:

TIME SERIES – STOCK PRICE PREDICTION USING BOOSTING ALGORITHM ML MODEL

A PROJECT REPORT

Submitted by

JAYAKUMAR S	**211420104109**
KRISHNAKUMAR M	**211420104139**
KUMARESWAAR M	**211420104142**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

:

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“Time Series – Stock Price Prediction using Boosting algorithm ML Model”** is the bonafide work of **“JAYAKUMAR S[211420104109], KRISHNAKUMAR M[211420104139], KUMARESWAAR M[211420104142]”** who carried out the project work under my supervision**.**

SIGNATURE	**SIGNATURE**
DR L. JABASHEELA M.E.,Ph.D., **HEAD OF THE DEPARTMENT**	**Dr. MOHANA PRAKASH TA,M.Tech,Ph.D.,** **SUPERVISOR** **ASSOCIATE PROFESSOR**
Department of Computer Science and Engineering, Panimalar Engineering College, Chennai - 123	Department of Computer Science and Engineering, Panimalar Engineering College, Chennai - 123

Certified that the above candidate(s) was examined in the End Semester Project

Viva-Voce Examination held on ....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

:

DECLARATION BY THE STUDENT

We JAYAKUMAR S [211420104109], KRISHNAKUMAR M[211420104139], KUMARESWAAR M [211420104142] hereby declare that this project report titled **"Time Series – Stock Price Prediction using Boosting algorithm ML Model",** under the guidance of **Mr. MOHANA PRAKASH M. Tech ,Ph.D.,**is the original work done by us and we havenot plagiarized or submitted to any other degree in any university by us.

JAYAKUMAR M [211420104109]
KRISHNAKUMAR M [211420104139]
KUMARESWAAR M [211420104142]

:

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI M.A. Ph.D**. for his fervent encouragement. His inspirationalsupport proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We wish to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHIKUMAR M.E. Ph.D. an Dr. SARANYASREE SAKTHIKUMAR B.E. M.B.A. Ph.D.** for graciously affording us the essential resources and facilities for undertaking this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI M.E. Ph.D.** whose facilitation proved pivotal in the successful completion of this project.

We express our heartfelt thanks to **Dr. L. JABASHEELA M.E. Ph.D.** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of the project.

We would like to thank the **Project Coordinator Dr.G.SENTHIL KUMAR,M.C.A.,M.Phil.,M.E.,M.B.A.,Ph.D** and **Project Guide Dr.T.A.MOHANA PRAKASH,M.Tech,Ph.D.,** all the facultymembers of the Department of CSE for their unwavering support for the successful completion of the project.

JAYAKUMAR M [211420104109]
KRISHNAKUMAR M [211420104139]
KUMARESWAAR M [211420104142]

:

Pantech e Learning
DIGITAL LEARNING SIMPLIFIED

CIN:U80902TN2021PTC141464

www.pantechelearning.com

20/03/2024

COMPLETION CERTIFICATE

This is to acknowledge that students **KRISHNAKUMAR M, JAYAKUMAR S, KUMARESWAAR M** from **“PANIMALAR ENGINEERING COLLEGE”** has completed **Project** on the **Title** of **“TIME SERIES - STOCK PRICE PREDICTION USING BOOSTING ALGORITHM ML MODEL (AMAZON STOCK PRICE)”** at our concern from **DEC2023** to **MARCH2024**

ForPantechelearning.,

PANTECH E LEARNING CHENNAI-17

AuthorizedSignatory

PantecheLearningPvtLtd.,
IIFloor, KottaSrinivasiahCharitiesBuilding,
ThanjavurStreet,NearDuraisamySubway,T.Nagar,Chennai–600017Phone:
91 44 42606470 | hr@pantechmail.com

:

ABSTRACT

This study delves into the future landscape of wealth forecasting in stock markets, emphasizing the pivotal role of advanced Machine Learning (ML) techniques for precise AI-driven stock predictions. Leveraging sophisticated ML methodologies, this approach aims to bolster predictive models without explicitly mentioning specific algorithms. Advancements in ML models, real-time data processing, alternative data sources, and robust risk management strategies are pivotal factors driving improved AI-based stock predictions. However, it remains essential to complement AI-generated insights with comprehensive financial analysis and ensure ethical compliance in AI-driven wealth forecasting, given the inherent unpredictability of financial markets.

The future of stock market wealth forecasting relies on advanced Machine Learning (ML) models, including deep learning and sentiment analysis. ML-driven improvements, real-time data processing, and robust risk management will enhance AI-based stock predictions. Yet, balancing AI insights with comprehensive financial analysis and ensuring ethical compliance remain crucial due to market unpredictability.

:

TABLE OF CONTENTS

:

:

LIST OF TABLES

-

:

LIST OF FIGURES

:

LIST OF ABBREVIATIONS

AI	**Artificial Intelligence**
ML	**Machine Learning**
CNN	**Convolutional Neural Network**
LLM	**Large Language Model**

:

CHAPTER 1
INTRODUCTION

:

Introduction:

1.1 Domain Overview:

The introduction of advanced Machine Learning (ML) methodologies has revolutionized wealth forecasting in stock markets, presenting a promising avenue for accurate AI-driven predictions. This introduction sets the stage for exploring the transformative potential of ML techniques in predicting stock market trends without specifying particular algorithms. By harnessing the power of ML, this approach aims to revolutionize wealth forecasting, emphasizing the importance of technological advancements, real-time data processing, alternative data sources, and risk management strategies. It also highlights the need for comprehensive financial analysis and ethical considerations in navigating the dynamic and unpredictable landscape of financial markets.

1.2 MACHINE LEARNING

Machine learning is to predict the future from past data. Machine learning (ML) isa type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using Python. The process of trainingand prediction involves the use of specialized algorithms. It feeds the training data to an algorithm, and the algorithm uses this training data to give predictions on new test data.Machine learning can be roughly separated into three categories. There are supervised learning, unsupervised learning, and reinforcement learning.

:

Data scientists use many different kinds of machine learning algorithms to discoverpatterns in Python that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they “learn” about data to make predictions: supervised and unsupervised learning.

Classification predictive modeling is the task of approximating a mapping functionfrom input variables(X) to discrete output variables(y). In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observations.

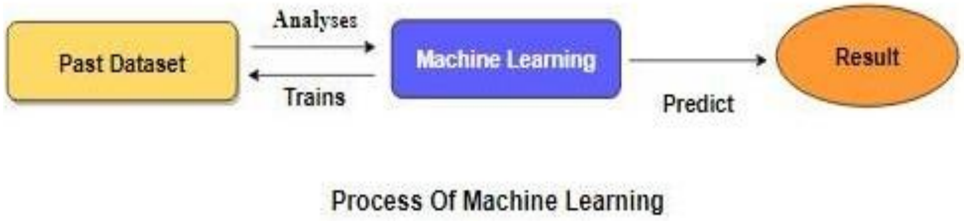


Fig 1.2 Process Flow

Supervised Machine Learning is the majority of practical machine learning uses supervised learning. Supervised learning is where have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from the input to the output is y = f(X). The goal is to approximate the mapping function so well that when youhave new input data (X) that you can predict the output variables (y) for that data. This problem has as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables.

:

The difference between the two tasks is the fact that the dependent attribute is numerical for categorical for classification. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will tryto predict the value of one or more outcomes. A classification problem is when the outputvariable is a category, such as “red” or “blue”.

1.3 ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term mayalso be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving.

Artificial intelligence (AI) is intelligence demonstrated by machines, as opposed tothe natural intelligence displayed by humans or animals. Leading AI textbooks define the field as the study of "intelligent agents" any system that perceives its environment and takes actions that maximize its chance of achieving its goals. Some popular accounts use the term "artificial intelligence" to describe machines that mimic "cognitive" functions thathumans associate with the human mind, such as "learning" and "problem solving", however this definition is rejected by major AI researchers.

The various sub-fields of AI research are centered around particular goals and the use of particular tools. The traditional goals of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception and the abilityto move and manipulate objects. General intelligence (the ability to solve an arbitrary problem) is among the field's long-term goals.

:

The field was founded on the assumption that human intelligence "can be so precisely described that a machine can be made to simulate it". This raises philosophical arguments about the mind and the ethics of creating artificial beings endowed with human-like intelligence.

AI requires a foundation of specialized hardware and software for writing and training machine learning algorithms. No one programming language is synonymous withAI, but a few, including Python, R, and Java, are popular.

In general, AI systems work by ingesting large amounts of labeled training data, analyzing the data for correlations and patterns, and using these patterns to make predictions about future states. In this way, a chatbot that is fed examples of text chats canlearn to produce life-like exchanges with people, or an image recognition tool can learn toidentify and describe objects in images by reviewing millions of examples.

AI programming focuses on three cognitive skills: learning, reasoning and self-correction.

Learning processes. This aspect of AI programming focuses on acquiring data and creating rules for how to turn the data into actionable information. The rules, which are called algorithms, provide computing devices with step-by-step instructions for how to complete a specific task.

Reasoning processes. This aspect of AI programming focuses on choosing the right algorithm to reach a desired outcome.

Self-correction processes. This aspect of AI programming is designed to continually fine-tune algorithms and ensure they provide the most accurate results possible.

:

CHAPTER 2
LIRTERATURE SURVEY

:

2.1 Review of the Literature Survey

1.Title: Parameters For Stock Market Prediction

Author: Prashant S., Chavan,Prof. , Dr.Shrishail. T.Patil

Year: 2018

Parameters for Stock Market Prediction" by Prashant S. Chavan and Prof. Dr. Shrishail T. Patil is a book that provides insights into the microeconomic factors influencing stock market movements. It covers foundational microeconomic concepts such as supply and demand, pricing mechanisms, and market efficiency. The book likely delves into company analysis, including how to assess financial statements, profitability, cost structures, and competitive positioning within industries. Readers can expect discussions on analyzing industry trends, technological disruptions, regulatory environments, and their impacts on stock prices. Economic indicators such as GDP growth rates, inflation, interest rates, and employment figures are likely explored for their relevance to stock market predictions.

2.Title: A LSTM-based method for stock returns prediction: A case study of China stock market.

Author: Kai Chen,Yi Zhou and FangyanDai

Year: 2022

A LSTM-based Method for Stock Returns Prediction: A Case Study of China Stock Market" by Kai Chen, Yi Zhou, and Fangyan Dai presents an approach using Long Short-Term Memory (LSTM) networks for predicting stock returns. LSTM networks are highlighted for their ability to capture long-term dependencies, particularly useful in time series prediction tasks like stock market forecasting. The paper likely details the training process of the LSTM network using historical China stock market data, aiming to predict future returns.

:

3.Title: Detection of statistical arbitrage using machine learning techniques in Indian Stock market.

Author: A.U.S.S Pradeep,Soren Goyal, J. A.Bloom, I. J.Cox, and M.Miller

Year: 2021

Detection of Statistical Arbitrage using Machine Learning Techniques in Indian Stock Market" authored by A.U.S.S Pradeep, Soren Goyal, J.A. Bloom, I.J. Cox, and M. Miller, published in 2021, likely explores the application of machine learning for detecting statistical arbitrage opportunities in the Indian stock market. The paper probably discusses the methodology of employing machine learning algorithms to analyze historical market data and identify instances where pricing discrepancies may lead to arbitrage opportunities. Expect an examination of various machine learning techniques such as decision trees, random forests, or neural networks, and their effectiveness in this context.

4.Title:Prediction of Stock Market Using Artificial Neural Network

Author: Neelima Budhani,Dr. C. K.Jha,Sandeep K.Budhani

Year: 2022

Prediction of Stock Market Using Artificial Neural Network" by Neelima Budhani, Dr. C.K. Jha, and Sandeep K. Budhani is likely a study focusing on using Artificial Neural Networks (ANNs) for stock market prediction. This paper, authored by the mentioned researchers, might delve into the application of ANNs, a type of machine learning algorithm inspired by the human brain's structure, to forecast stock market movements. Expect discussions on how ANNs can analyze historical market data to identify patterns and trends for predictive purposes.

:

2.2 Feasibility study

Feasibility study in the sense it's a practical approach of implementing the proposed model of system . Here for a machine learning projects .we generally collect the input from online websites and filter the input data and visualize them in graphical format and then the data is divided for training and testing . That training is testing data is given to the algorithms to predict the data .

1. In the first place, we take cervical dataset.
2. Channel dataset as per prerequisites and make a new dataset which has trait as indicated by examination to be finished
3. Perform Pre-Handling on the dataset
4. Divide the information into preparing and testing
5. Train the model with preparing information then examine testing dataset over characterization calculation
6. At long last you will come by results as exactness measurements.
7. Atlast we will making site page utilizing jar bundle with pickle python library record.

Algorithms

- Decision tree

2.3 Decision tree:

The Decision Tree algorithm is one of the numerous calculations that work on the idea of administered learning. This calculation can be utilized to settle both relapse and grouping based use cases. It performs magnificently when utilized in characterization based errands overall and produces a tree-put together construction with respect to the dataset. As this calculation for the most part settles on choices in view of specific factors that it views as significant, it is very appealing to the human perspective while going with genuine choices. The rationale behind the choices can

:

likewise be effortlessly perceived because of the tree-like construction that the calculation gives.

·Relapse is utilized when the information that we have and the information that we are making expectations on are nonstop.

·Order is utilized when the information we have and the information that we are making expectations on are discrete(or)categorical.

How about we figure out the importance of discrete and nonstop information. Have you ever known about something like "The framework has half imperfection"? No right!. A framework either has a total imperfection or no deformity by any means. There are no parts, quarters, or portions. This sort of information is called Discrete information ex: Yes/no, Great/awful, and so forth. Just a limited number of values are conceivable, and the qualities can't be partitioned further.

1.Root Hub: The root hub is the absolute first hub in the tree. This hub starts the entire dynamic cycle by further getting partitioned into at least 2 arrangements of hubs and every hub comprises of a component in the informational collection.

2.Splitting: A hub is separated into sub-hubs at each level and this cycle is called parting.

3.Decision Hub: When a hub can be separated into at least 2 hubs then, at that point, it's known as a choice hub. This separating occurs as indicated by the quantity of various choices that can be produced using that specific hub.

4.Leaf/Terminal Hub: Hubs that can't be separated into sub-hubs any longer.

5.Pruning: This is the most common way of eliminating an undesirable piece of a tree. To be exact it tends to be a hub (or) a branch in the tree as well.

6.Branch/Sub-Tree: When a tree is parted into various sub-parts it is known to be a branch in a tree (or) a subtree.

7.Parent and Youngster Hub: When a hub is partitioned into sub-hubs the sub-hubs are known as the kid hubs and the hub that got parted is known as the parent hub of this large number of kid hubs

:

CHAPTER 3
THEORETICAL BACKGROUND

:

3.1 Implementation Environment:

Anaconda Package versions are been managed by the package managements system “Conda”. The Anaconda distribution is used by over 12 million users and includesmore than 1400 popular data science packages suitable for Windows, Linux, and MacOS.So, Anaconda distribution comes with more than 1,400 packages as well as the Conda package and virtual environment manager called Anaconda Navigator and it eliminates the need to learn to install each library independently. The open source packagescan be individually installed from the Anaconda repository with the condo install command or using the pip install command that is installed with Anaconda. Pip packages provide many of the features of condo packages and in most cases they can work together. Custom packages can be made using the `conda build` command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7.However, you can create new environments that include any version of Python packaged with conda.

ANACONDA NAVIGATOR

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage condo packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository.

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing,

:

predictive analytics, etc.), that aims to simplify package management and deployment. In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.

The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly

The best way to get started using Python for machine learning is to complete a project.

- It will force you to install and start the Python interpreter (at the very least).
- It will give you a bird’s eye view of how to step through a small project.
- It will give you confidence, maybe to go on to your own small projects.

FLASK FRAMWORK

Flask is a lightweight and flexible Python web framework designed to make web development simple and straightforward. It follows a minimalist philosophy, providing just the essentials for building web applications. Developers appreciate Flask for its simplicity, making it easy to get started with web projects. The framework offers a robust set of features including routing, request handling, and templating. Flask is known for its modular design, allowing developers to add extensions for specific functionalities as needed. It supports various web development tasks such as URL routing, HTTP request handling, and rendering templates.

:

3.2 Existing System:

The existing systems for wealth forecasting in stock markets involve statistical, quantitative, technical, and fundamental analysis, complemented by expert opinions and news analysis. However, these methods face challenges in adapting to market complexities, unexpected events, and may struggle to consistently outperform the market due to the efficient market hypothesis. There's a growing need to explore more advanced Machine Learning (ML) techniques and diverse data integration to enhance accuracy and adaptability in predicting market trends.

Disadvantages:

- Limitations in Adaptability
- Complexity in Predictive Accuracy
- Risk of Financial Loss
- Inadequate Integration of Advanced Techniques

3.3 Proposed System:

The proposed system for improved wealth forecasting in stock markets integrates advanced Machine Learning (ML) techniques, alternative data sources, real-time analysis, and ethical compliance. By employing sophisticated ML algorithms, embracing diverse datasets, ensuring adaptability to changing market conditions, and focusing on transparency and ethical considerations, the system aims to enhance accuracy and reliability in predicting market trends and wealth forecasting.

