other names of feature selection are variable selection or attribute selection.
"The method of reducing the number of input variables during the development of a predictive model."

OR

"Feature selection is a process of automatic selection of a subset of relevant features or variables from a set of all features, used in the process of model building."

Why is Feature Selection Important?

Irrelevant and misleading data features can **negatively impact** the performance of our machine learning model. That is why feature selection and data cleaning should be the first step of our model designing.

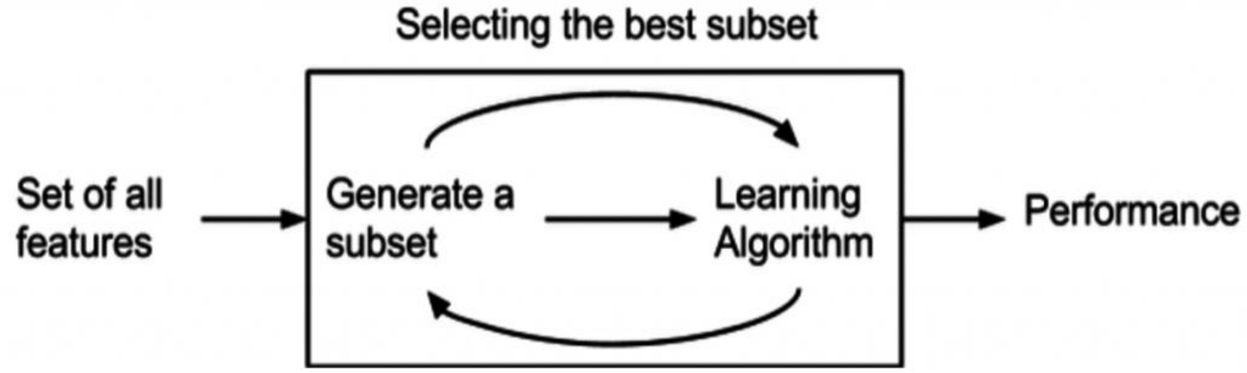


Figure 3.3 Feature Selection

Benefits of Feature Selection:

Having irrelevant features in your data can decrease the accuracy of many models, especially linear algorithms like linear and logistic regression.

The benefits of performing feature selection before modeling the model are as under:

- **Reduction in Model Overfitting:** Less redundant data implies less opportunity to make noise based decisions.
- **Improvement in Accuracy:** Less misleading and misguiding data implies improvement in modeling accuracy.
- **Reduction in Training Time:** Fewer data implies that algorithms train at a faster rate.

Feature Selection Models:

There are two types of models in Feature Selection:

Supervised Models: Supervised feature selection involves using the output label class to guide the feature selection process. These models utilize the target variables to identify which features contributeto enhancing the model's effectiveness. In the supervised method we can devide this in 3 more category:

- Filter Method
- Wrapper Method
- Embedded/Intrinsic Method

Filter Method:

This method, as the name implies, filters and selects only a subset of the relevant features. Following the selection of features, the model is constructed. The correlation matrix is used to filter the data, and Pearson correlation is the most commonly used.

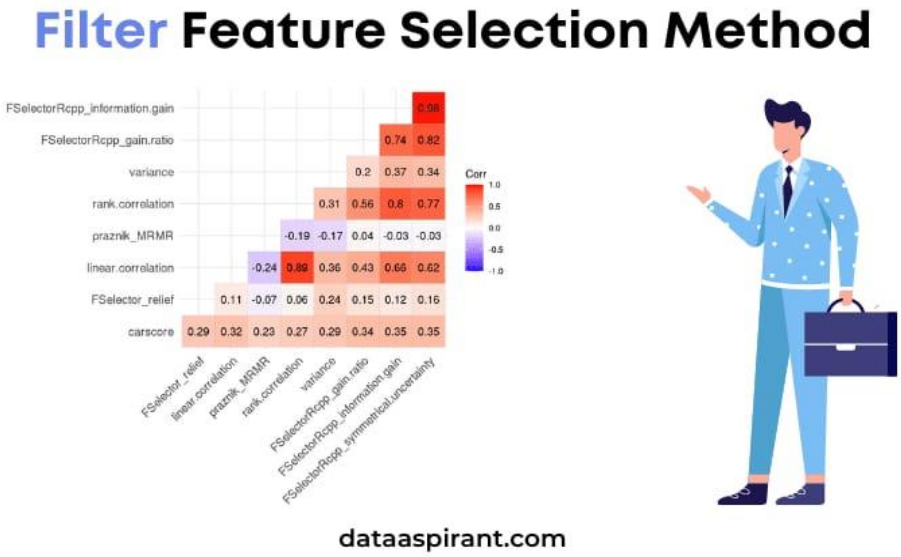


Figure 3.4 Filter Method

Wrapper Method:

Wrappering requires a single algorithm for machine Learning, and evaluation criteria shall be the performance of that algorithm. For example, you add or remove features according to model performance by feeding them into an algorithm of your choice for machine learning. This is an iterative, computationally expensive process but it's more precise than the filter method.

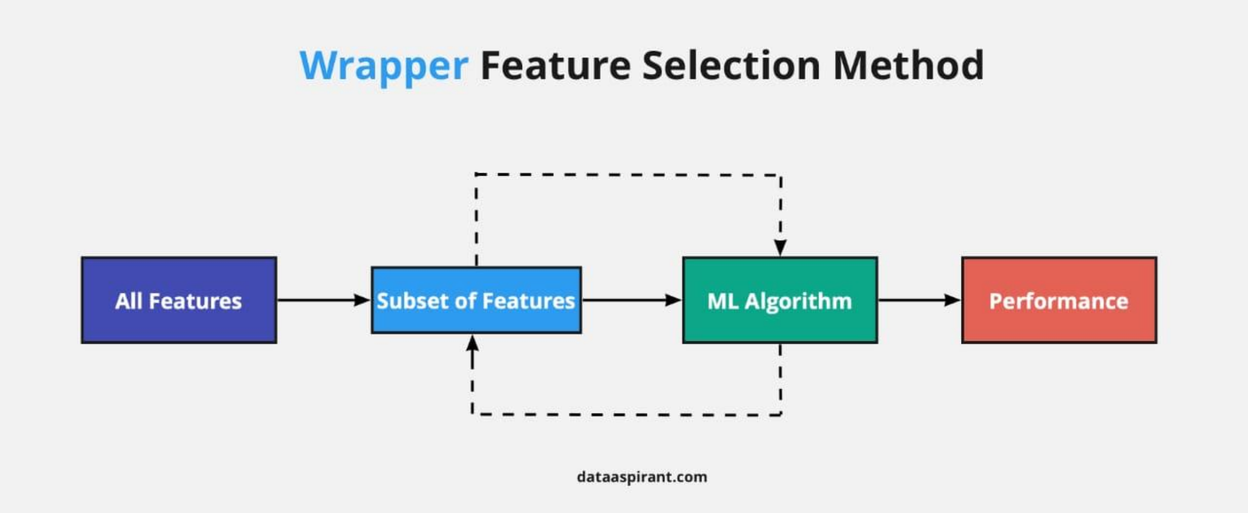


Figure 3.5 Wrapper Method

Intrinsic Method:

1. The intrinsic method integrates the characteristics of both the Filter and Wrapper methods to form an optimal subset of features.