Advantages:

- Enhanced Accuracy
- Improved Decision-making
- Enhanced Safety
- Early Detection
- Real-time Data Processing

:

Block Diagram:

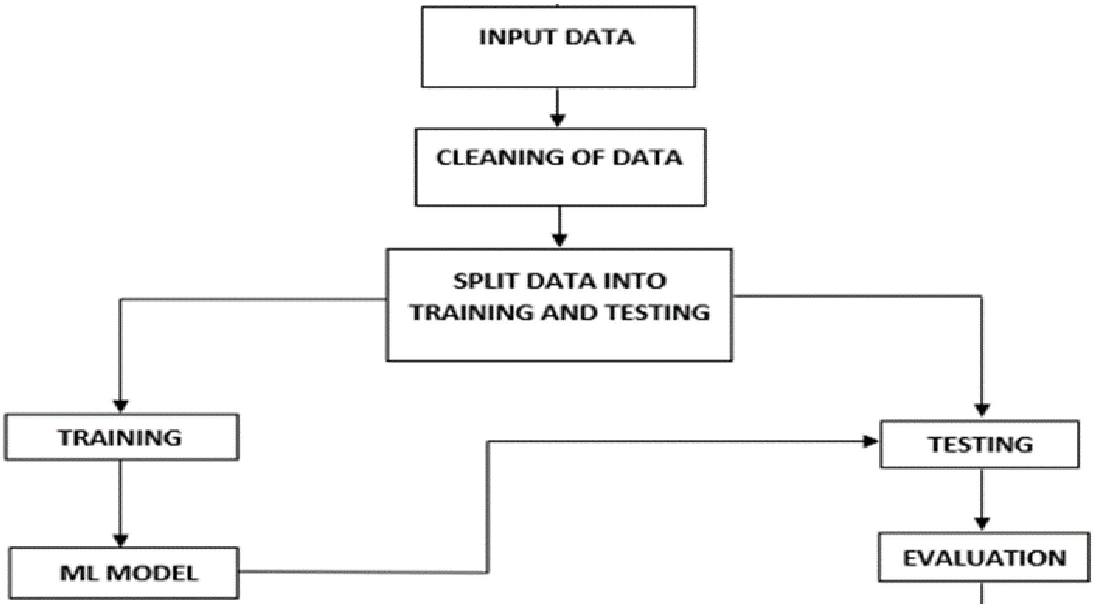


Fig 3.2 Block Diagram

Software and Hardware Requirements:

Hardware:

- OS – Windows 7, 8 and 10 (32 and 64 bit)
- RAM – 4GB
- Database

Software:

- Python / Anaconda Navigator
- Packages: numpy, Pandas, matplotlib, Sklearn
- Flask Framework

:

CHAPTER 4
SYSTEM DESIGN

:

SYSTEM DESIGN

4.1 System Archietecture:

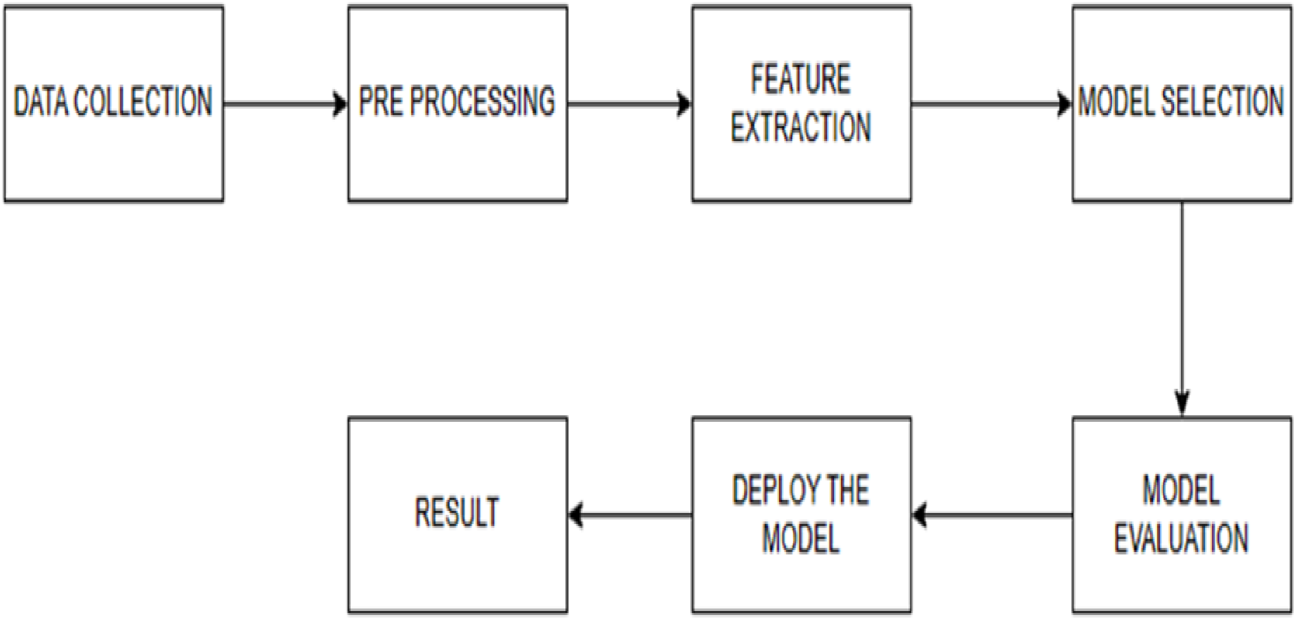


Fig 4.1 System Architecture

DATA GATHERING

This paper's information assortment comprises of various records. The determination of the subset of all open information that you will be working with is the focal point of this stage. Preferably, ML challenges start with a lot of information (models or perceptions) for which you definitely know the ideal arrangement. Marked information will be data for which you are as of now mindful of the ideal result.

PRE-PROCESSING OF DATA

Format, clean, and sample from your chosen data to organise it.

There are three typical steps in data pre-processing:

:

1.Designing

2.Information cleaning

3.Inspecting

Designing:

It's conceivable that the information you've picked isn't in a structure that you can use to work with it. The information might be in an exclusive record configuration and you would like it in a social data set or text document, or the information might be in a social data set and you would like it in a level document.

Information cleaning:

It is the most common way of eliminating or supplanting missing information. There can be information examples that are inadequate and come up short on data you assume you really want to resolve the issue. These events could should be eliminated.Moreover, a portion of the traits might contain delicate data, and it very well might be important to antonymize or totally eliminate these properties from the information.

Inspecting:

You might approach significantly more painstakingly picked information than you want. Calculations might take significantly longer to perform on greater measures of information, and their computational and memory prerequisites may likewise increment. Prior to considering the whole datasets, you can take a more modest delegate test of the picked information that might be fundamentally quicker for investigating and creating thoughts.

FEATURE EXTRACTION

The following stage is to A course of quality decrease is include extraction. Highlight extraction really modifies the traits instead of element choice, which positions the

:

ongoing ascribes as indicated by their prescient pertinence. The first ascribes are straightly joined to create the changed traits, or elements. Finally, the Classifier calculation is utilized to prepare our models. We utilize the Python Normal Language Tool stash's classify module.

We utilize the gained marked dataset. The models will be surveyed utilizing the excess marked information we have. Pre-handled information was ordered utilizing a couple of AI strategies. Irregular woodland classifiers were chosen. These calculations are generally utilized in positions including text grouping.

_MODEL EVALUATION

Model The method involved with fostering a model incorporates assessment. Finding the model that best portrays our information and predicts how well the model will act in what's to come is useful. In information science, it isn't adequate to assess model execution utilizing the preparation information since this can rapidly prompt excessively hopeful and overfitted models. Wait and Cross-Approval are two procedures utilized in information science to evaluate models.

4.2 Data Flow Diagrams

The abbreviation for Information Stream Outline is DFD. DFD deals with the information flow of a framework or an interaction. Additionally, it provides information on the data sources, outcomes, and actual interactions for each factor. There are no circles, choice criteria, or control streams in DFD. A flowchart can make sense of explicit tasks reliant on the type of data.

It is a graphical tool that makes communicating with clients, supervisors, and other faculty members easier. It is useful for dissecting both the current and the suggested structure.

It provides a summary of the framework procedures for what information there is.

:

What adjustments are made, what data is archived, what outputs are made, and so on.

There are various ways to address the Information Stream Outline. There are organised investigation exhibiting gadgets at the DFD. Information Stream graphs are well known because they let us visualise the important steps and information involved in programming framework activities.

Four sections make up the information stream graph:

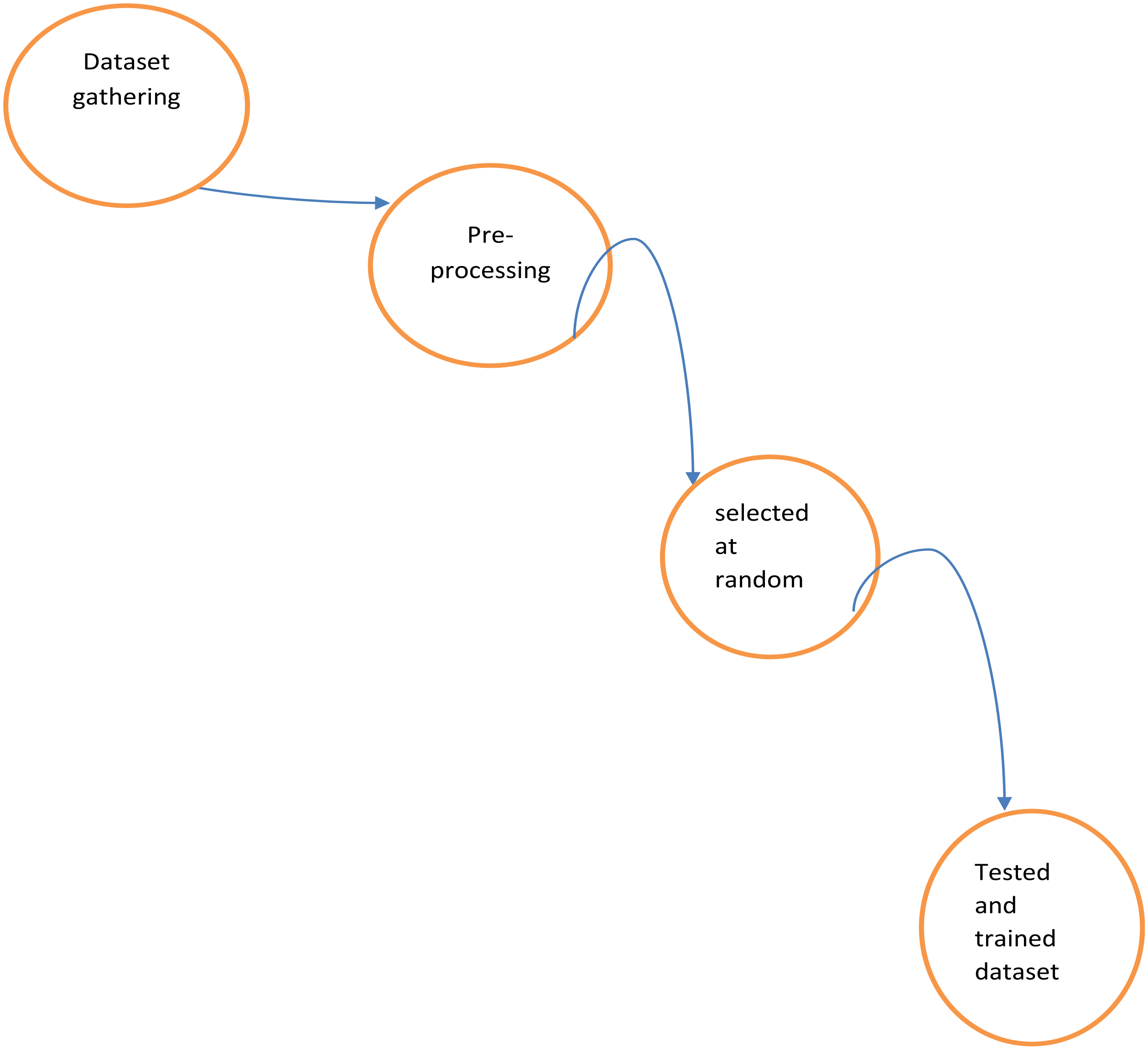
Process Due to dealing capability, a framework's ability to produce change is affected. Images of a conversation may be round, oval, square, or rectangular with rounded corners. The cycle is given a brief name that communicates its essence in a single word or statement.

Information Stream Information stream depicts the data moving between various pieces of the frameworks. The bolt image is the image of information stream. An engaging name ought to be given to the stream to decide the data which is being moved. Information stream additionally addresses material alongside data that is being moved. Material movements are displayed in frameworks that are not simply useful. A given stream ought to just exchange a solitary kind of data. The course of stream is addressed by the bolt which can likewise be bi-directional.

Distribution center The information is put away in the stockroom for sometime in the future. Two flat lines address the image of the store. The distribution center is essentially not confined to being an information record rather it very well may be in any way similar to an envelope with reports, an optical circle, a file organizer. The information distribution center can be seen autonomous of its execution. At the point when the information stream from the stockroom it is considered as information perusing and when information streams to the distribution center it is called information passage or information refreshing.

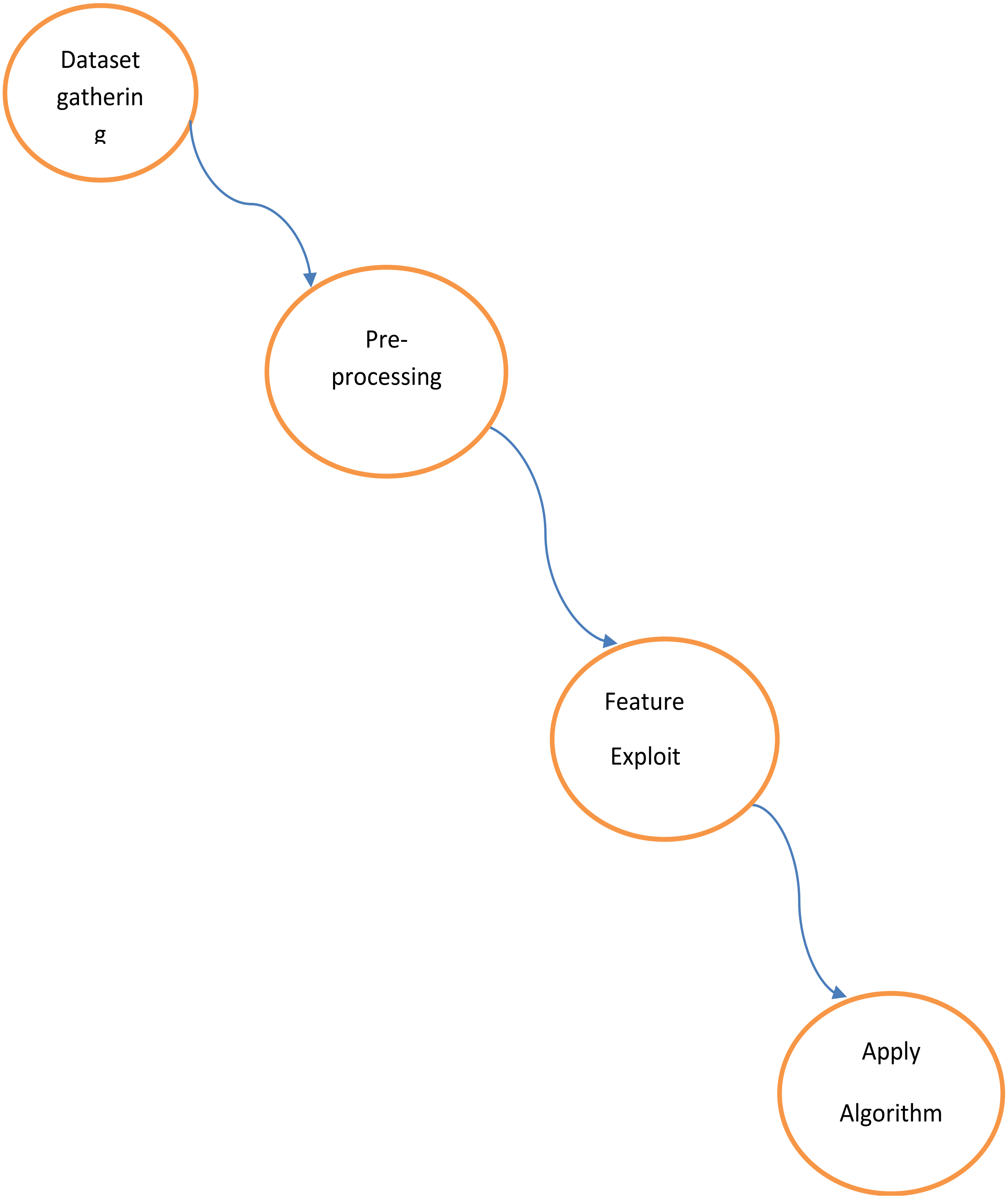
:

LEVEL 0


Dataset gathering
Pre-processing
selected at random
Tested and trained dataset

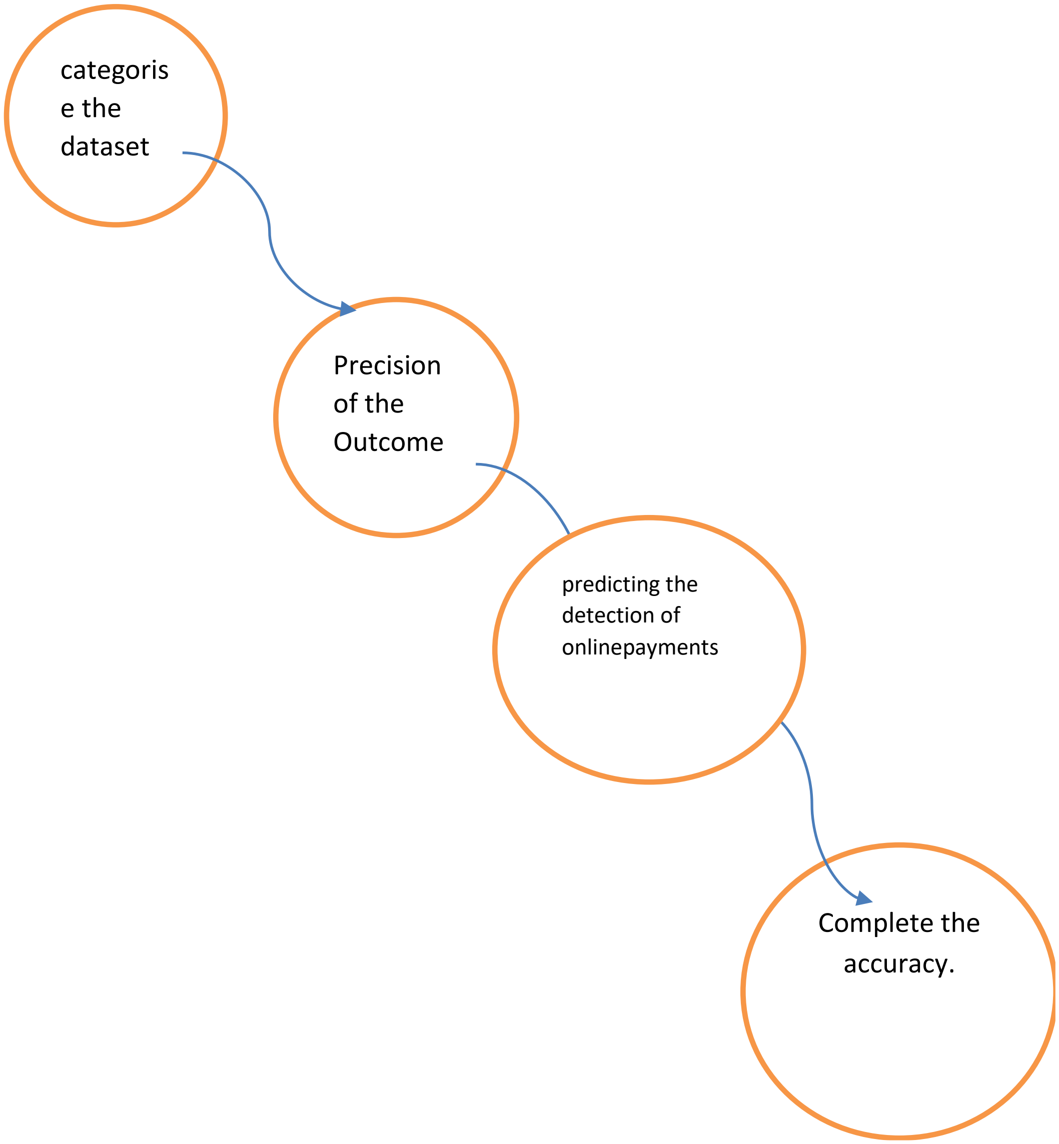
:

Level 1:


Dataset
gatherin
g
Pre-
processing
Feature
Exploit
Apply
Algorithm

:

Level 2:


categorise the dataset
Precision of the Outcome
predicting the detection of onlinepayments
Complete the accuracy.

:

UML DIAGRAMS

The United Showing Language (UML) is used to decide, imagine, modify, construct and record the relics of an article arranged programming heightened system a work underway. UML offers a standard strategy for envisioning a system's designing charts, including parts, for instance,

- performers
- business processes
- (authentic) parts
- works out
- programming language clarifications
- informational collection frames, and
- Reusable programming parts.

UML unites best methods from data illustrating (substance relationship diagrams), business showing (work processes), object illustrating, and part illustrating. It will in general be used with all cycles, all through the item improvement life cycle, and across different execution developments. UML has integrated the documentations of the Booch system, the Thing showing technique (OMT), and Thing arranged PC programming (OOSE) by consolidating them into a single, ordinary, and comprehensively usable exhibiting language. UML expects to be a standard showing language which can show concurrent and conveyed systems.

Grouping Graph:

Grouping Graphs Address the articles taking part the communication evenly and time upward. A Utilization Case is a sort of social classifier that addresses a statement of an offered conduct. Each utilization case determines some way of behaving, conceivably including variations that the subject can act in a joint effort with at least one entertainers. Use cases characterize the offered conduct of the subject without reference to its inner design. These ways of behaving, including connections between the entertainer and the

:

subject, may bring about changes to the condition of the subject and correspondences with its current circumstance. A utilization case can incorporate potential varieties of its essential way of behaving, including excellent way of behaving and blunder dealing with.

Action Graphs:

Action outlines are graphical representations of the sequential exercises and activities that aid in decision-making, emphasis, and simultaneity. Movement charts can be used to represent the business and functional piece-by-piece work processes of parts in a framework in the Brought together Displaying Language. The general evolution of control is displayed in an action graph.

Usecase summary:

The standard language for describing, visualising, creating, and documenting the peculiarities of programming frameworks is called UML.

•The Organisation for Management and Guidance (OMG) created UML, and a specific draught of UML 1.0 was presented to the OMG in January 1997.

•OMG continually expends effort to create a true industry standard.

•UML stands for Combined Displaying Language.A visual language called UML is used to create programme blueprints.

class diagram

The main building component of an item-arranged display is the class outline. It is used for both specific exhibiting making an interpretation of the models into programming code, as well as for general reasonable demonstrating of the effectiveness of the application. Information modelling can also make use of class outlines.[1] A class chart's classes cover the application's linkages, core elements, and classes that need to be updated.

Classes are referenced in the outline with boxes that have three compartments each:

The class name is located in the top compartment. It is strongly and narrowly concentrated, and the main letter is emphasised.

:

4.3 Use Case Diagram

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

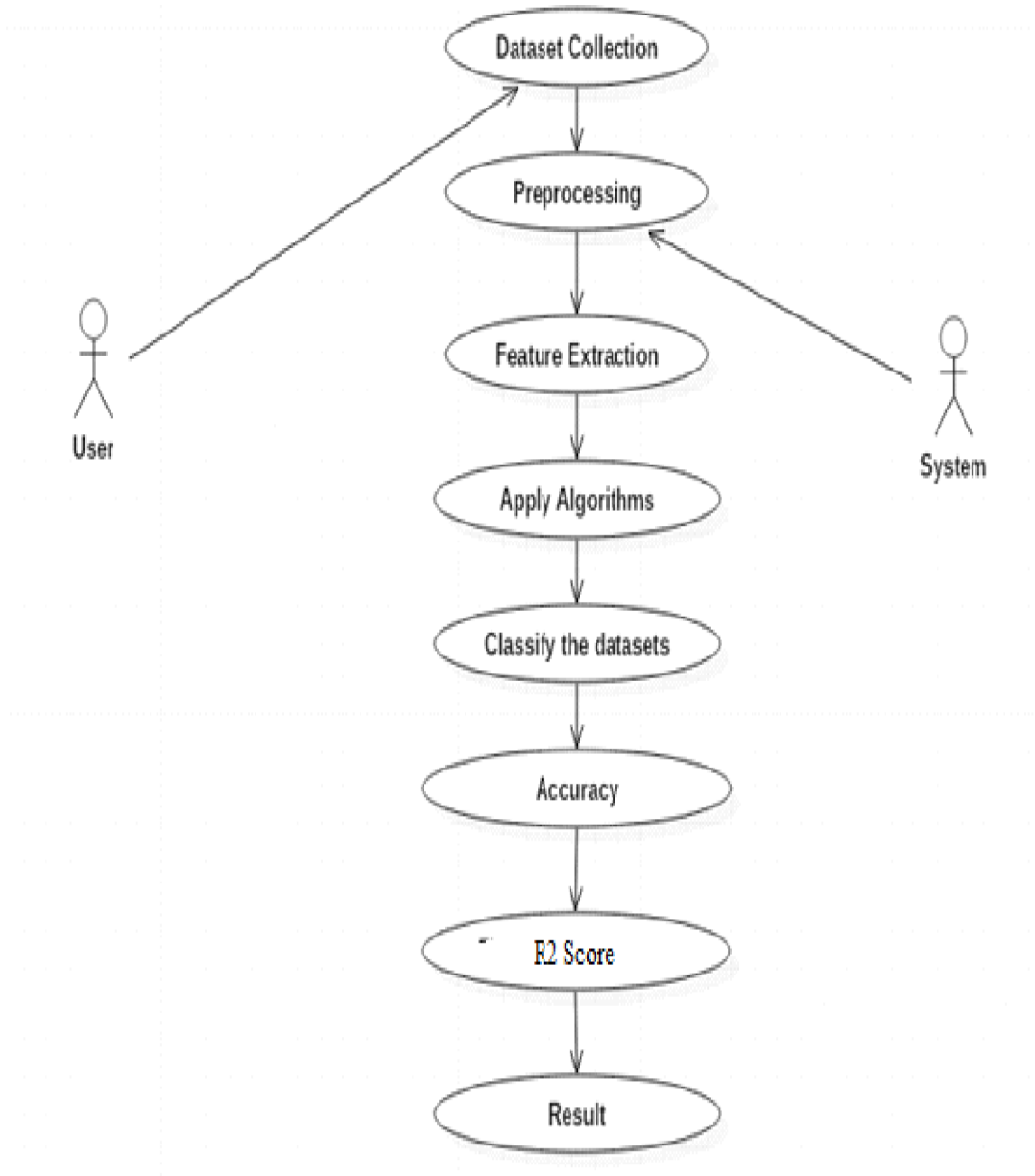


Fig 4.3 Use Case Diagram

:

4.4 Class Diagram

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified and because, unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the end of the drawing it should be understandable to the developer/coder.

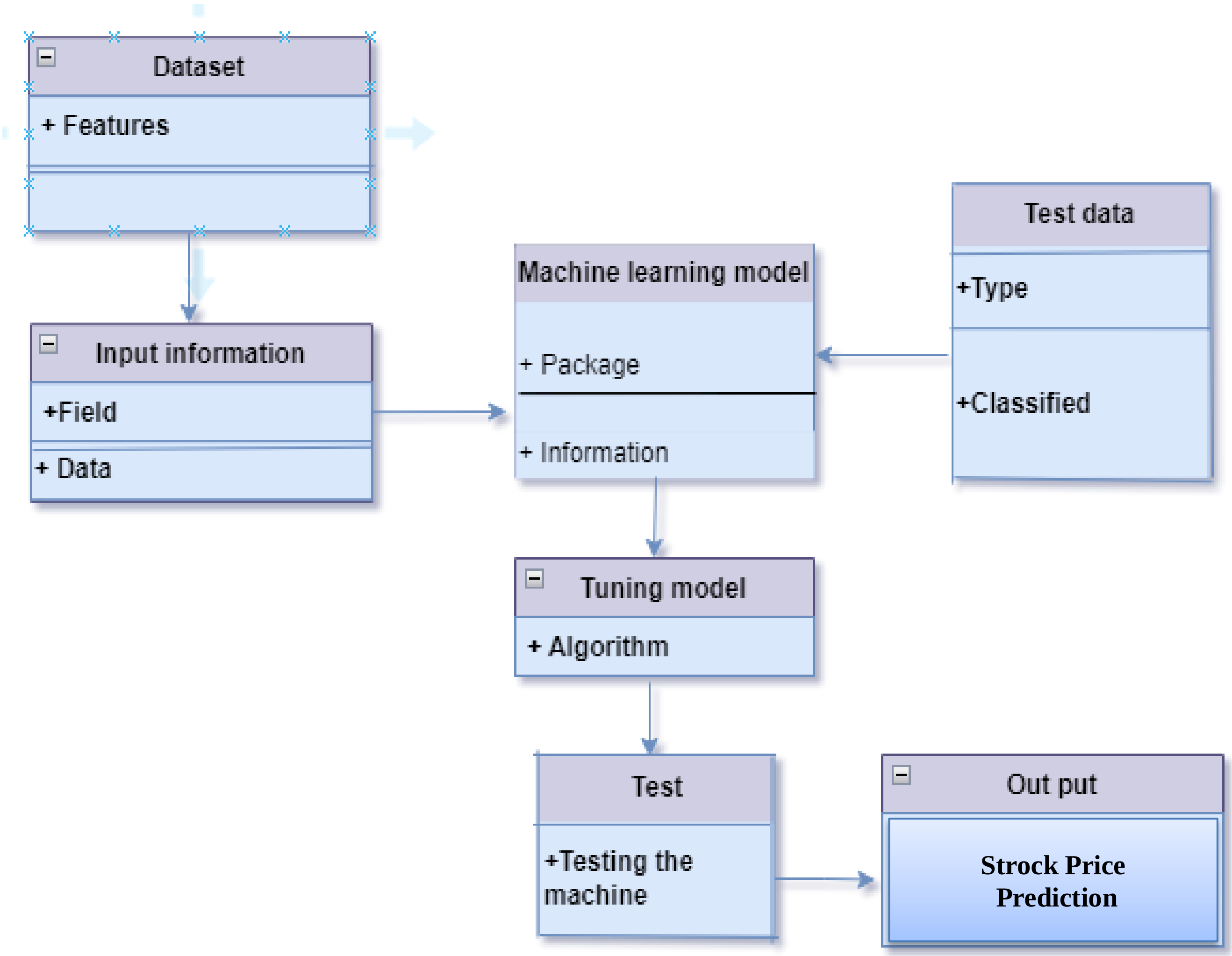


Fig 4.4 Class Diagram

:

4.5 Sequence Diagram

A sequence diagram is a type of interaction diagram in Unified Modeling Language (UML) that illustrates how objects interact in a particular scenario of a system or application. It shows the flow of messages, or interactions, between objects over time, typically from top to bottom. Each object involved in the scenario is represented by a vertical lifeline, with messages between objects shown as horizontal arrows. These messages indicate the order and direction of communication between objects, helping to visualize the sequence of actions in a system. Sequence diagrams are valuable for understanding the dynamic behavior of a system, including the order of method calls, collaborations between objects, and the flow of control.

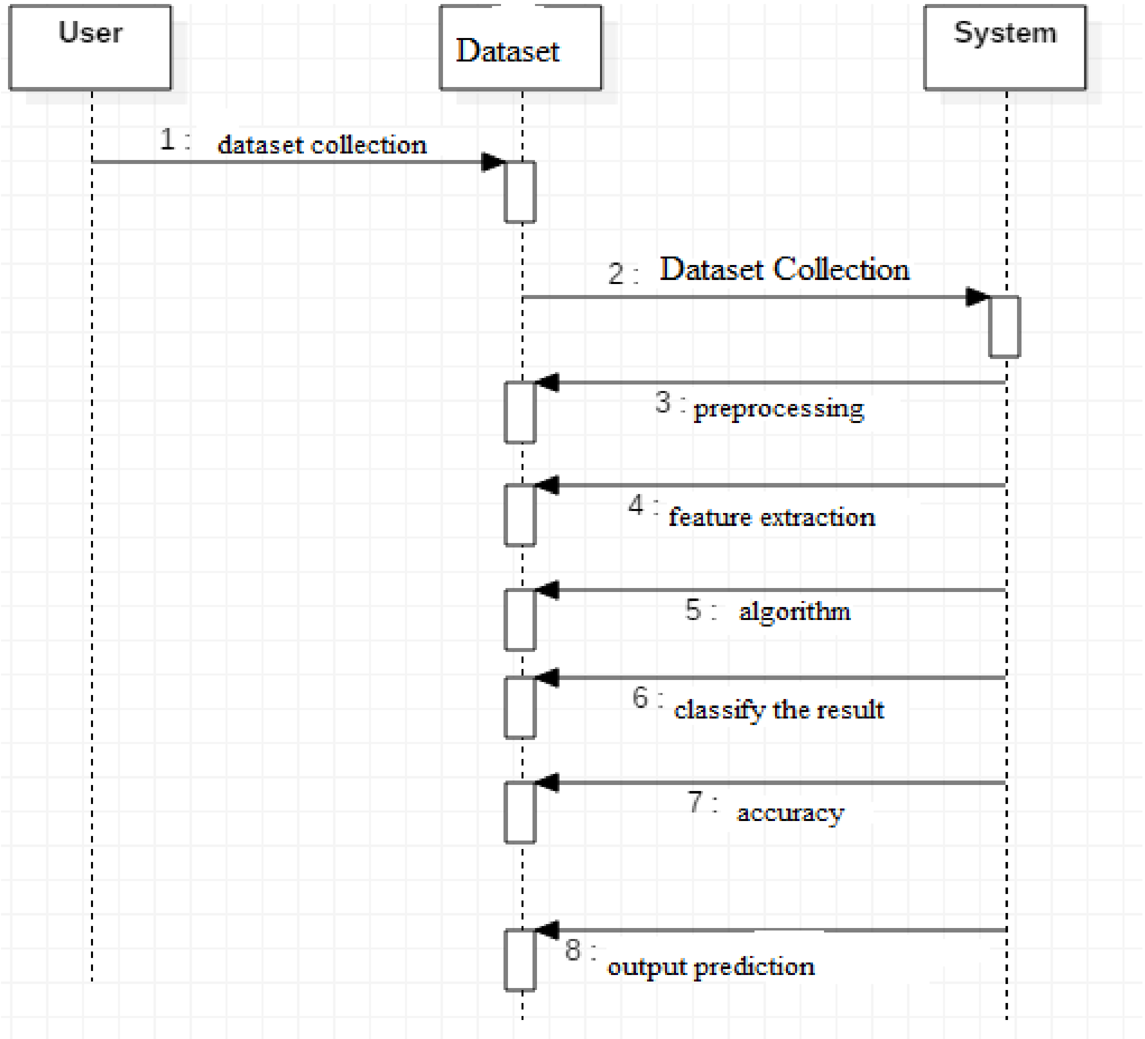


Fig 4.5 Sequence Diagram

:

4.6 Activity Diagram

Activity is a particular operation of the system. Activity diagrams are not only usedfor visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missingthing in activity diagram is the message part. It does not show any message flow from oneactivity to another. Activity diagram is some time considered as the flow chart. Althoughthe diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

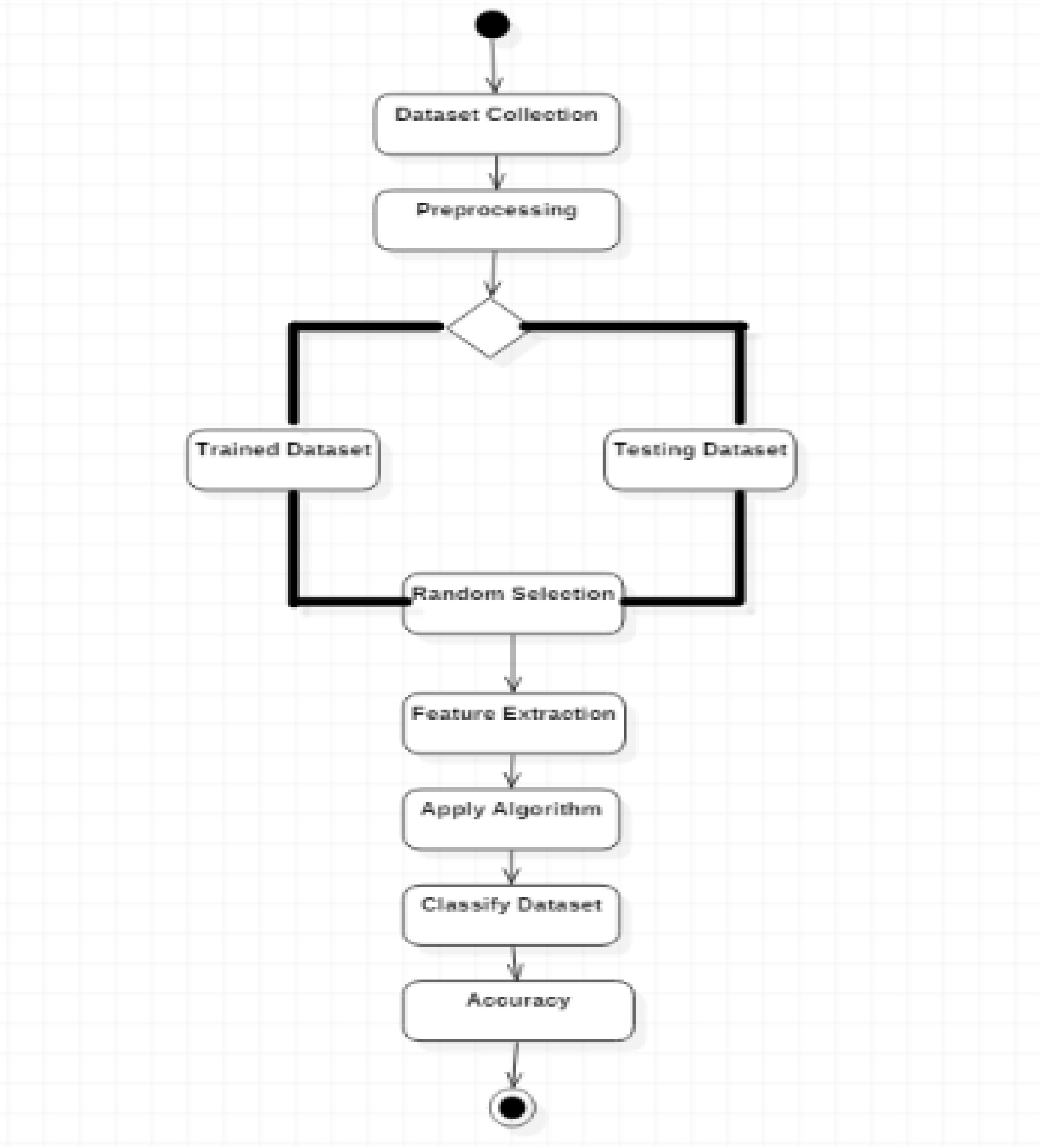


Fig 4.6 Activity Diagram

:

CHAPTER 5
SYSTEM IMPLEMENTATION

:

5.1 MODULES:

DATA GATHERING

This paper's information assortment comprises of various records. The determination of the subset of all open information that you will be working with is the focal point of this stage. Preferably, ML challenges start with a lot of information (models or perceptions) for which you definitely know the ideal arrangement. Marked information will be data for which you are as of now mindful of the ideal result.

PRE-PROCESSING OF DATA

Format, clean, and sample from your chosen data to organise it.

There are three typical steps in data pre-processing:

1.Designing

2.Information cleaning

3.Inspecting

- **Designing:** It's conceivable that the information you've picked isn't in a structure that you can use to work with it. The information might be in an exclusive record configuration and you would like it in a social data set or text document, or the information might be in a social data set and you would like it in a level document.

- **Information cleaning:** is the most common way of eliminating or supplanting missing information. There can be information examples that are inadequate and come up short on data you assume you really want to resolve the issue. These events could should be eliminated..Moreover, a portion of the traits might contain delicate data, and it very well might be important to antonymize or totally eliminate these properties from the information.

:

- **Inspecting:** You might approach significantly more painstakingly picked information than you want. Calculations might take significantly longer to perform on greater measures of information, and their computational and memory prerequisites may likewise increment. Prior to considering the whole datasets, you can take a more modest delegate test of the picked information that might be fundamentally quicker for investigating and creating thoughts.

FEATURE EXTRACTION

The following stage is to A course of quality decrease is include extraction. Highlight extraction really modifies the traits instead of element choice, which positions the ongoing ascribes as indicated by their prescient pertinence. The first ascribes are straightly joined to create the changed traits, or elements. Finally, the Classifier calculation is utilized to prepare our models. We utilize the Python Normal Language Tool stash's classify module.

We utilize the gained marked dataset. The models will be surveyed utilizing the excess marked information we have. Pre-handled information was ordered utilizing a couple of AI strategies. Irregular woodland classifiers were chosen. These calculations are generally utilized in positions including text grouping.

ASSESSMENT MODEL

Model The method involved with fostering a model incorporates assessment. Finding the model that best portrays our information and predicts how well the model will act in what's to come is useful. In information science, it isn't adequate to assess model execution utilizing the preparation information since this can rapidly prompt excessively hopeful and overfitted models. Wait and Cross-Approval are two procedures utilized in information science to evaluate models.

The two methodologies utilize a test set (concealed by the model) to survey model execution to forestall over fitting. In light of its normal, every classification model's

:

presentation is assessed. The result will take on the structure that was envisioned. diagram portrayal of information that has been ordered.

Accuracy: The level of precise expectations for the test information is implied by precision. By partitioning the quantity of exact expectations by the complete number of forecasts, it very well might still up in the air.

5.2 Python Libraries &Packages:

Python provides a variety of GUI (Graphical User Interface) creation options. Tkinter is the most frequently used GUI method out of all of them. It is a typical Python connection point to the Python distribution's Tk GUI tool stack. The quickest and easiest way to create GUI apps is with Python and Tkinter. Tkinter makes creating a GUI an easy task.

A tkinter application can be created by:

I. Introducing the tkinter module
I. Create the basic window (holder)
II. Add a lot of devices to the main window.
III. Apply the event Trigger to the devices.
IV. Tkinter can be inserted into the Python code just like any other module.
V. Note that the module's name changes from "Tkinter" in Python 2.x to "tkinter" in Python 3.x.

When creating a Python application with a GUI, the client needs to keep in mind two main strategies that are used.

Tk(screenName = 'None', base = 'None', class = 'Tk', useTk = 1):

Tkinter provides the method 'Tk(screenName=None, baseName=None, className='Tk', useTk=1)' to create a main window.

You can alter the className to the desired one to change the window's name. The primary code used to create the application's main window is m=tkinter.Tk(),

:

mainloop(), where m is the name of the basic window object.

When your application is ready to run, a technique called as mainloop() is used.
As long as the window is open, the application will run indefinitely throughout the mainloop() loop, which is used to anticipate events and respond to them.

Additionally, tkinter provides access to the mathematical configuration of the devices that can arrange the devices in the parent windows. Fundamentally, there are three main maths classes.
Pack() technique:Prior to inserting the parent device, it organises the devices into blocks.
matrix() technique:Before setting in the parent gadget, it networks the devices together (in a table-like form).
place() technique:It organises the devices by positioning them in clear positions decided upon by the software programmer.

You can include a variety of tools in your tkinter application.

Button: This gadget is used to add a button to your programme.
w=Button(master, option=value) is the general grammar, and ace is the boundary used to refer to the parent window.
The arrangement of the Buttons can be changed using a variety of options. Choices can be passed as boundaries that are separated by commas. The list below includes a few of them.

To set the background tone when the button is under the cursor, use activebackground.
When a button is under the cursor, the activeforeground setting sets the forefront tone.

:

bg: to establish the standard foundational tone.

to call a capability is the order.

textual style: to change the button mark's text style.

image: to place an image on a button.

width: to modify the button's width.

level: to modify the button's level.

NumPy:

The free source Python package NumPy (Mathematical Python) is used in virtually every field of science and design. It serves as the foundation of the logical Python and PyData environments and is the standard practise for working with mathematical data in Python. Clients of NumPy range from novice programmers to seasoned analysts performing top-tier logical and cutting-edge inventive work. Numerous information scientific and logical Python bundles, including Pandas, SciPy, Matplotlib, scikit-learn, and scikit-picture, make extensive use of the NumPy Programming interface. Complex network and exhibit information structures are available in the NumPy library (you'll find out more information about this in later sections). It offers ways to successfully work on ndarray, a homogenous n-layered cluster object.

Numerous numerical operations can be played out on exhibitions using NumPy. It provides a vast library of high level numerical capabilities that are compatible with these displays and frameworks, as well as powerful information designs that are added to Python that assure accurate estimations with clusters and lattices.

The NumPy library's exhibit is its main information design. A matrix of numbers that serves as an exhibit comprises information on the raw data, how to locate a component, and how to interpret a component. It is made up of a lattice of parts that can be filed in

:

various ways.

The components can be expressed as whole numbers, booleans, another exhibit, or analogous types of nonnegative numbers. The number of factors determines where the cluster is located. A tuple of numbers indicating the size of the exhibit along each axis makes up the cluster's state.

Pandas: The cluster dtype is hinted at.

A tuple of pandas, a Python library that provides rapid, adaptive, and expressive information structures designed to enable working with "social" or "marked" information simple and natural, can be filed into a cluster. It intends to serve as the primary major level building piece for conducting practical, accurate information analysis in Python.

Additionally, it aspires to become the most remarkable and flexible open source information examination/control tool obtainable in any language. It has already made significant progress in achieving this goal.

Primary Components

Size variability: portions can be inserted in and erased from DataFrame and higher layered objects Simple treatment of missing information (addressed as NaN, NA, or NaT) in drifting point information as well as non-drifting point information

Programmed and explicit information organisation: Items can be explicitly changed to a group of names, or the client can merely ignore the markings and let Series, DataFrame, and other similar entities subsequently alter the information for calculations.

Make it simple to convert battered, unexpectedly listed information in other Python and NumPy information structures into DataFrame objects. Strong, adaptable collection by usefulness to perform split-apply-join procedure on informational collections, for both aggregating and changing information.

Pip intall pandas

:

Matplotlib:

A remarkable Python representation library for 2D cluster plots is Matplotlib. Based on NumPy clusters, Matplotlib is a multi-stage information perception library designed to operate with the more comprehensive SciPy stack. In the year 2002, John Tracker delivered the speech. The ability to visually access vast amounts of information in delicious pictures is one of perception's biggest benefits. Several plots, including line, bar, scatter, histogram, and others, are included in Matplotlib. Matplotlib and the great majority of its conditions are available as wheel bundles for Windows, Linux, and macOS.

Run the supplemental command to launch the matplotlib bundle:

```
matplotlib - mpip introduction - python
```

Seaborn:

A matplotlib-based information perception library called Seaborn is tightly integrated with Python's pandas data structures. The main component of Seaborn is perception, which aids in information research and comprehension. To learn about Seaborn, one needs be knowledgeable in Numpy, Matplotlib, and Pandas.

Seaborn provides the following features:

Programming interface with a dataset to determine the relationships between variables.

Straight relapse plots are assessed and plotted using a programme.

For multi-plot frameworks, it supports important level debates.

Imagining the dispersion of univariate and bivariate data.

The procedure used to establish the Seaborn library is as follows:

```
sns import seaborn
```

Here are a handful of well-known plotting libraries to get an outline:

1.Matplotlib: accessible and full of opportunities

:

2. Pandas Perception is a user-friendly interface built on Matplotlib.

3. Seaborn: outstanding default styles, important level connection point

4.ggplot: makes use of the Syntax of Illustrations in R's ggplot2

5.Plotly: able to create clever plots

In this post, we'll learn how to use Matplotlib, Pandas Perception, and Seaborn to create basic plots and how to make use of a few specific features from each library. Instead of attempting to interpret the schematics, this essay will focus just on the punctuation.

Line Outline

In Matplotlib we can make a line outline by calling the plot strategy. We can likewise plot different sections in a single diagram, by circling through the segments we need, and plotting every section on a similar hub.

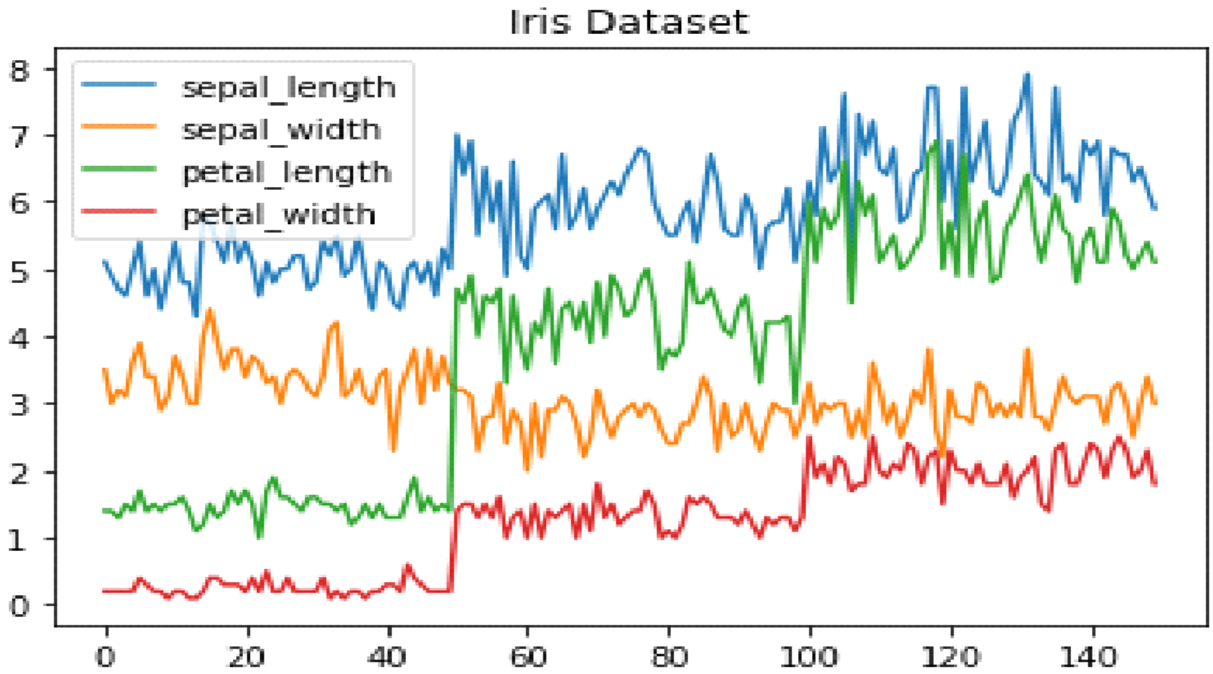


Fig 5.2.1 Line Chart

:

Histogram:

In Matplotlib we can make a Histogram utilizing the hist technique. Assuming that we pass it downright information like the focuses segment from the wine-survey dataset it will naturally ascertain how frequently each class happens.