By managing the iterative process of machine training while minimizing computational overhead, this method strikes a balance. Examples of techniques falling under this category include Lasso and Ridge Regression. Embedded methods are combination of filter and wrapper methods.

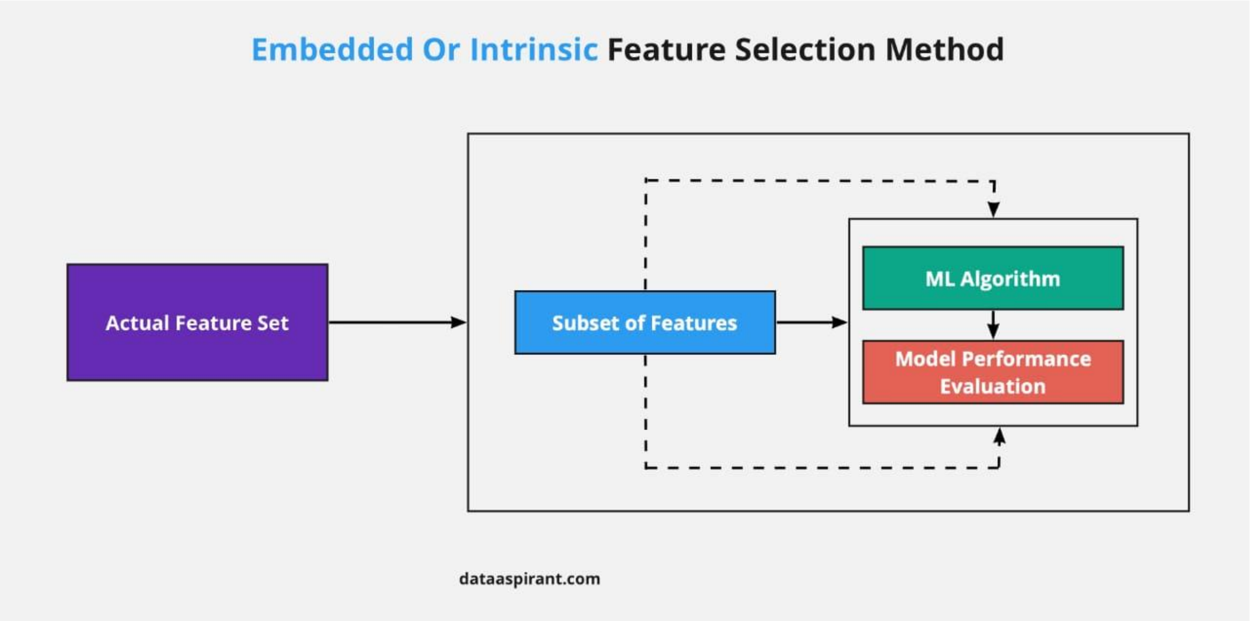

Embedded Or Intrinsic Feature Selection Method
Actual Feature Set
Subset of Features
ML Algorithm
Model Performance Evaluation
dataaspirant.com

Figure 3.6 Intrinsic Model

Unsupervised Models: Unsupervised feature selection operates without requiring the output label classfor the selection process. These models are employed when dealing with unlabelled data, where the focus is on identifying relevant features without relying on labeled information.

Key points about feature selection:

- **Relevance and Redundancy:** Feature selection aims to retain features that are directly related to the target variable, eliminating irrelevant features. It also seeks to remove redundant features that convey similar information, which can lead to multicollinearity issues.
- **Simplification and Interpretability:** By working with a smaller set of features, the model becomes simpler and more interpretable. This is particularly valuable when explaining the model's results to stakeholders.
- **Improved Efficiency:** Fewer features mean reduced computational demands during model training and inference, leading to faster processing times.
- **Overfitting Prevention:** Including too many features in a model can lead to overfitting, where the model fits the noise in the data rather than the underlying patterns. Feature selection helps mitigate this issue
- **Domain Knowledge and Exploration:** While some feature selection methods are automated, domain knowledge can guide the selection process, ensuring that relevant features are retained

based on their significance in the problem context.

Step 5: Machine Learning Model Selection and Training

What is Machine Learning?

Machine Learning is the science to make computers learn from data without explicitly programming them and improving their learning over time in an autonomous fashion.
This learning comes by feeding them data in the form of observations and real-world interactions."
Machine Learning can also be defined as a tool to predict future events or values using past data.

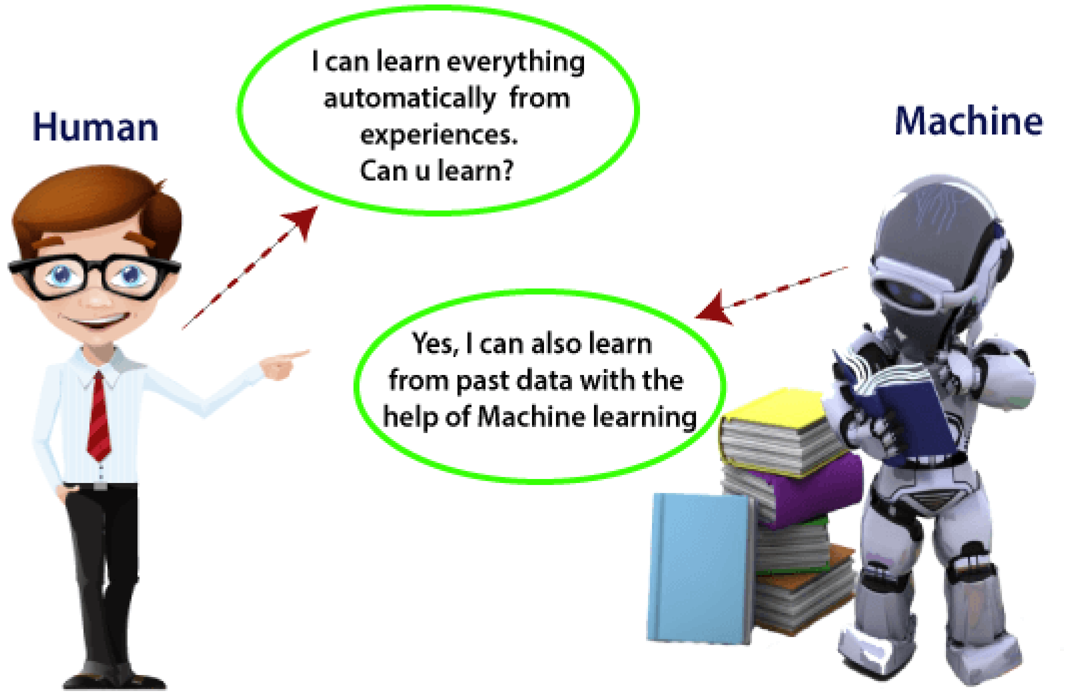

Human
I can learn everything automatically from experiences. Can u learn?
Machine
Yes, I can also learn from past data with the help of Machine learning

Figure 3.7 Machine Learning

Working of machine learning models:

We use past data to train the model, and evaluate the model on fresh data and predict outcomes. We can assess the efficiency of the trained ML model using a subset of the available historical data (which is not present during training). This is commonly known as the validation procedure. Accuracy measures the ML model's performance over unknown data by dividing the number of properly predicted features by the total number of available features to be predicted.

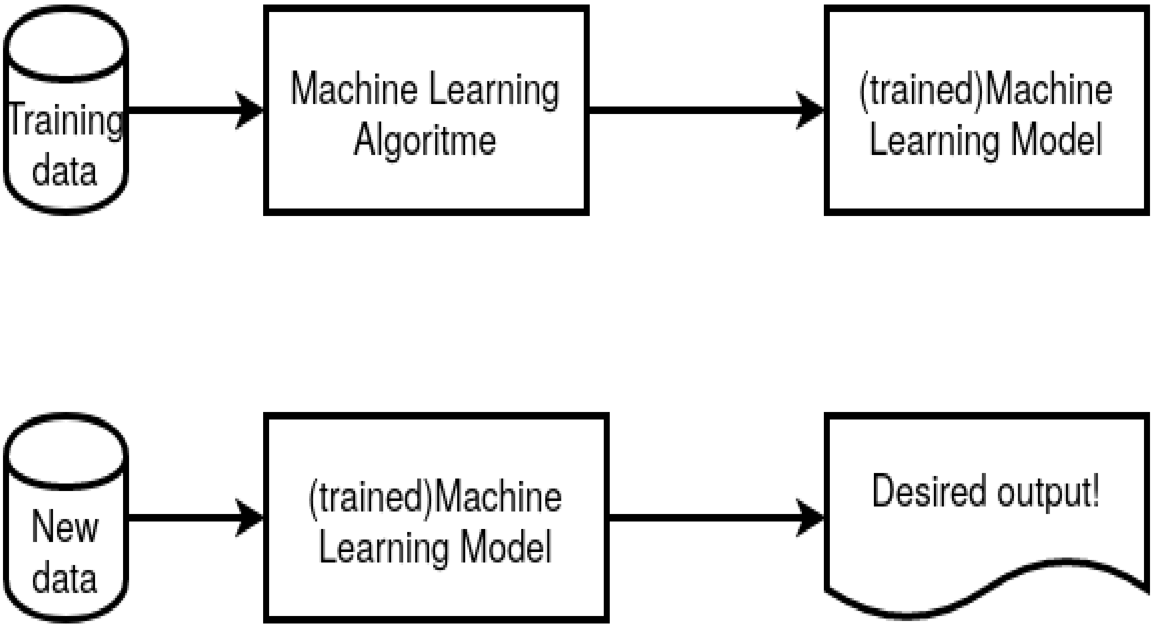

Training data
Machine Learning Algoritme
(trained)Machine Learning Model
New data
(trained)Machine Learning Model
Desired output!

Figure 3.8 Working of Machine Learning

Logistic Regression:

Logistic regression is a type of machine learning algorithm that is used for classification tasks and models the probability that a sample belongs to a certain class using a logistic function.

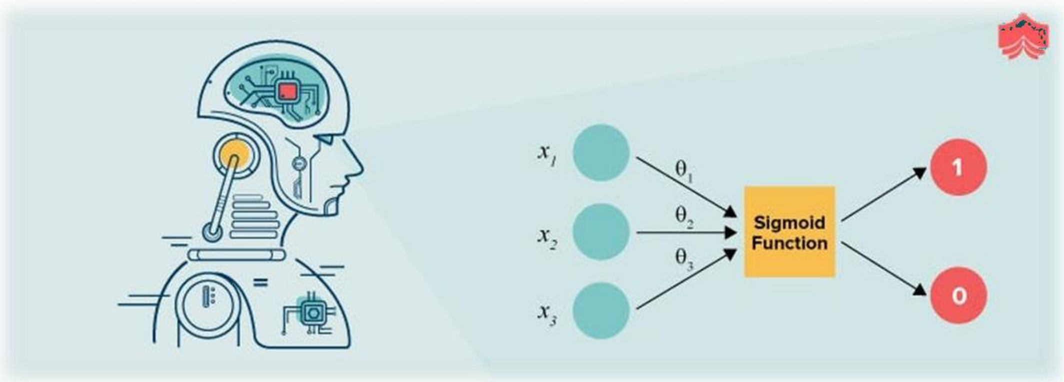


Figure 3.9 Logistic Regression

Logistic regression is a statistical model used to analyze the relationship between a dependent variable (usually binary, meaning it can take one of two values) and one or more independent variables. It is commonly used for classification problems, such as predicting the probability of an event occurring or not occurring based on a set of independent variables

$$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}} = \frac{1}{e^{-\log(odds)} + 1} = \frac{1}{1 + e^{-z}} = Sigmoid\ Function$$

Figure 3.10 Logistic Sigmoid Function

Usage of Logistics Regression:

Fraud Detection: Logistic regression models can help teams identify data anomalies, which are predictive of fraud.

Investment Prediction: In market investment, this analytics approach can be used to predict the price of any stock.

Churn prediction: Specific behaviors may be indicative of churn in different functions of an organization. Logistics regression can be applied to predict churn in areas like sales, HR, etc.

Binary Classification: Logistic Regression is mostly used for binary classification problems in which the aim is to forecast the likelihood of an instance falling into one of two groups.