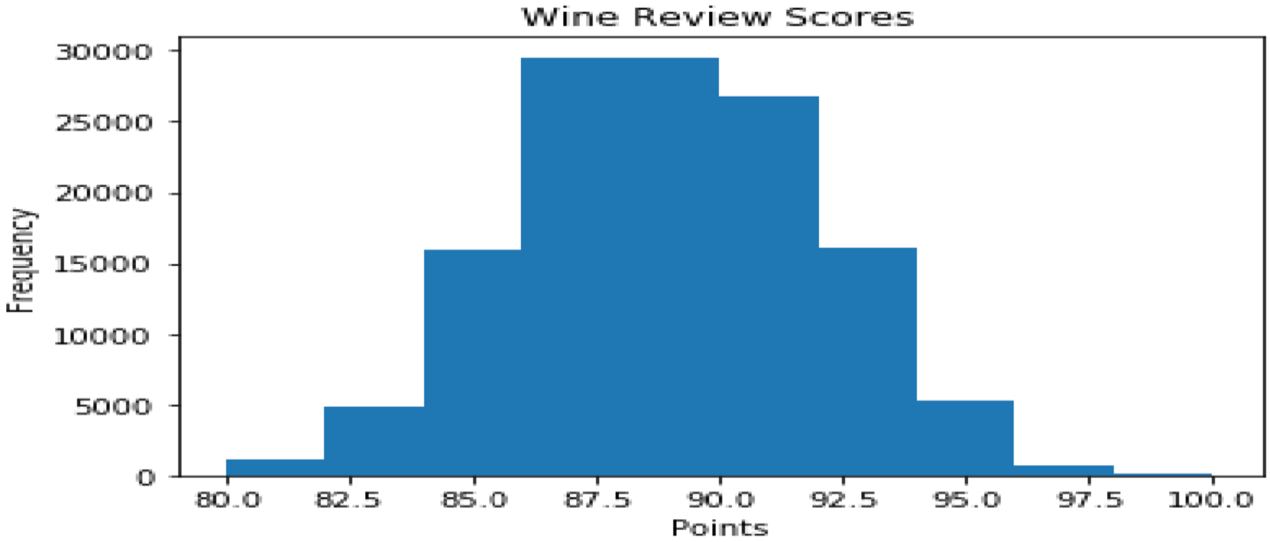


Fig 5.2.2 Histogram

Bar Diagram:

A bar-diagram can be made utilizing the bar strategy. The bar-outline isn't consequently computing the recurrence of a classification so we will utilize pandas value_counts capability to do this. The bar-outline is valuable for downright information that has relatively little various classifications (under 30) in light of the fact that else it can get very chaotic.

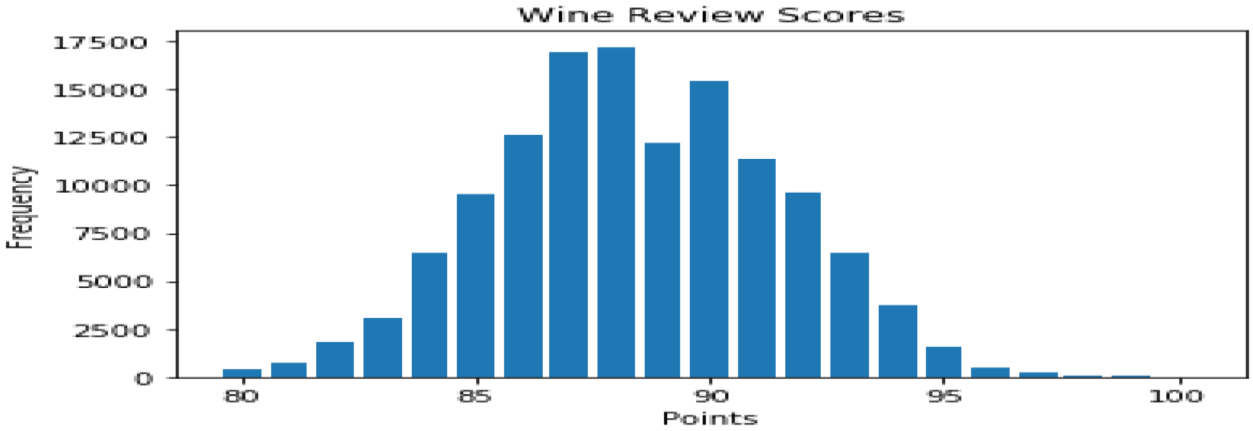


Fig 5.2.3 Bar Chart

:

Pandas Visualization:

Pandas is an open source superior execution, simple to-utilize library giving information structures, for example, dataframes, and information examination devices like the representation instruments we will use in this article.

Pandas Perception makes it truly simple to make plots out of a pandas dataframe and series. It likewise has a more elevated level Programming interface than Matplotlib and hence we want less code for similar outcomes.

●Pandas can be introduced utilizing either pip or conda.

●pip introduce pandas

●conda introduce pandas

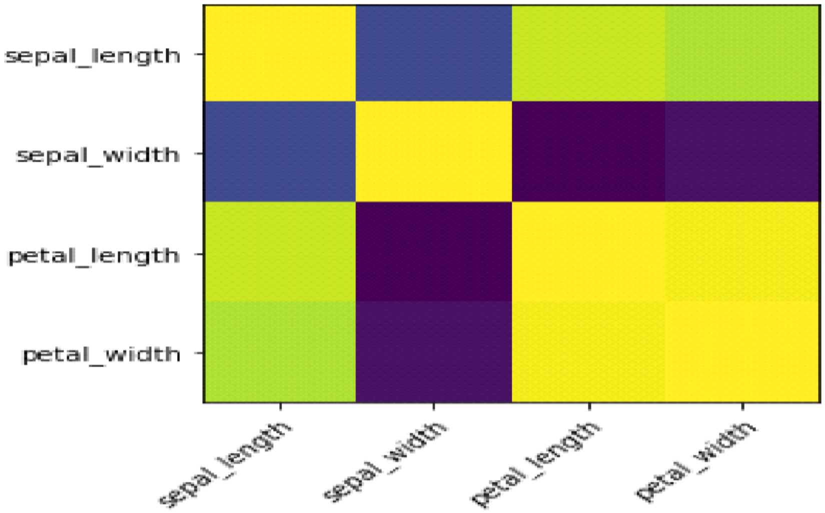
Heatmap:

A Heatmap is a graphical portrayal of information where the singular qualities contained in a lattice are addressed as varieties. Heatmaps are ideally suited for investigating the connection of elements in a dataset.

To get the relationship of the highlights inside a dataset we can call <dataset>.corr() , which is a Pandas dataframe strategy. This will give utilize the connection grid.

We can now utilize either Matplotlib or Seaborn to make the heatmap.

Matplotlib:



:

Heatmap without annotations

Information perception is the discipline of attempting to comprehend information by setting it in a visual setting, so that examples, patterns and relationships that could not in any case be recognized can be uncovered.

Python offers various incredible diagramming libraries that come loaded with bunches of various highlights. In this article we checked out at Matplotlib, Pandas representation and Seaborn.

Scikit-learn:

Scikit-learn is an open source information examination library, and the best quality level for AI (ML) in the Python biological system

Ideas And Highlights:

Characterization: distinguishing and ordering information in light of examples

Relapse: anticipating or projecting information values in light of the typical mean of existing and arranged information.

Bunching: programmed gathering of comparable information into datasets.

.

Interoperability with NumPy, pandas, and matplotlib libraries.

ML is an innovation that empowers PCs to gain from input information and to fabricate/train a prescient model without unequivocal programming. ML is a subset of Man-made consciousness (simulated intelligence).

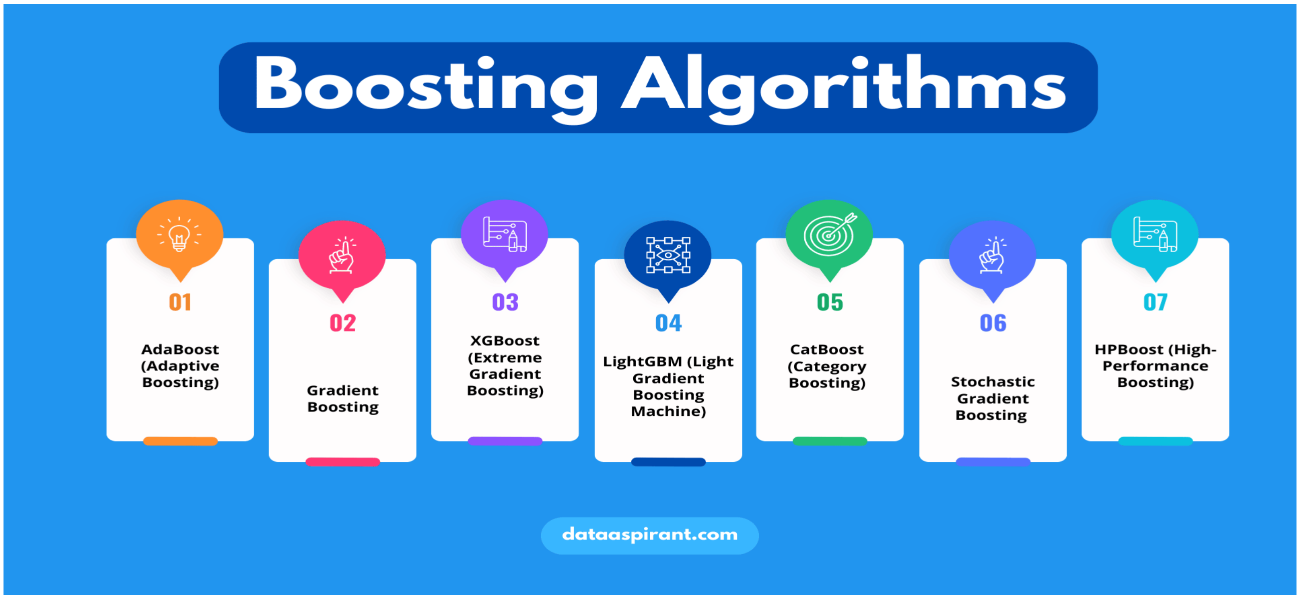
Whether you are only searching for a prologue to ML, need to make ready quick, or are searching for the most recent ML research instrument, you will find that scikit-learn is both factual and simple to learn/use. As a significant level library, it allows you to characterize a prescient information model in only a couple of lines of code, and afterward utilize that model to accommodate your information. It's adaptable and coordinates well with other Python libraries, for example, matplotlib for plotting, numpy for cluster vectorization, and pandas for information outlines.

:

5.3 Algorithms:

Boosting Algorithm:

Boosting algorithms are a class of machine learning techniques that aim to enhance the predictive performance of models by combining the strengths of multiple weak learners. Weak learners, often simple decision trees, are sequentially trained on the data, with each subsequent learner focusing on the mistakes of its predecessors. The key idea is to assign higher weights to misclassified instances, thereby emphasizing their importance in subsequent iterations. Gradient Boosting, AdaBoost, and XG Boost are popular boosting algorithms. Gradient Boosting optimizes model errors by minimizing the gradient of the loss function, AdaBoost assigns varying weights to instances based on their classification success, and XG Boost extends these concepts with regularization and parallel processing capabilities. Boosting algorithms excel in improving predictive accuracy and are widely utilized in diverse machine learning applications.


Boosting Algorithms
01
AdaBoost (Adaptive Boosting)
02
Gradient Boosting
03
XGBoost (Extreme Gradient Boosting)
04
LightGBM (Light Gradient Boosting Machine)
05
CatBoost (Category Boosting)
06
Stochastic Gradient Boosting
07
HPBoost (High-Performance Boosting)
dataaspirant.com

:

Boosting algorithms, like Gradient Boosting or AdaBoost, can be used to predict stock price movements. These algorithms work by combining multiple weak learners (usually decision trees) into a strong learner. Each weak learner corrects the errors of its predecessor, gradually improving prediction accuracy. In the context of stock prices, boosting algorithms can learn complex patterns and relationships in the data, helping to forecast price movements more accurately. Trained on historical price data and relevant features, boosting models can provide insights into potential stock price deductions, indicating when prices might decrease based on the learned patterns. These algorithms are popular in financial modeling due to their ability to handle non-linear relationships and adapt to changing market conditions. However, like all predictive models, their effectiveness depends on the quality of data and features used for training.

:

CHAPTER 6
RESULTS AND TEST CASES

:

6.1 TESTING APPROACH

Test Integration Approaches

There are fundamentally 2 approaches for doing test integration:

1.Bottom-up approach

2.Top-down approach.

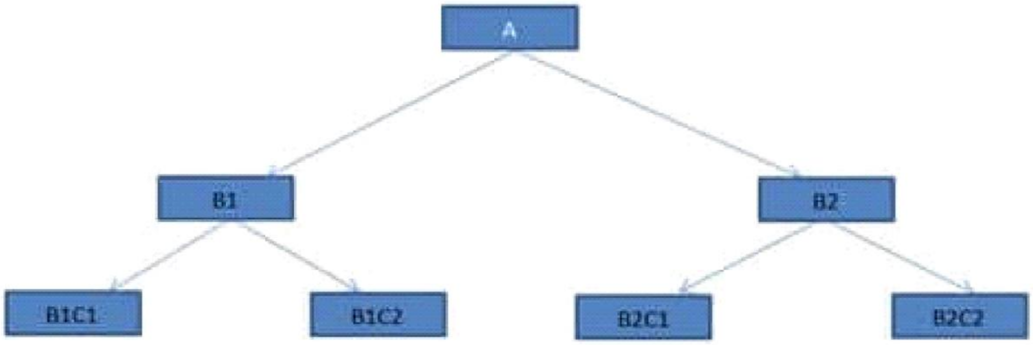


Fig 6.1 Testing Approach

Bottom-up approach

As the name implies, base up testing starts with the smallest or deepest unit of the application and gradually works its way up. Reconciliation testing starts with the smallest module and works its way up to the largest components in the application. This cooperation continues until all of the modules are integrated, at which point the entire application is tested as a single entity.

Hierarchical approach

This approach starts with the topmost module and works its way down to the lower modules gradually. Only the top module is being tested in isolation. The lower modules are then independently integrated after this. The cycle is repeated until each module has been used and tested.

:

Programming testing is an examination directed to furnish partners with data about the nature of the item or administration under test. Programming Testing additionally gives a goal, free perspective on the product to permit the business to appreciate and figure out the dangers at execution of the product. Test procedures incorporate, however are not restricted to, the most common way of executing a program or application with the plan of finding programming bugs.

Programming Testing can likewise be expressed as the method involved with approving and confirming that a product program/application/item:

●Meets the business and specialized prerequisites that directed its plan and Improvement.

●Fills in true to form and can be executed with similar qualities.

6.2 TESTING Techniques

Useful Testing

Utilitarian tests give deliberate exhibitions that capabilities tried are accessible as determined by the business and specialized necessities, framework documentation, and client manuals.

Useful testing is focused on the accompanying things:

●Capabilities: Recognized capabilities should be worked out.

●Yield: Distinguished classes of programming yields should be worked out.

●Frameworks/Strategies: framework ought to work appropriately

Combination Testing

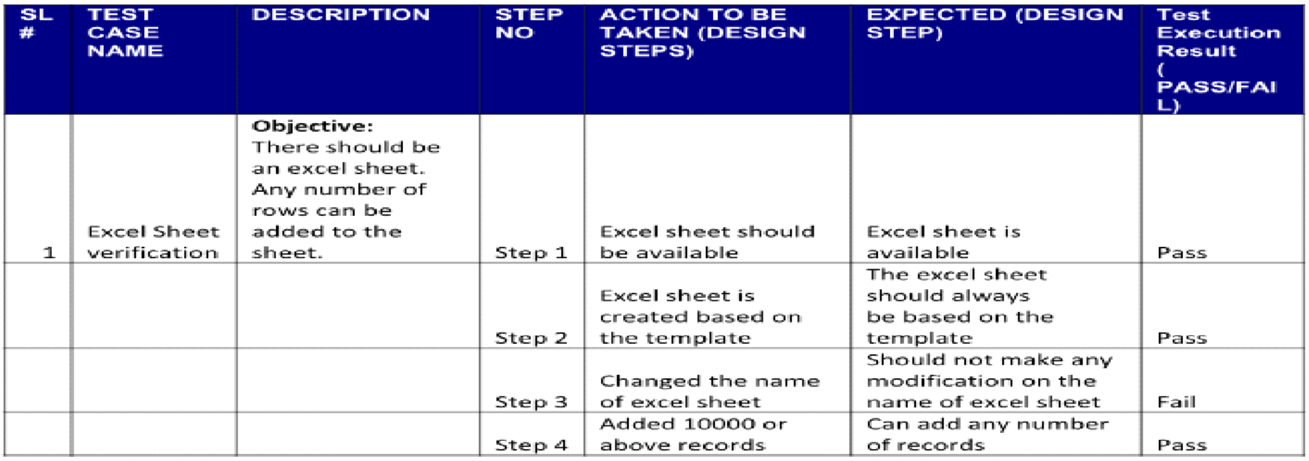
Programming combination testing is the steady coordination testing of at least two

:

incorporated programming parts on a solitary stage to deliver disappointments brought about by interface surrenders.

Experiment for Succeed Sheet Check:

Here in AI we are managing dataset which is in succeed sheet design so assuming that any experiment we really want implies we really want to check succeed document.



SL #	TEST CASE NAME	DESCRIPTION	STEP NO	ACTION TO BE TAKEN (DESIGN STEPS)	EXPECTED (DESIGN STEP)	Test Execution Result (PASS/FAIL)
1	Excel Sheet verification	**Objective:** There should be an excel sheet. Any number of rows can be added to the sheet.	Step 1	Excel sheet should be available	Excel sheet is available	Pass
			Step 2	Excel sheet is created based on the template	The excel sheet should always be based on the template	Pass
			Step 3	Changed the name of excel sheet	Should not make any modification on the name of excel sheet	Fail
			Step 4	Added 10000 or above records	Can add any number of records	Pass

6.2 Test Cases

White Box

TEST CASE1:

INPUT: Anaconda Navigator

OUTPUT: Jupyter Notebook and Browser

TET CASE2:

INPUT: PYTHON PACKAGES IMPORT (Pandas, Numpy, Scikit, Matplot, Seaborn)

OUTPUT: CHECKING OF MODULE

TEST CASE3:

INPUT: EXECUTION OF CODE

OUTPUT: GRAPH AND ACCURACY

:

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

:

7.1 CONCLUSION:

The proposed system, leveraging advanced Machine Learning (ML) and diverse data sources, promises enhanced wealth forecasting in stock markets. Through sophisticated algorithms, real-time adaptability, and ethical compliance, this approach aims to significantly improve accuracy and reliability in predicting market trends and wealth outcomes.

7.2 FUTURE SCOPE:

Future enhancement could involve exploring more advanced boosting algorithms such as XGBoost, LightGBM, or CatBoost to further improve prediction accuracy. Additionally, incorporating feature engineering techniques to extract more relevant information from the time series data, and integrating sentiment analysis of news and social media data to capture market sentiment could enhance the model's performance.

Moreover, implementing ensemble methods to combine predictions from multiple models and employing techniques like hyperparameter tuning and cross-validation for model optimization could lead to better results.

7.3 SOURCE CODE:

PYTHON CODE:

```
import os
import pandas as pd
import numpy as np
import math
import datetime as dt

# For Evalution we will use these library

# For PLotting we will use these library

import matplotlib.pyplot as plt
from itertools import cycle
```

:

```
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
# First we will import the necessary Library

import os
import pandas as pd
import numpy as np
import math
import datetime as dt

# For Evalution we will use these library

# For PLotting we will use these library

import matplotlib.pyplot as plt
from itertools import cycle
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
# Load our dataset
# Note it should be in same dir


maindf=pd.read_csv('amazon.csv')

print('Total number of days present in the dataset: ',maindf.shape[0])
print('Total number of fields present in the dataset: ',maindf.shape[1])

maindf.shape

maindf.head()

maindf.tail()

maindf.info()

# Printing the start date and End date of the dataset

sd=maindf.iloc[0][0]
ed=maindf.iloc[-1][0]
```

:

```
print('Starting Date',sd)
print('Ending Date',ed)


maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2014 = maindf.loc[(maindf['Date'] >= '2014-09-17')
                    & (maindf['Date'] < '2014-12-31')]

y_2014.drop(y_2014[['Adj Close','Volume']],axis=1)


monthvise= y_2014.groupby(y_2014['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
             'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise


fig = go.Figure()

fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Open'],
    name='amazon Open Price',
    marker_color='crimson'
))
fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Close'],
    name='amazon Close Price',
    marker_color='lightsalmon'
))

fig.update_layout(barmode='group', xaxis_tickangle=-45,

                  title='Monthwise comparision between amazon open and close price')
fig.show()
```

:

```
y_2014.groupby(y_2014['Date'].dt.strftime('%B'))['Low'].min()
monthvise_high = y_2014.groupby(maindf['Date'].dt.strftime('%B'))['High'].max()
monthvise_high = monthvise_high.reindex(new_order, axis=0)

monthvise_low = y_2014.groupby(y_2014['Date'].dt.strftime('%B'))['Low'].min()
monthvise_low = monthvise_low.reindex(new_order, axis=0)

fig = go.Figure()
fig.add_trace(go.Bar(
    x=monthvise_high.index,
    y=monthvise_high,
    name='amazon high Price',
    marker_color='rgb(0, 153, 204)'
))
fig.add_trace(go.Bar(
    x=monthvise_low.index,
    y=monthvise_low,
    name='amazon low Price',
    marker_color='rgb(255, 128, 0)'
))

fig.update_layout(barmode='group',
                  title=' Monthwise High and Low amazon price')
fig.show()


names = cycle(['amazon Open Price','amazon Close Price','amazon High Price','amazon
          Low Price'])

fig = px.line(y_2014, x=y_2014.Date, y=[y_2014['Open'], y_2014['Close'],
                                          y_2014['High'], y_2014['Low']],
             labels={'Date': 'Date','value':'amazon value'})
fig.update_layout(title_text='amazon analysis chart', font_size=15,
          font_color='black',legend_title_text='amazon Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

:

```
maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2015 = maindf.loc[(maindf['Date'] >= '2015-01-01')
                     & (maindf['Date'] < '2016-01-01')]

y_2015.drop(y_2015[['Adj Close','Volume']],axis=1)
monthvise= y_2015.groupby(y_2015['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
             'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise


fig = go.Figure()

fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Open'],
    name='Stock Open Price',
    marker_color='crimson'
))
fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Close'],
    name='Stock Close Price',
    marker_color='lightsalmon'
))

fig.update_layout(barmode='group', xaxis_tickangle=-45,
                  title='Monthwise comparision between Stock open and close price')
fig.show()


y_2015.groupby(y_2015['Date'].dt.strftime('%B'))['Low'].min()
monthvise_high = y_2015.groupby(maindf['Date'].dt.strftime('%B'))['High'].max()
monthvise_high = monthvise_high.reindex(new_order, axis=0)

monthvise_low = y_2015.groupby(y_2015['Date'].dt.strftime('%B'))['Low'].min()
monthvise_low = monthvise_low.reindex(new_order, axis=0)

fig = go.Figure()
```

:

```
fig.add_trace(go.Bar(
    x=monthvise_high.index,
    y=monthvise_high,
    name='Stock high Price',
    marker_color='rgb(0, 153, 204)'
))
fig.add_trace(go.Bar(
    x=monthvise_low.index,
    y=monthvise_low,
    name='Stock low Price',
    marker_color='rgb(255, 128, 0)'
))

fig.update_layout(barmode='group',
                  title=' Monthwise High and Low stock price')
fig.show()
```

```
names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low
          Price'])

fig = px.line(y_2015, x=y_2015.Date, y=[y_2015['Open'], y_2015['Close'],
                                          y_2015['High'], y_2015['Low']],
             labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15,
          font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

```
maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2016 = maindf.loc[(maindf['Date'] >= '2016-01-01')
                     & (maindf['Date'] < '2017-01-01')]
y_2016.drop(y_2016[['Adj Close','Volume']],axis=1)
```

:

```
monthvise= y_2016.groupby(y_2016['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
             'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise

fig = go.Figure()

fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Open'],
    name='Stock Open Price',
    marker_color='crimson'
))
fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Close'],
    name='Stock Close Price',
    marker_color='lightsalmon'
))

fig.update_layout(barmode='group', xaxis_tickangle=-45,
                  title='Monthwise comparision between Stock open and close price')
fig.show()


y_2016.groupby(y_2016['Date'].dt.strftime('%B'))['Low'].min()
monthvise_high = y_2016.groupby(maindf['Date'].dt.strftime('%B'))['High'].max()
monthvise_high = monthvise_high.reindex(new_order, axis=0)

monthvise_low = y_2016.groupby(y_2016['Date'].dt.strftime('%B'))['Low'].min()
monthvise_low = monthvise_low.reindex(new_order, axis=0)

fig = go.Figure()
fig.add_trace(go.Bar(
    x=monthvise_high.index,
    y=monthvise_high,
    name='Stock high Price',
```

:

```
   marker_color='rgb(0, 153, 204)'
))
fig.add_trace(go.Bar(
   x=monthvise_low.index,
   y=monthvise_low,
   name='Stock low Price',
   marker_color='rgb(255, 128, 0)'
))

fig.update_layout(barmode='group',
              title=' Monthwise High and Low stock price')
fig.show()

names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low
        Price'])

fig = px.line(y_2016, x=y_2016.Date, y=[y_2016['Open'], y_2016['Close'],
                                y_2016['High'], y_2016['Low']],
          labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15,
        font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()

maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2017 = maindf.loc[(maindf['Date'] >= '2017-01-01')
                & (maindf['Date'] < '2018-01-01')]

y_2017.drop(y_2017[['Adj Close','Volume']],axis=1)


monthvise= y_2017.groupby(y_2017['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
          'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise
fig = go.Figure()
```

:

```
    fig.add_trace(go.Bar(
        x=monthvise.index,
        y=monthvise['Open'],
        name='Stock Open Price',
        marker_color='crimson'
    ))
    fig.add_trace(go.Bar(
        x=monthvise.index,
        y=monthvise['Close'],
        name='Stock Close Price',
        marker_color='lightsalmon'
    ))

    fig.update_layout(barmode='group', xaxis_tickangle=-45,
                      title='Monthwise comparision between Stock open and close price')
    fig.show()


    y_2017.groupby(y_2017['Date'].dt.strftime('%B'))['Low'].min()
    monthvise_high = y_2017.groupby(maindf['Date'].dt.strftime('%B'))['High'].max()
    monthvise_high = monthvise_high.reindex(new_order, axis=0)

    monthvise_low = y_2017.groupby(y_2017['Date'].dt.strftime('%B'))['Low'].min()
    monthvise_low = monthvise_low.reindex(new_order, axis=0)

    fig = go.Figure()
    fig.add_trace(go.Bar(
        x=monthvise_high.index,
        y=monthvise_high,
name='Stock high Price',
        marker_color='rgb(0, 153, 204)'
    ))
    fig.add_trace(go.Bar(
        x=monthvise_low.index,
        y=monthvise_low,
        name='Stock low Price',
        marker_color='rgb(255, 128, 0)'
    ))

    fig.update_layout(barmode='group',
```

:

```
                title=' Monthwise High and Low stock price')
fig.show()



names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low
        Price'])

fig = px.line(y_2017, x=y_2017.Date, y=[y_2017['Open'], y_2017['Close'],
                                          y_2017['High'], y_2017['Low']],
          labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15,
        font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()


maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2018 = maindf.loc[(maindf['Date'] >= '2018-01-01')
                 & (maindf['Date'] < '2019-01-01')]

y_2018.drop(y_2018[['Adj Close','Volume']],axis=1)


monthvise= y_2018.groupby(y_2018['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
          'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise



fig = go.Figure()
fig.add_trace(go.Bar(
   x=monthvise.index,
   y=monthvise['Open'],
   name='Stock Open Price',
```

:

```
    marker_color='crimson'
))
fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Close'],
    name='Stock Close Price',
    marker_color='lightsalmon'
))

fig.update_layout(barmode='group', xaxis_tickangle=-45,
                  title='Monthwise comparision between Stock open and close price')
fig.show()


y_2018.groupby(y_2018['Date'].dt.strftime('%B'))['Low'].min()
monthvise_high = y_2018.groupby(maindf['Date'].dt.strftime('%B'))['High'].max()
monthvise_high = monthvise_high.reindex(new_order, axis=0)

monthvise_low = y_2018.groupby(y_2018['Date'].dt.strftime('%B'))['Low'].min()
monthvise_low = monthvise_low.reindex(new_order, axis=0)

fig = go.Figure()
fig.add_trace(go.Bar(
    x=monthvise_high.index,
    y=monthvise_high,
    name='Stock high Price',
    marker_color='rgb(0, 153, 204)'
))
fig.add_trace(go.Bar(
    x=monthvise_low.index,
    y=monthvise_low,
    name='Stock low Price',
    marker_color='rgb(255, 128, 0)'
))

fig.update_layout(barmode='group',
                  title=' Monthwise High and Low stock price')
fig.show()
```

:

```
names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low
        Price'])

fig = px.line(y_2018, x=y_2018.Date, y=[y_2018['Open'], y_2018['Close'],
                                y_2018['High'], y_2018['Low']],
          labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15,
        font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()



maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2019 = maindf.loc[(maindf['Date'] >= '2019-01-01')
                & (maindf['Date'] < '2020-01-01')]

y_2019.drop(y_2019[['Adj Close','Volume']],axis=1)


monthvise= y_2019.groupby(y_2019['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
          'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise


fig = go.Figure()

fig.add_trace(go.Bar(
   x=monthvise.index,
   y=monthvise['Open'],
   name='Stock Open Price',
   marker_color='crimson'
))
fig.add_trace(go.Bar(
   x=monthvise.index,
   y=monthvise['Close'],
```

:

```
    name='Stock Close Price',
    marker_color='lightsalmon'
))

fig.update_layout(barmode='group', xaxis_tickangle=-45,
                  title='Monthwise comparision between Stock open and close price')
fig.show()


y_2019.groupby(y_2019['Date'].dt.strftime('%B'))['Low'].min()
monthvise_high = y_2019.groupby(maindf['Date'].dt.strftime('%B'))['High'].max()
monthvise_high = monthvise_high.reindex(new_order, axis=0)

monthvise_low = y_2019.groupby(y_2019['Date'].dt.strftime('%B'))['Low'].min()
monthvise_low = monthvise_low.reindex(new_order, axis=0)

fig = go.Figure()
fig.add_trace(go.Bar(
    x=monthvise_high.index,
    y=monthvise_high,
    name='Stock high Price',
    marker_color='rgb(0, 153, 204)'
))
fig.add_trace(go.Bar(
    x=monthvise_low.index,
    y=monthvise_low,
    name='Stock low Price',
    marker_color='rgb(255, 128, 0)'
))

fig.update_layout(barmode='group',
                  title=' Monthwise High and Low stock price')
fig.show()



names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low
          Price'])

fig = px.line(y_2019, x=y_2019.Date, y=[y_2019['Open'], y_2019['Close'],
                                          y_2019['High'], y_2019['Low']],
```

:

```
            labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15,
         font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()



maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2020 = maindf.loc[(maindf['Date'] >= '2020-01-01')
                    & (maindf['Date'] < '2021-01-01')]

y_2020.drop(y_2020[['Adj Close','Volume']],axis=1)



monthvise= y_2020.groupby(y_2020['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
            'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise

fig = go.Figure()

fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Open'],
    name='Stock Open Price',
    marker_color='crimson'
))
fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Close'],
    name='Stock Close Price',
    marker_color='lightsalmon'
))
```

:

```
fig.update_layout(barmode='group', xaxis_tickangle=-45,
              title='Monthwise comparision between Stock open and close price')
fig.show()



y_2020.groupby(y_2020['Date'].dt.strftime('%B'))['Low'].min()
monthvise_high = y_2020.groupby(maindf['Date'].dt.strftime('%B'))['High'].max()
monthvise_high = monthvise_high.reindex(new_order, axis=0)

monthvise_low = y_2020.groupby(y_2020['Date'].dt.strftime('%B'))['Low'].min()
monthvise_low = monthvise_low.reindex(new_order, axis=0)

fig = go.Figure()
fig.add_trace(go.Bar(
   x=monthvise_high.index,
   y=monthvise_high,
   name='Stock high Price',
   marker_color='rgb(0, 153, 204)'
))
fig.add_trace(go.Bar(
   x=monthvise_low.index,
   y=monthvise_low,
   name='Stock low Price',
   marker_color='rgb(255, 128, 0)'
))

fig.update_layout(barmode='group',
              title=' Monthwise High and Low stock price')
fig.show()


names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low
        Price'])

fig = px.line(y_2020, x=y_2020.Date, y=[y_2020['Open'], y_2020['Close'],
                                 y_2020['High'], y_2020['Low']],
          labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15,
        font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
```

:

```
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()


maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2019 = maindf.loc[(maindf['Date'] >= '2019-01-01')
                    & (maindf['Date'] < '2020-01-01')]

y_2019.drop(y_2019[['Adj Close','Volume']],axis=1)


monthvise= y_2019.groupby(y_2019['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
             'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise

fig = go.Figure()

fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Open'],
    name='Stock Open Price',
    marker_color='crimson'
))
fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Close'],
    name='Stock Close Price',
    marker_color='lightsalmon'
))

fig.update_layout(barmode='group', xaxis_tickangle=-45,
                  title='Monthwise comparision between Stock open and close price')
fig.show()


y_2019.groupby(y_2019['Date'].dt.strftime('%B'))['Low'].min()
```

:

```
monthvise_high = y_2019.groupby(maindf['Date'].dt.strftime('%B'))['High'].max()
monthvise_high = monthvise_high.reindex(new_order, axis=0)

monthvise_low = y_2019.groupby(y_2019['Date'].dt.strftime('%B'))['Low'].min()
monthvise_low = monthvise_low.reindex(new_order, axis=0)

fig = go.Figure()
fig.add_trace(go.Bar(
    x=monthvise_high.index,
    y=monthvise_high,
    name='Stock high Price',
    marker_color='rgb(0, 153, 204)'
))
fig.add_trace(go.Bar(
    x=monthvise_low.index,
    y=monthvise_low,
    name='Stock low Price',
    marker_color='rgb(255, 128, 0)'
))

fig.update_layout(barmode='group',
                  title=' Monthwise High and Low stock price')
fig.show()



names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low
        Price'])

fig = px.line(y_2019, x=y_2019.Date, y=[y_2019['Open'], y_2019['Close'],
                                          y_2019['High'], y_2019['Low']],
             labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15,
        font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()

y_2020.drop(y_2020[['Adj Close','Volume']],axis=1)
```

:

HTML CODE:

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>Amazon </title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet'
          type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
          type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
          type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
          rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style1.css') }}">

<style>
  /* Hide the video controls */
  video::-webkit-media-controls {
    display: none !important;
  }

  /* Set the video to fill the entire page */
  video {
    position: fixed;
    top: 50%;
    left: 50%;
    min-width: 100%;
    min-height: 100%;
    width: auto;
    height: auto;
    z-index: -100;
    transform: translate(-50%, -50%);
    background-size: cover;
  }
  h1.{
     color: white;
  }
```

:

```
  p.{
     color: white;
  }
</style>
</head>

<body>

  <div class="slide">
    <!-- Add the video element -->
    <video autoplay loop muted>
      <source
          src="https://player.vimeo.com/external/466550941.sd.mp4?s=db79f6a6272dbcb
          6744729c6cecaa8fcfce7e16e&profile_id=164&oauth2_token_id=57447761"
          type="video/mp4">
     <!--<source
          src="https://player.vimeo.com/external/406129183.sd.mp4?s=bed913d36c27c7
          1aa37366dade5dcae3a9c87de0&profile_id=164&oauth2_token_id=57447761"
          type="video/webm">-->
      <p>Your browser does not support the video element.</p>
    </video>
    </div>
    <div class="slide">
    <video autoplay loop muted>
      <source
          src="https://player.vimeo.com/external/371572588.sd.mp4?s=999ddda416fac6a
          fa5b2904568f548e1e812b834&profile_id=164&oauth2_token_id=57447761"
          type="video/mp4">
     <!--<source
          src="https://player.vimeo.com/external/406129183.sd.mp4?s=bed913d36c27c7
          1aa37366dade5dcae3a9c87de0&profile_id=164&oauth2_token_id=57447761"
          type="video/webm">-->
      <p>Your browser does not support the video element.</p>
    </video>
    </div>