As an example:

- Predicting whether or not an email is spam.
- Whether a consumer will leave or stay with a service.
- Determining whether a transaction is legitimate or fraudulent

Linear vs Logistic Regression:

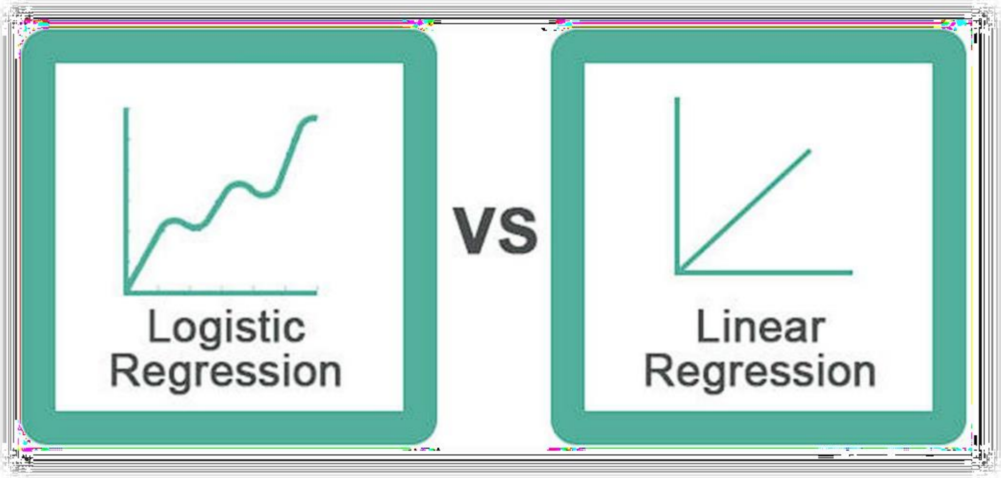


Figure 3.11 Linear Regression Vs Logistics Regression

The main difference between linear and logistic regression is the type of response variable.
A linear regression model is used when the response variable takes on a continuous value such as price, height, age, and distance. Conversely, a logistic regression model is used when the response variable takes on a categorical value such as yes or no, win or not win.

How Logistic Regression Works?

The logistic regression model uses a sigmoid or logistic function to map the input variables to a probability value between 0 and 1. The sigmoid function is given by:
p = 1 / (1 + e^(-z))
where p is the predicted probability of the event occurring, e is the mathematical constant, and z is a linear combination of the input variables and their associated coefficients.

How to Build a Logistics Regression Model:

1. Collect and preprocess the data: Collect a dataset that includes both the predictor variables (also known as features) and the dependent variable. Preprocess the data to remove missing values and outliers, normalize the data, and perform other data cleaning steps as necessary.
2. Choose the features: Select the features that are most relevant to the problem being addressed. This can involve using domain knowledge, statistical tests, or feature selection techniques.
3. Fit the model: Estimate the coefficients of the logistic regression model using maximum likelihood estimation. This involves finding the set of coefficients that maximizes the likelihood of the observed data.
4. Evaluate the model: Evaluate the performance of the model using appropriate metrics such as accuracy, precision, recall, and F1 score. Use techniques such as cross-validation to ensure that the model is not overfitting the data.
5. Use the model to make predictions: Once the model has been trained and evaluated, it can be used to make predictions on new data. To make a prediction, the values of the predictor variables are input

into the model, and the sigmoid function is used to estimate the probability of the event occurring.

Drawbacks of Logistics Regression:

a) **Linear decision boundaries:** Logistic regression models can only represent linear decision boundaries between classes. If the classes are not linearly separable, then the model may not perform well.

b) **Sensitive to outliers:** Logistic regression models are sensitive to outliers, which can have a large influence on the estimated coefficients and predictions.

c) **Assumes independence of predictor variables:** Logistic regression assumes that the predictor variables are independent of each other. If there are correlations among the predictor variables, then the coefficients may be unstable and the model may not perform well.

d) **May not work well with small sample sizes:** Logistic regression requires a relatively large sample size to estimate the coefficients accurately. If the sample size is small, then the model may not perform well.

e) **Cannot handle non-linear relationships:** Logistic regression assumes a linear relationship between the predictor variables and the logit of the dependent variable. If there are non-linear relationships, then the model may not capture them.

f) **May overfit the data:** Logistic regression models can overfit the data if too many predictor variables are included or if the model is too complex. This can result in poor performance on new data.

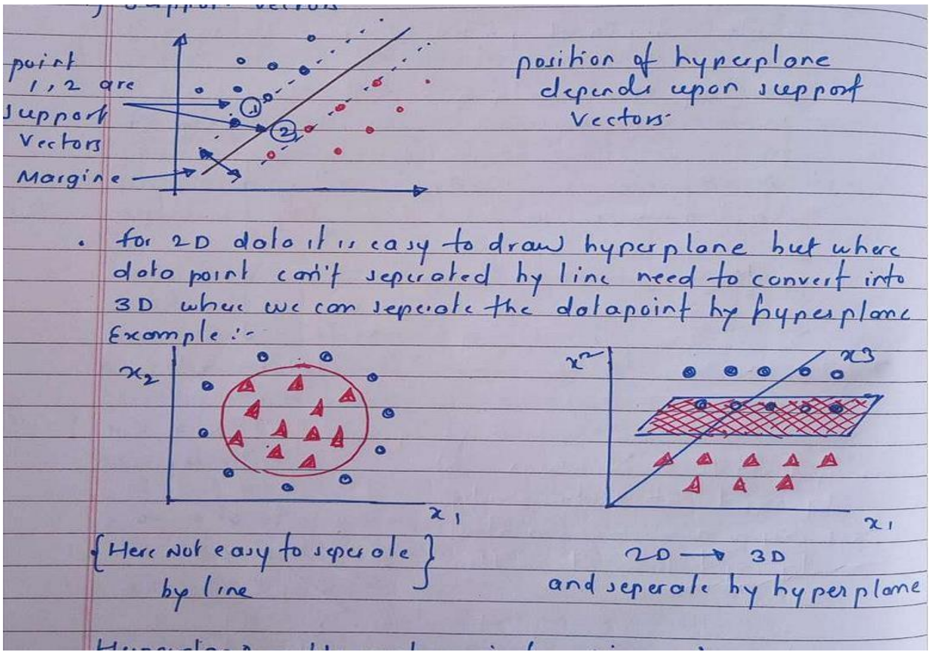
Advantages:

1) Easy to implement and interpret yet efficient in training.
2) The predicted parameters give inference about the importance of each feature.
3) Performs well on low-dimensional data.
4) Very efficient when the dataset has features that are linearly separable.

Disadvantages:

1) Overfits on high dimensional data.
2) Non-linear problems can't be solved with logistic regression since it has a linear decision surface.
3) Assumes linearity between dependent and independent variables.
4) Fails to capture the complex relationship

Support Vector Machine SVM:
Basic about SVM. It is a supervised machine learning model.
It can be used for both classification as well as regression but it predominantly use for binary Classification
Hyperplane / Support Vector


point 1, 2 are Support Vectors
Margine
1
2
position of hyperplane depends upon support Vectors
x2
x1
x2
x3
x1
{Here Not easy to seperate by line}
2D → 3D
and seperate by hyperplane

- for 2D data it is easy to draw hyperplane but where data point can't seperated by line need to convert into 3D where we can seperate the datapoint by hyperplane
Example :-

Figure 3.12 SVM

Hyperplan: A hyperplane is line (in 2D) or plane that separates the data into two classes.

Support Vectors: These are the data points that are nearest to the hyperplane if these data points change position to the hyperplane changes

A well-known machine learning technique that rose to prominence in the late 1990s is the Support Vector Machine (SVM). This method runs in the supervised learning space and does two tasks simultaneously: regression and classification.
The Support Vector Machine (SVM) is a discriminative classifier that divides data into multiple classes. Its goal is to identify the ideal hyperplane. This ideal hyperplane can be seen as a line that separates the space into two segments, one containing data points belonging to one class and the other containing data points from a different class, in a two-dimensional setting.

It's important to remember that this linear separation only applies when the data points can be separated linearly. SVMs, however, provide a more flexible option because they may be used to locate the best curve for classifying data points when linear separation is not an option.

In the context of regression, SVM can be utilized to fit a curve or a line to a given set of data points.

Let us understand this with respect to two-dimensional space and lines which can be extended to greater than two-dimensional space and hyper-planes

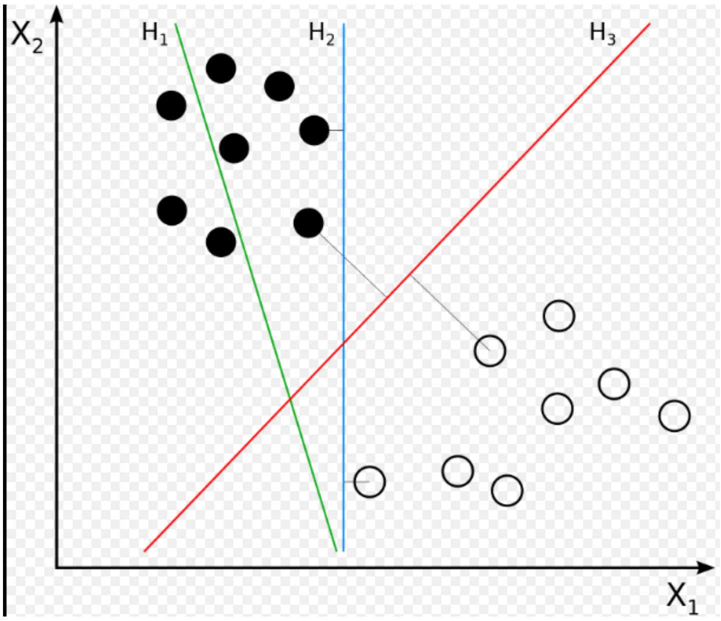


Figure 1.13 SVM Margin

Let us look at the image above, which shows two different types of data points. It is significant that the black-filled data points represent the positive class, whilst the black-bordered data points represent the negative class. In this case, we can see three different straight lines, each of which has the ability to divide the data points into different classes. Which of these straight lines is best for this categorization task, though, is the crucial question at hand.

Intuition About SVMs

Earlier we employed the term “optimal hyperplane.” Now we will delve into the precise meaning of this concept.

The term "optimal hyperplane" refers to a separating hyperplane that keeps the greatest feasible separation between data points of different classes on either side.

The hyperplane that maximizes the perpendicular separation between the hyperplane and the nearest samples is known as the ideal hyperplane. Support vectors can be used to bridge this gap effectively.

Moving forward, we will introduce a crucial concept known as the “margin.”

Let us see the illustration in the figure below

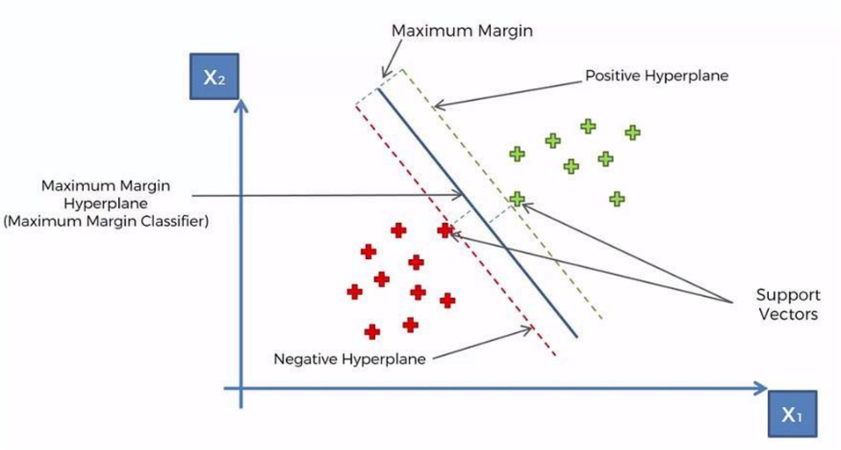


Figure 3.14 Showing positive, negative, and separating hyper-planes, support vectors, and margin

We can observe three distinct hyperplanes in the figure:

1) The hyperplane which touches the positive class points is referred to as the positive hyperplane.
2) The hyperplane which touches the negative class points is termed the negative hyperplane.
3) The hyperplane positioned between the positive and negative classes is known as the separating hyperplane.

Crucially, all three of these hyperplanes are parallel to each other.

The "margin" is the separation between the positive and negative hyperplanes. The positive and negative points must be far apart from one another and from the separating hyperplane in order to maximize this margin. As a result, the categorization task's accuracy increases. A more accurate classification is correlated with a larger margin. An improvement in "generalization accuracy"—a measure of the model's performance on hypothetical, upcoming data points—is closely correlated with an increase in margin.

A hyperplane that maximizes the margin is what Support Vector Machines (SVMs) are designed to find. As a result, the best or separating hyperplane is frequently referred to as the "margin-maximizing hyperplane."

In Figure 11, it is evident that the H3 hyperplane represents the most suitable choice in this context.

Support Vectors

Data points that touch both the positive and negative hyperplanes make up support vectors. some data points that are clearly situated near the point where the positive and negative hyperplanes meet. These particular data points are known as "Support Vectors."

Please note that we will not take into account the positive and negative hyperplanes when classifying future, unseen data points; rather, we will only evaluate the position, direction, and side of the data point in reference to the separating hyperplane.

Advantages:

1) Support Vector Machine works relatively well when there is a clear margin of separation between classes.

2) Support Vector Machine is more effective in high-dimensional spaces.
3) Support Vector Machine is effective in cases where the number of dimensions is greater than the number of samples.
4) Support Vector Machine is relatively memory efficient

Disadvantages: -

1) Support Vector Machine algorithm is not suitable for large data sets.
2) Support Vector Machine does not perform very well when the data set has more noise i.e., target classes are overlapping. 3. In cases where the number of features for each data point exceeds the number of training data samples, the Support Vector Machine will underperform.
3) As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification.

XGBoost

XGBoost (eXtreme Gradient Boosting) is a popular machine learning algorithm that belongs to the family of gradient boosting methods. It is designed to improve the performance of models for both regression classification and tasks. XGBoost is known for its efficiency, speed, and predictive power, and it has been widely used and successful in various machine-learning real-world applications.

Here are some key features and concepts associated with XGBoost:

Gradient Boosting: XGBoost is based on the gradient boosting framework, which is an ensemble learning technique. It involves training a sequence of weak learners (typically decision trees) in such a way that each new learner corrects the errors made by the previous ones.

Regularization: XGBoost employs a technique called regularization to prevent overfitting. It includesparameters that control the complexity of the individual trees and the overall model, thus helping to improve generalization to unseen data.

Gradient Descent Optimization: XGBoost optimizes its model by minimizing a loss function using gradient descent techniques. It computes the gradients of the loss function with respect to the model's predictions and updates the model's parameters accordingly.

Boosting Rounds: During training, XGBoost builds an ensemble of weak learners sequentially, with each new learner focusing on the errors made by the previous ones. The number of boosting rounds is ahyperparameter that determines how many weak learners are trained.

Feature Importance: XGBoost provides a way to compute feature importance scores, which indicate the contribution of each feature in making predictions. This can be helpful for understanding the impactof different features on the model's decisions.

Handling Missing Values: XGBoost can handle missing values within the dataset. It learns how to make decisions even when some data points have missing values for certain features.

Parallel and Distributed Computing: XGBoost is designed for efficiency and can utilize parallel and distributed computing resources to speed up training and prediction, making it suitable for large datasets.**Flexibility:** XGBoost supports both classification and regression tasks, and it can handle various typesof input features, including numerical and categorical features.

Cross-Validation Support: XGBoost provides tools for performing cross-validation to assess the model's performance and tune hyperparameters effectively.

Overall, XGBoost's combination of boosting, regularization, and optimization techniques makes it a powerful algorithm for a wide range of machine-learning problems. It's worth noting that since my knowledge is based on information available until September 2021, there might have been developmentsor variations of XGBoost since that time.

Applications:

Classification: XGBoost is commonly used for binary and multiclass classification jobs, such as:

- Fraud detection is the process of detecting fraudulent transactions based on transaction characteristics.
- Disease diagnosis is the prediction of the presence of a disease based on the findings of medical tests.
- Predicting customer churn: determining if a consumer is likely to depart a service.

Time Series Forecasting: Because XGBoost can capture temporal dependencies, it is beneficial for time series forecasting:
Stock price forecasting: The prediction of future stock prices based on previous data.
Predicting future energy demand for resource allocation is known as energy consumption prediction.

Advantages:

1) Regularization: XGBoost has in-built L1 (Lasso Regression) and L2 (Ridge Regression) regularization which prevents the model from overfitting. That is why, XGBoost is also called aregularized form of GBM (Gradient Boosting Machine).
2) Handling Missing Values: XGBoost has an in-built capability to handle missing values. When XGBoost encounters a missing value at a node, it tries both the left and right-hand split and learns the way leading to a higher loss for each node. It then does the same when working on thetesting data.
3) Parallel Processing: XGBoost utilizes the power of parallel processing and that is why it is muchfaster than GBM. It uses multiple CPU cores to execute the model.

Disadvantages:

1. Overfitting: Overfitting is likely to occur in xgboost if xgboost parameters are not tuned properly.
2. Training time: Training time is pretty high for the larger dataset, if you compare it against catboost/lightgbm.

Step 6. Model Evaluation and Interpretation:

It is useful to evaluate the model's accuracy and determine how well the trained modelperformed. By comparing a machine learning model's adaptiveness to that of non-adaptive models, we may determine how effectively it generalizes to new data. Any data science projectthat aims to establish the generalization accuracy of a model on future data must include analyzing the performance of a machine learning model.

Assess the selected model's performance using various metrics like accuracy, precision, recall,Select the performing model and the Model Selection phase is complete.

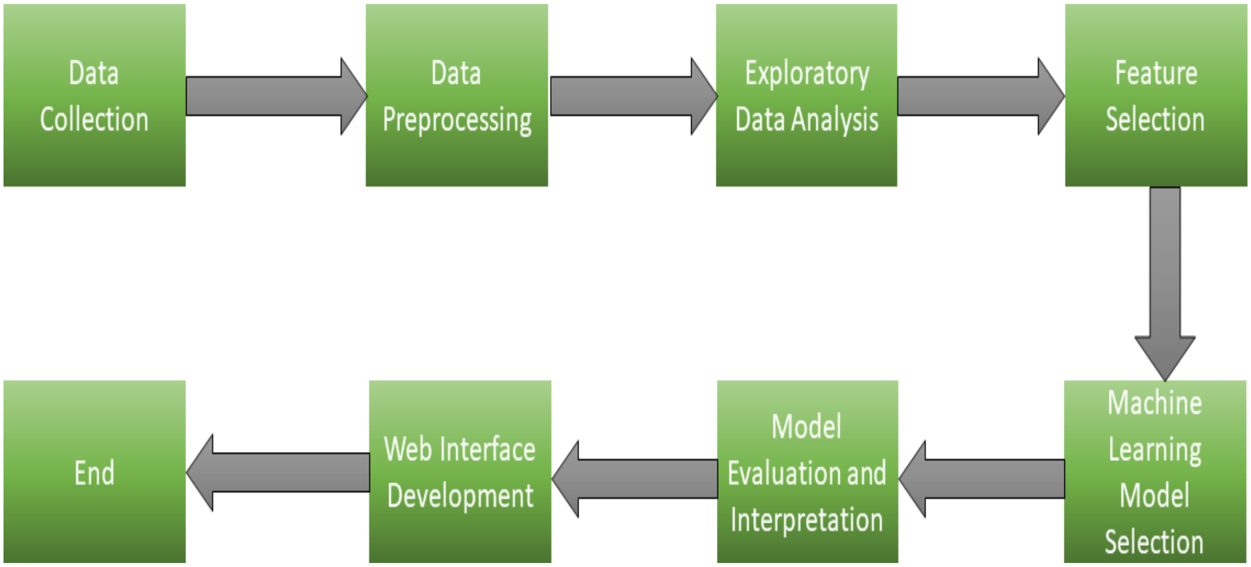

Data Collection
Data Preprocessing
Exploratory Data Analysis
Feature Selection
Machine Learning Model Selection
Model Evaluation and Interpretation
Web Interface Development
End

Figure 3.15 Circuit Layout

CHAPTER 4 IMPLEMENTATION

4.1 Modules

Introduction

This implementation report presents a step-by-step analysis of stock price data for Tata Motors (TATAMOTORS.NS) using various data visualization techniques and machine learning models. The analysis aims to predict whether the stock price will increase or decrease in the future.

Libraries and Data

The following libraries are imported:

- numpy and pandas for data manipulation
- matplotlib.pyplot and seaborn for data visualization
- yfinance for fetching stock data
- Standard.Scaler from sklearn.preprocessing for feature scaling
- Logistic Regression, SVC, and XGB Classifier from sklearn for building machine learning models
- Various functions from sklearn.metrics for evaluation purposes
- The data is fetched using the yfinance library, containing Tata Motors stock price data from January 1, 1998, to January 1, 2023.

Data Visualization

Closing Price Time Series Plot: A line plot of the closing price of Tata Motors stock over time is created.

Distribution Plots:

Distribution plots (histograms) for the features 'Open', 'High', 'Low', 'Close', and 'Volume' are generated.

Box Plots:

Box plots for the same features are created to visualize the spread and distribution of data.

Price Aggregations by Year:

Average open, high, low, and close prices are calculated and visualized in separate bar plots for each year.

Correlation Heatmap:

A heatmap depicting the correlation matrix of the features is displayed to understand the relationships between variables.

Feature Engineering

Time-Related Features:

'Day', 'Month', and 'Year' columns are extracted from the 'Date' column. An 'is_quarter_end' column is created to identify quarter-end months.

Derived Features:
Two derived features are created: 'open-close' (difference between opening and closing prices) and 'low-high' (difference between low and high prices). A 'target' column is also added to indicate whether the next day's closing price will increase (1) or not (0).

Model Building and Evaluation
Data Preprocessing:
The features 'open-close', 'low-high', and 'is_quarter_end' are selected. The target variable is 'target'. The features are scaled using StandardScaler.

Train-Test Split:
The data is split into training and validation sets using a 90-10 split ratio.

Model Selection and Training:
Three models are chosen: Logistic Regression, Support Vector Classifier (SVC) with a polynomial kernel, and XGBoost Classifier. Each model is trained on the training data.

Model Evaluation:
For each model, the following evaluations are performed on the validation set:

- ROC-AUC score
- F1-Score
- Confusion Matrix

Model Comparison:
F1-Scores and accuracy scores of the three models are compared using bar plots.

Time-Series Analysis
Baseline Accuracy:
The baseline accuracy is calculated as the mean of the target variable.

Time-Step Analysis:
For different time steps (1, 10, and 30 days), the following steps are performed:

- Calculate shifted labels for the target variable
- Evaluate accuracy and F1-Score for each model
- Compare the results with the baseline accuracy

Results Display:
The accuracy and F1-Score for each model and time step are displayed in a structured tabular format.

Visualization:
Bar plots are created to compare accuracy and F1-Score across different time steps for each model, with a baseline accuracy reference line.

4.2 Prototype

The Flask web framework was used to install an XGBoost machine learning model in this prototype report. Based on the features provided, the model is trained to forecast the movement of stock prices. The report summarizes the deployment process, including model saving, web application setup, testing, and recommended improvements.

The purpose of this deployment is to construct a web application that predicts stock price movement using the trained XGBoost model. This program offers an API endpoint for making predictions based on input characteristics.

The Flask application creates an API endpoint that offers stock price forecasts depending on input characteristics. Endpoint testing with sample input yielded accurate prediction results.

User Interface:

We leverage the Flask framework's features to create a user-friendly web interface.
What is the definition of Flask?

Flask is a Python package that simplifies the process of developing web applications.
Its fundamental functionality is intended to be easily extendable, resulting in a microframework that purposefully excludes components such as an Object Relational Manager (ORM) and other complicated capabilities.

Its power resides in its simplicity. Flask does not overburden itself with capabilities such as URL routing and a sophisticated template engine. Instead, it stays focused on being a WSGI web application framework.

Now, let us delve into the concept of a Web Framework:
A Web Framework serves as the grouping of libraries and modules that empowers web application developers to craft their applications without the need to worry about complex detailsat a lower level, such as protocols and thread management.

In the realm of Python, the distinguished template engine known as Jinja2 plays a pivotal role. This particular web template system merges templates with specific data, completing the dynamic rendering of web pages that adapt to the context at hand.

CHAPTER 5 RESULTS AND ANALYSIS

This project aims to predict the price of stocks.

Problem Statement:

The area Stock market is something which is with lots of ups and downs. It can change in no time. Therefore we can use machine learning technique to identify market changes earlier than possible with traditional investment models.

5.1 Results

- Accuracy Score for the Algorithms which has used in project.

Time-Stamp	Logistic Regression	XG BOOST	Support Vector Classifier	Baseline Accuracy
For 1 Day	0.4928	0.4944	0.4928	0.5183
For 10 Day	0.4832	**0.5263**	0.4864	0.5183
For 30 Day	0.4896	0.4816	0.4896	0.5183

- ROC-AUC Score for the Algorithms which has used in project.

	Logistic Regression	*XG BOOST*	*Support Vector Classifier*
ROC-AUC Score	0.4878	**0.5370**	0.4614

5.2 Analysis

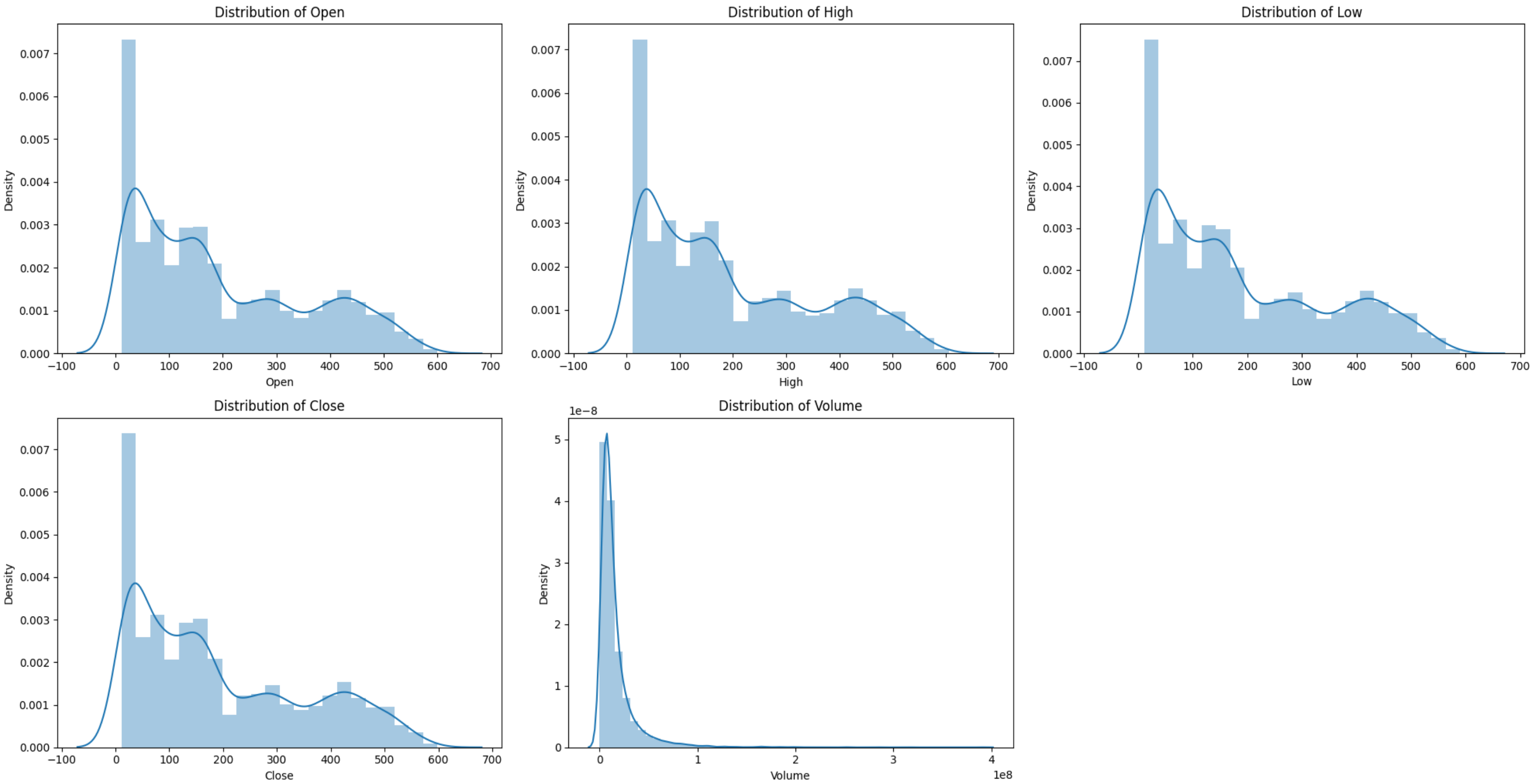


Figure 5.1 Plotting the Distribution for Diff Features

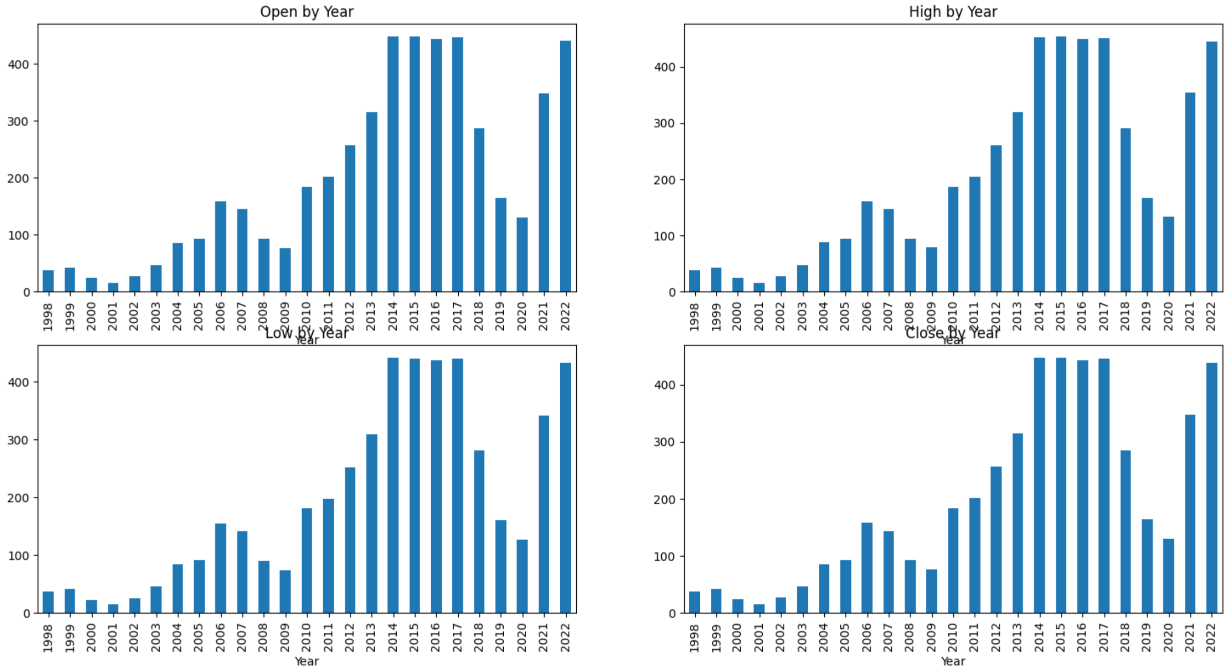


Figure 5.2 Bar Graph for Close Price Over the years

CHAPTER 6 CONCLUSION AND FUTURE SCOPE

6.1 Conclusions:

Using a combination of data visualization, feature engineering, and machine learning modeling, the offered code analyzes historical stock data for Tata Motors.
By above analysis we can say that XG Boost is very good for prediction over a long time with accuracy_score 0.5263.

6.2 Future Improvements:

- Prediction accuracy may be improved by including other financial information..
- Investigating deep learning methods such as convolutional neural networks (CNNs), recurrent neural networks (RNNs) or long short-term memory networks (LSTMs) maybe able to identify more intricate patterns in the data.

REFERENCES

Reference / Hand Books

[1] Name 1, "Stock Price Prediction Using Machine Learning" Yixin Guo,
Södertörn University | School of Social ScienceMaster
Dissertation 30hp
Economics Spring 2022

[2] Name 2, "Stock Price Prediction Using Machine Learning" :Maithili Patel,
Birla Vishvakarma Mahavidyalaya, Vallabh Vidyanagar – 388120 GUJARAT,
INDIA

[3] Name 3, "EQUITY PRICE PREDICTION" Ayush Nandwana and Pradyuman Mishra,
DEPARTMENT OF INFORMATION TECHNOLOGY FACULTY OF ENGINEERING
AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603203

Web:
Feature Selection, www.dataaspirant.com
Feature Engineering, www.Analyticsvidya.ai
Model Selection, www.quantinsti.com;
www.section.io,
www.towardsdatascience.com