  </div>
  <div class="slide">
  <video autoplay loop muted>
    <source
          src="https://player.vimeo.com/external/539039272.sd.mp4?s=1d7cc0affa7a12a3
```

:

```
        f5697538e8de2f95597bfcda&profile_id=164&oauth2_token_id=57447761s"
        type="video/mp4">
     <p>Your browser does not support the video element.</p>
    </video>
    </div>

   <div class="login">
        <h1>Amazon stock price Prediction</h1>


     <form action="{{ url_for('predict')}}"method="post">



      <input type="text" name="Date" placeholder="Date" required="required" />
      <br>
      <br>

           <input type="text" name="Open" placeholder="Open" required="required"
        />
           <br>
           <br>
           <input type="text" name="High" placeholder="High" required="required" />
           <br>
           <br>
           <input type="text" name="Low" placeholder="Low" required="required" />
           <br>
           <br>
           <input type="text" name="Close" placeholder="close" required="required"
        />
           <br>
           <br>

           <input type="text" name="Adj Close" placeholder="Adj Close"
        required="required" />
           <br>
           <br>



       <button type="submit" class="btn btn-primary btn-block btn-
```

:

```
         large">Predict</button>
    </form>

   <br><br><h3 style="color:white;">{{ prediction_text }}<h3>

 </div>


</body>
</html>
```

BACKEND FLASK CODE:

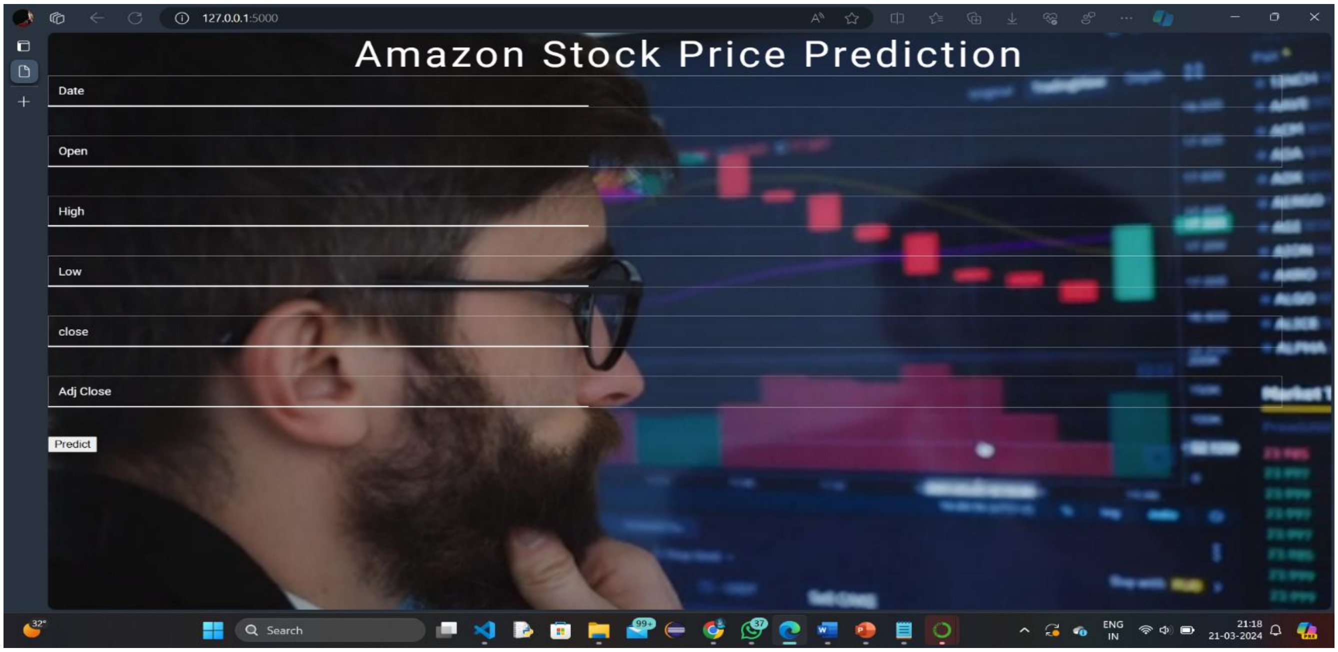
```
import numpy as np
import pandas as pd
from flask import Flask, request, jsonify, render_template
import pickle
import joblib

app = Flask(__name__,template_folder='templates')
model = pickle.load(open('model.pkl', 'rb'))




@app.route('/')
def home():
    return render_template('index1.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    float_features = [float(x) for x in request.form.values()]

    #final_features = [np.array(float_features)]
    final_features_series = pd.Series(float_features)

    #prediction = model.predict(final_features))
```

:

```
    prediction = model.predict(final_features_series.values.reshape(1,-1))



    output=round(prediction[0],1)



    return render_template('index1.html', prediction_text='Value : {}'.format(output))

if __name__ == "__main__":
    app.debug=True
    app.run(host='0.0.0.0',port=5000)
```

:

7.4 SCREENSHOTS



127.0.0.1:5000

Amazon Stock Price Prediction

Date

Open

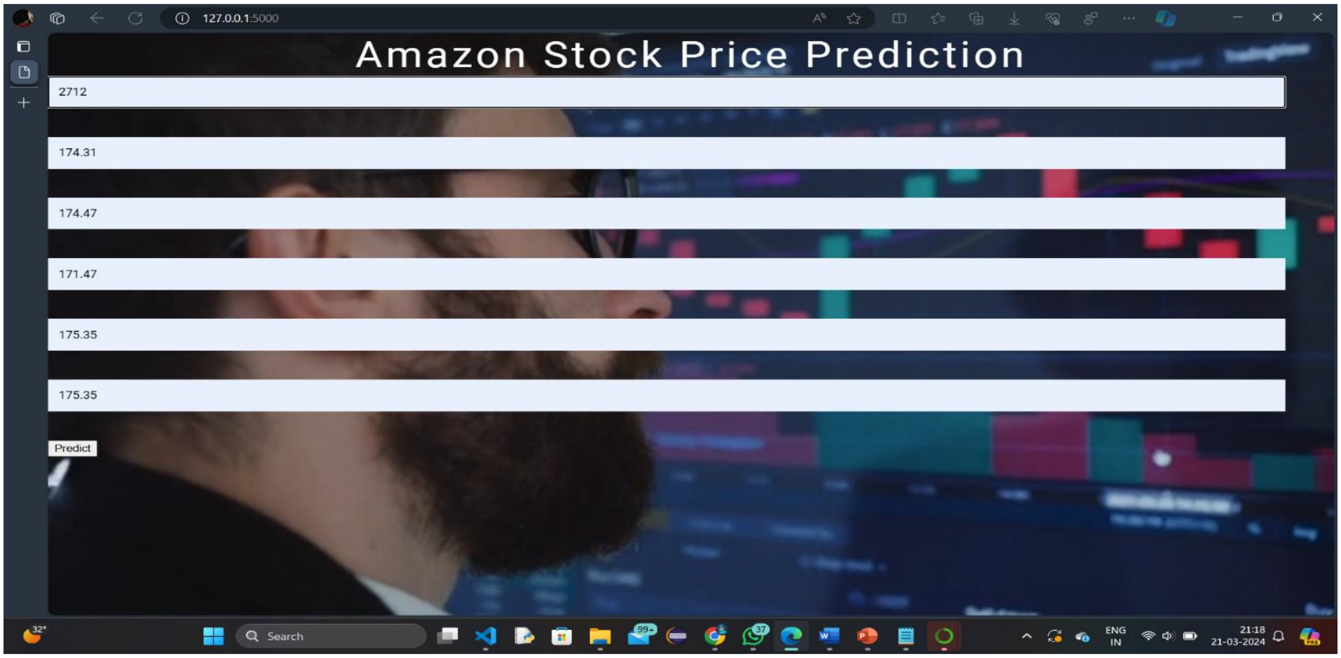
High

Low

close

Adj Close

Predict



127.0.0.1:5000

Amazon Stock Price Prediction

2712

174.31

174.47

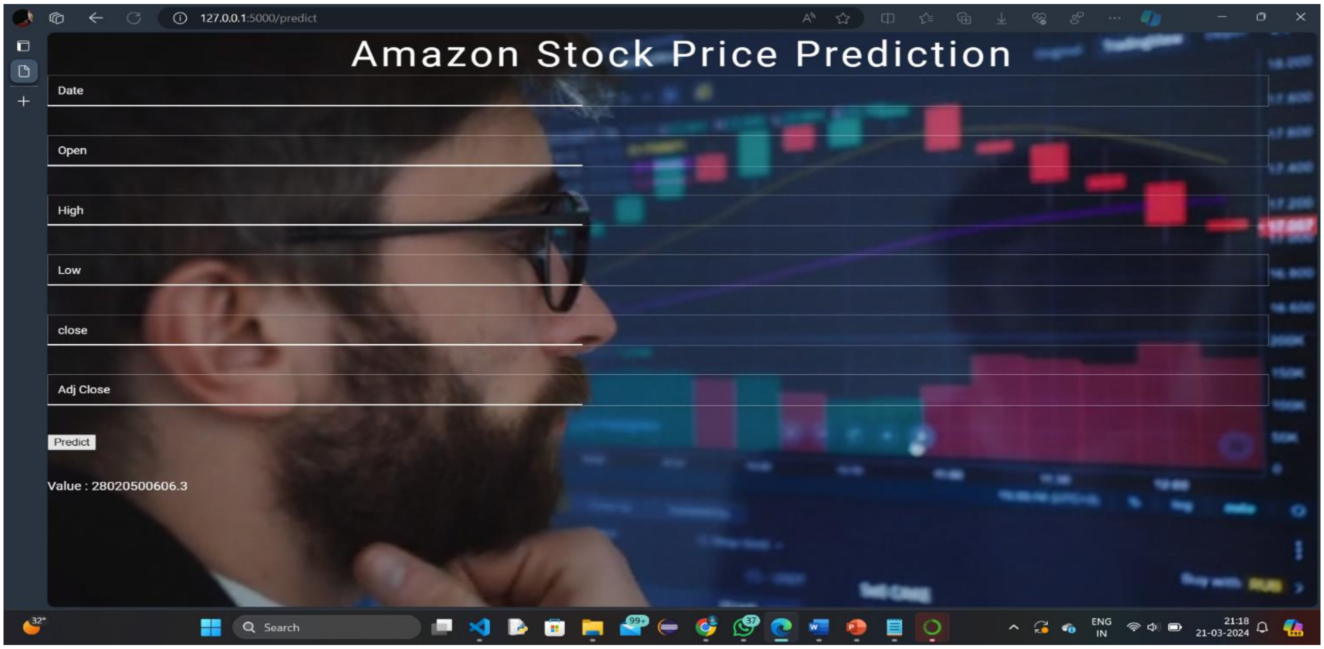
171.47

175.35

175.35

Predict

:



127.0.0.1:5000/predict

Amazon Stock Price Prediction

Date

Open

High

Low

close

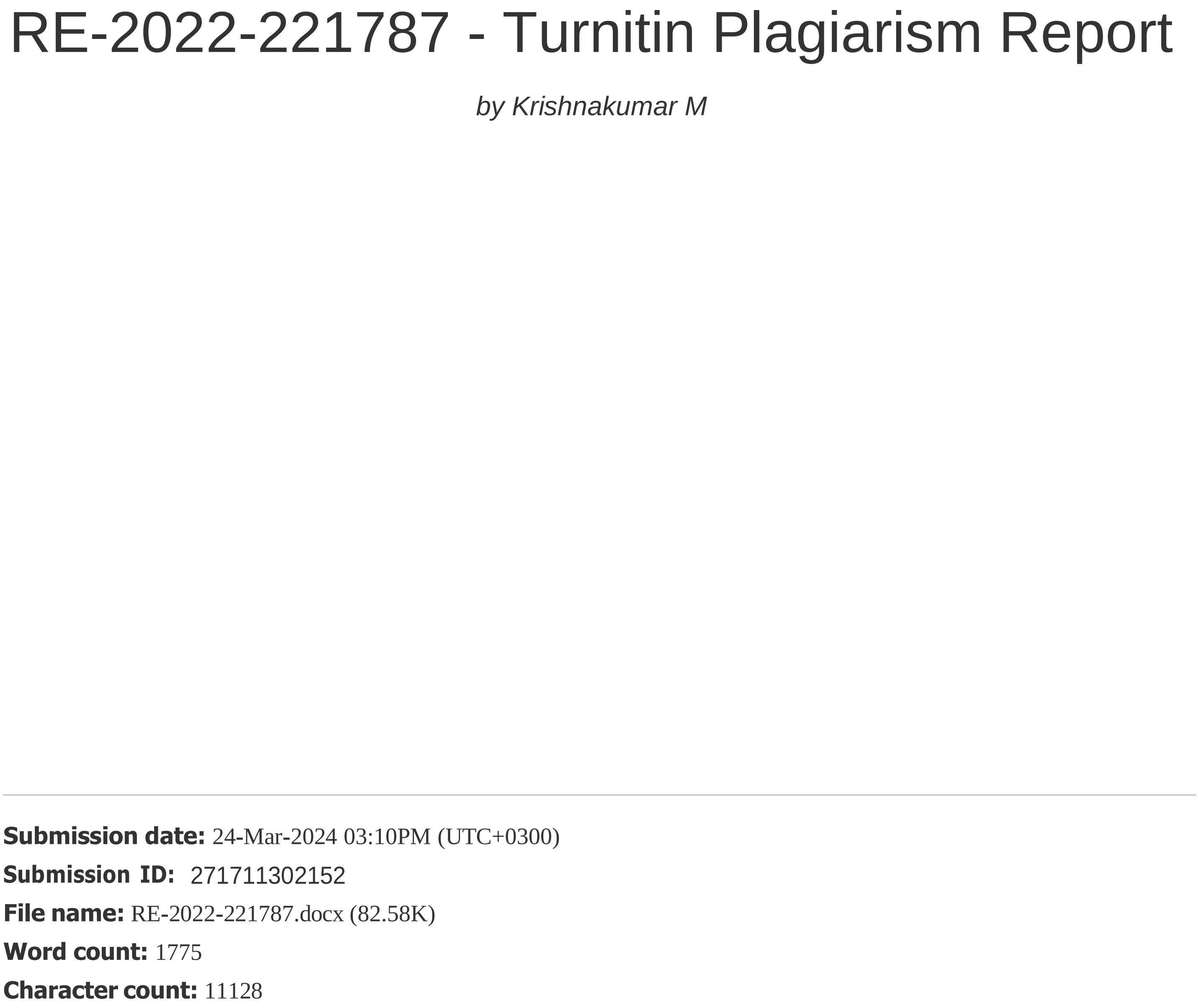
Adj Close

Predict

Value : 28020500606.3

:

7.5 PLAGAIRISM REPORT



RE-2022-221787 - Turnitin Plagiarism Report

by Krishnakumar M

Submission date: 24-Mar-2024 03:10PM (UTC+0300)
Submission ID: 271711302152
File name: RE-2022-221787.docx (82.58K)
Word count: 1775
Character count: 11128

:

TIME SERIES – STOCK PRICE PREDICTION USING BOOSTING ALGORITHM ML MODEL

Dr. T.A.MOHANA PRAKASH
Associate Professor
Department of Computer Science and Engineering
Panimalar Engineering College
Chennai, India
tamohanaprakash@gmail.com

M.KRISHNAKUMAR
UG Scholar
Department of Computer Science and Engineering
Panimalar Engineering College
Chennai, India
mkmkrishna11@gmail.com

S.JAYAKUMAR
UG Scholar
I
Panimalar
jayakumarsubramani2003@gmail.com

M .KUMARESWAAR
UG Scholar
Department of Computer Science and Engineering
Panimalar Engineering College
Chennai,India.
jack007kumareswaar@gmail.com

Abstract:

In exploring the future of wealth forecasting in stock markets, this paper highlights the critical role that cutting-edge Machine Learning (ML) approaches play in providing accurate AI-driven stock predictions. This strategy uses advanced machine learning techniques to support prediction models without naming specific algorithms. Key developments influencing better AI-based stock predictions include improvements in machine learning models, real-time data processing, alternate data sources, and effective risk management techniques. Nevertheless, considering the intrinsic volatility of financial markets, it is still necessary to supplement AI-generated insights with thorough financial analysis and guarantee ethical compliance in AI-driven wealth forecasting.

Deep learning and sentiment analysis are two examples of the advanced machine learning (ML) models that will be necessary for stock market wealth predictions in the future. Real-time data processing, strong risk management, and ML-driven advancements will all increase.

Introduction :

The introduction of advanced Machine Learning (ML) methodologies has revolutionized wealth forecasting in stock markets, presenting a promising avenue for accurate AI-driven predictions. This introduction sets the stage for exploring the transformative potential of ML techniques in predicting stock market trends without specifying particular algorithms. By harnessing the power of ML, this approach aims to revolutionize wealth forecasting, emphasizing the importance of technological advancements, real-time data processing, alternative data sources, and risk management strategies. It also highlights the need for comprehensive financial analysis and ethical considerations in navigating the dynamic and unpredictable landscape of financial market.

Data scientists use many different kinds of machine learning

a high level, the

nsuperv

Classification predictive modeling is the task of approximating a mapping functionfrom input variables(X) to discrete output variables(y). In machine learning and statistics,classification is a supervised learning approach in which the computer program learns fromthe data input given to it and then uses this learning to classify new observations.

Literature Survey:

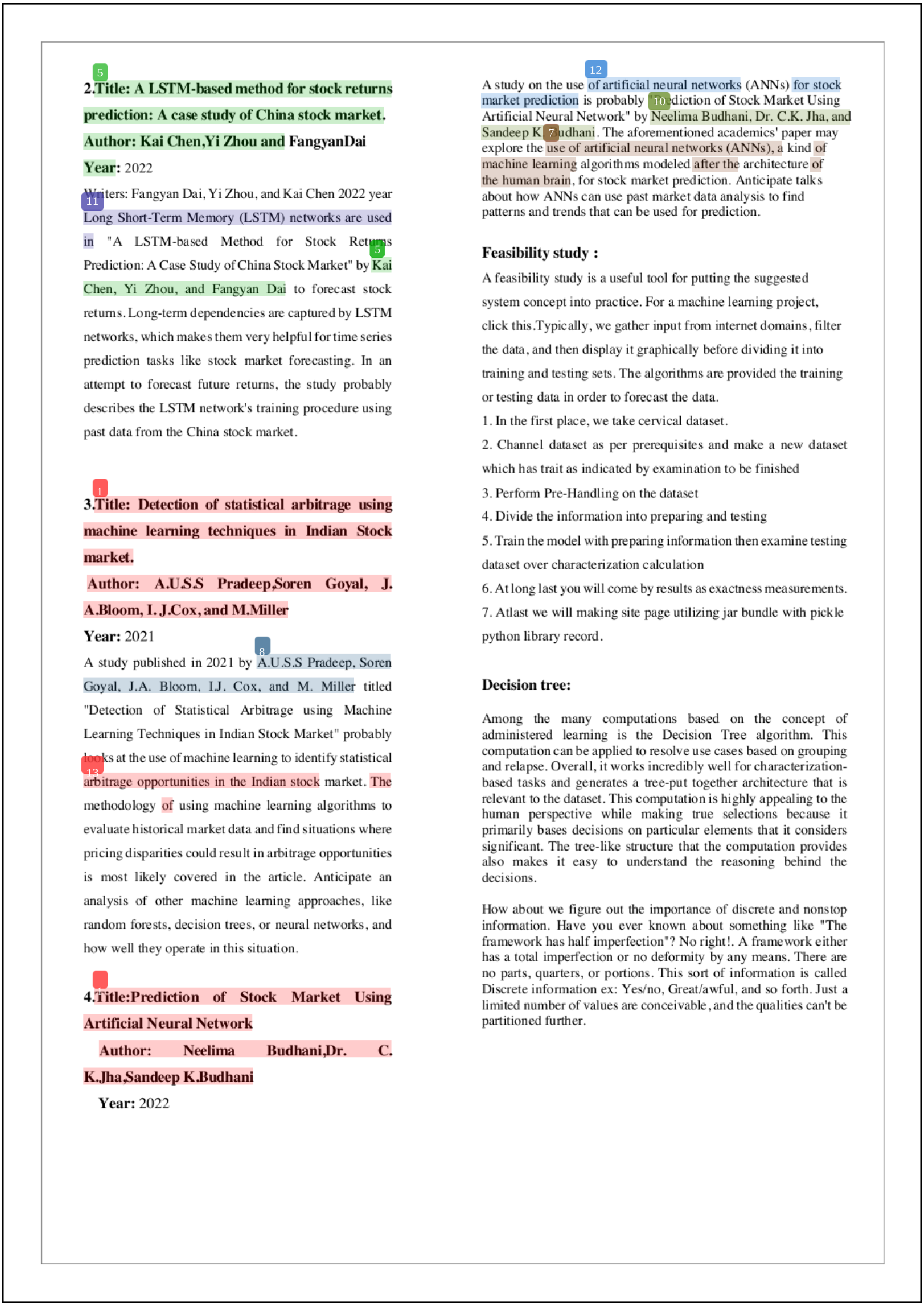
1.Title: Parameters For Stock Market Prediction

Author: Prashant S., Chavan,Prof. , Dr.Shrishail. T.Patil

Year:2018

"Parameters for Stock Market Prediction" offers insights into the microeconomic aspects impacting stock market movements. Basic microeconomic ideas including supply and demand, pricing strategies, and market efficiency are covered. It is probable that the book goes into great detail about analyzing companies, including how to evaluate financial statements, profitability, cost structures, and industry competitive positioning. Anticipate talks on regulatory environments, technology disruptions, industry developments, and stock price analysis. Economic variables that are investigated for their applicability to stock market forecasts include GDP growth rates, inflation, interest rates, and employment statistics.

:



2.Title: A LSTM-based method for stock returns prediction: A case study of China stock market.

Author: Kai Chen,Yi Zhou and FangyanDai

Year: 2022

Writers: Fangyan Dai, Yi Zhou, and Kai Chen 2022 year Long Short-Term Memory (LSTM) networks are used in "A LSTM-based Method for Stock Returns Prediction: A Case Study of China Stock Market" by Kai Chen, Yi Zhou, and Fangyan Dai to forecast stock returns. Long-term dependencies are captured by LSTM networks, which makes them very helpful for time series prediction tasks like stock market forecasting. In an attempt to forecast future returns, the study probably describes the LSTM network's training procedure using past data from the China stock market.

3.Title: Detection of statistical arbitrage using machine learning techniques in Indian Stock market.

Author: A.U.S.S Pradeep,Soren Goyal, J. A.Bloom, I. J.Cox, and M.Miller

Year: 2021

A study published in 2021 by A.U.S.S Pradeep, Soren Goyal, J.A. Bloom, I.J. Cox, and M. Miller titled "Detection of Statistical Arbitrage using Machine Learning Techniques in Indian Stock Market" probably looks at the use of machine learning to identify statistical arbitrage opportunities in the Indian stock market. The methodology of using machine learning algorithms to evaluate historical market data and find situations where pricing disparities could result in arbitrage opportunities is most likely covered in the article. Anticipate an analysis of other machine learning approaches, like random forests, decision trees, or neural networks, and how well they operate in this situation.

4.Title:Prediction of Stock Market Using Artificial Neural Network

Author: Neelima Budhani,Dr. C. K.Jha,Sandeep K.Budhani

Year: 2022

A study on the use of artificial neural networks (ANNs) for stock market prediction is probably Prediction of Stock Market Using Artificial Neural Network" by Neelima Budhani, Dr. C.K. Jha, and Sandeep K Budhani. The aforementioned academics' paper may explore the use of artificial neural networks (ANNs), a kind of machine learning algorithms modeled after the architecture of the human brain, for stock market prediction. Anticipate talks about how ANNs can use past market data analysis to find patterns and trends that can be used for prediction.

Feasibility study :

A feasibility study is a useful tool for putting the suggested system concept into practice. For a machine learning project, click this.Typically, we gather input from internet domains, filter the data, and then display it graphically before dividing it into training and testing sets. The algorithms are provided the training or testing data in order to forecast the data.

1. In the first place, we take cervical dataset.

2. Channel dataset as per prerequisites and make a new dataset which has trait as indicated by examination to be finished

3. Perform Pre-Handling on the dataset

4. Divide the information into preparing and testing

5. Train the model with preparing information then examine testing dataset over characterization calculation

6. At long last you will come by results as exactness measurements.

7. Atlast we will making site page utilizing jar bundle with pickle python library record.

Decision tree:

Among the many computations based on the concept of administered learning is the Decision Tree algorithm. This computation can be applied to resolve use cases based on grouping and relapse. Overall, it works incredibly well for characterization-based tasks and generates a tree-put together architecture that is relevant to the dataset. This computation is highly appealing to the human perspective while making true selections because it primarily bases decisions on particular elements that it considers significant. The tree-like structure that the computation provides also makes it easy to understand the reasoning behind the decisions.

How about we figure out the importance of discrete and nonstop information. Have you ever known about something like "The framework has half imperfection"? No right!. A framework either has a total imperfection or no deformity by any means. There are no parts, quarters, or portions. This sort of information is called Discrete information ex: Yes/no, Great/awful, and so forth. Just a limited number of values are conceivable, and the qualities can't be partitioned further.

:

III. Existing System:

The existing systems for wealth forecasting in stock markets involve statistical, quantitative, technical, and fundamental analysis, complemented by expert opinions and news analysis. However, these methods face challenges in adapting to market complexities, unexpected events, and may struggle to consistently outperform the market due to the efficient market hypothesis. There's a growing need to explore more advanced Machine Learning (ML) techniques and diverse data integration to enhance accuracy and adaptability in predicting market trends.

Disadvantages:

- Limitations in Adaptability
- Complexity in Predictive Accuracy
- Risk of Financial Loss
- Inadequate Integration of Advanced Techniques

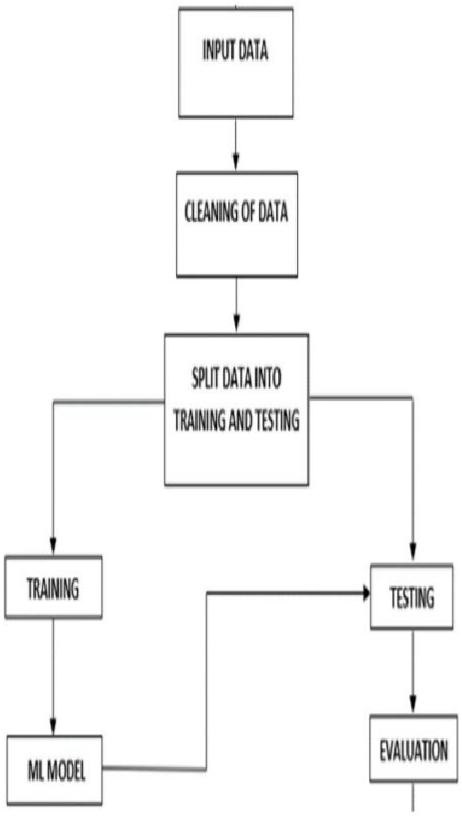
IV. Proposed System:

The proposed system for improved wealth forecasting in stock markets integrates advanced Machine Learning (ML) techniques, alternative data sources, real-time analysis, and ethical compliance. By employing sophisticated ML algorithms, embracing diverse datasets, ensuring adaptability to changing market conditions, and focusing on transparency and ethical considerations, the system aims to enhance accuracy and reliability in predicting market trends and wealth forecasting. The framework offers a robust set of features including routing, request handling, and templating. Flask is known for its modular design, allowing developers to add extensions for specific functionalities as needed. It supports various web development tasks such as URL routing, HTTP request handling, and rendering templates.

Advantages:

- Enhanced Accuracy
- Improved Decision-making
- Enhanced Safety
- Early Detection

Block Diagram:



Software and Hardware Requirements:

Hardware:

- OS – Windows 7, 8 and 10 (32 and 64 bit)
- RAM – 4GB
- Database

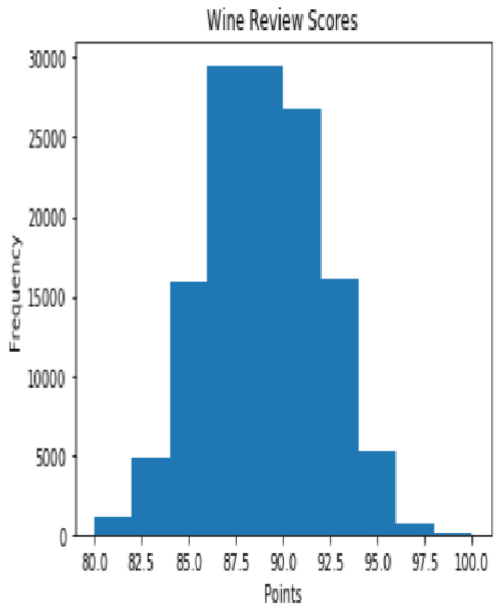
Software:

- Python / Anaconda Navigator
- Packages: numpy, Pandas, matplotlib, Sklearn
- Flask Framework

:

Histogram:

In Matplotlib we can make a Histogram utilizing the hist technique. Assuming that we pass it downright information like the focuses segment from the wine-survey dataset it will naturally ascertain how frequently each class happens.



V.Conclusion:

The proposed system, leveraging advanced Machine Learning (ML) and diverse data sources, promises enhanced wealth forecasting in stock markets. Through sophisticated algorithms, real-time adaptability, and ethical compliance, this approach aims to significantly improve accuracy and reliability in predicting market trends and wealth outcomes.

VI.Future works:

Future enhancement could involve exploring more advanced boosting algorithms such as XGBoost, LightGBM, or CatBoost to further improve prediction accuracy. Additionally, incorporating feature engineering techniques to extract more relevant information from the time series data, and integrating sentiment analysis of news and social media data to capture market sentiment could enhance the model's performance.

Moreover, implementing ensemble methods to combine predictions from multiple models and employing techniques like hyperparameter tuning and cross-validation for model optimization could lead to better results.

Reference:

1] RakhiMahant, TrilokNathPandey, Alok Kumar Jagadev, and SatchidanandaDehuri —Optimized Radial Basis Functional Neural Network for Stock Index Prediction,‖ International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016.

[2] Kai Chen, Yi Zhou and FangyanDai —A LSTM-based method for stock returns prediction: A case study of China stock market,‖ IEEE International Conference on Big Data,2015.

[3] A.U.S.S Pradeep, SorenGoyal, J. A. Bloom, I. J. Cox, and M. Miller, —Detection of statistical arbitrage using machine learning techniques in Indian Stock market,‖ IIT Kanpur, April 15, 2013.

[4] Prashant S. Chavan, Prof. Dr. Shrishail. T. Patil —Parameters for Stock Market Prediction,‖ Prashant S Chavan et al, Int.J.Computer Technology & Applications, Vol 4 (2),337-340.

[5] NeelimaBudhani, Dr. C. K. Jha, Sandeep K. Budhani—Prediction of Stock Market Using Artificial Neural Network,‖ International Conference on Soft Computing Techniques for Engineering and Technology (ICSCTET)- 2014. [6] SharvilKatariya, Saurabh Jain—Stock Price Trend Forecasting using Supervised Learning Methods.

[7] DhirajMundada, GauravChhaparwal, SachinChaudhari, and TruptiBhamare— Stock Value Prediction System,‖ International Journal on Recent and Innovation Trends in Computing and Communication, April 2015

[8] Hochreiter, Sepp, and Jürgen Schmidhuber— Long short-term memory,‖ Neural computation 9.7740291\78165

32..8 (1997): 1735-1780

:

RE-2022-221787-plag-report

ORIGINALITY REPORT

14 %	9%	10%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Gitam University Student Paper	3 %
2	ijircce.com Internet Source	2 %
3	Submitted to Higher Education CommissionPakistan Student Paper	1 %
4	www.ijritcc.org Internet Source	1 %
5	mdanderson.influuent.utsystem.edu Internet Source	1 %
6	www.pantechproed.com Internet Source	1 %
7	Submitted to University of Surrey Student Paper	1 %
8	www.ijraset.com Internet Source	1 %
9	ijritcc.org Internet Source	1 %

:

10	www.ijert.org Internet Source	1%
11	David M. Q. Nelson, Adriano C. M. Pereira, Renato A. de Oliveira. "Stock market's price movement prediction with LSTM neural networks", 2017 International Joint Conference on Neural Networks (IJCNN), 2017 Publication	1%
12	Neelima Budhani, C. K. Jha, Sandeep K. Budhani. "Prediction of stock market using artificial neural network", 2014 International Conference of Soft Computing Techniques forEngineering and Technology (ICSCTET), 2014 Publication	1%
13	cse.iitk.ac.in Internet Source	1%

:

CHAPTER 8
REFERENCES

:

References:

[1] RakhiMahant, TrilokNathPandey, Alok Kumar Jagadev, and SatchidanandaDehuri ―Optimized Radial Basis Functional Neural Network for Stock Index Prediction,‖ International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016.

[2] Kai Chen, Yi Zhou and FangyanDai ―A LSTM-based method for stock returns prediction: A case study of China stock market,‖ IEEE International Conference on Big Data,2015.

[3] A.U.S.S Pradeep, SorenGoyal, J. A. Bloom, I. J. Cox, and M. Miller, ―Detection of statistical arbitrage using machine learning techniques in Indian Stock market,‖ IIT Kanpur, April 15, 2013.

[4] Prashant S. Chavan, Prof. Dr. Shrishail. T. Patil ―Parameters for Stock Market Prediction,‖ Prashant S Chavan et al, Int.J.Computer Technology & Applications, Vol 4 (2),337-340.

[5] NeelimaBudhani, Dr. C. K. Jha, Sandeep K. Budhani―Prediction of Stock Market Using Artificial Neural Network,‖ International Conference on Soft Computing Techniques for Engineering and Technology (ICSCTET)- 2014. [6] SharvilKatariya, Saurabh Jain―Stock Price Trend Forecasting using Supervised Learning Methods.

[7] DhirajMundada, GauravChhaparwal, SachinChaudhari, and TruptiBhamare― Stock Value Prediction System,‖ International Journal on Recent and Innovation Trends in Computing and Communication, April 2015

[8] Hochreiter, Sepp, and Jürgen Schmidhuber― Long short-term memory,‖ Neural computation 9.7740291\78165
32..8 (1997): 1735-1780

: