## CRYPTO CURRENCY MARKET PRICE PREDICTION USING DATA SCIENCE PROCESS

#### A PROJECT REPORT

***Submitted by***

|  |  |
| --- | --- |
| **SASHI KIRAN S** | **211418104234** |
| **SATISH R** | **211418104236** |
| **SENTHIL RAJ S** | **211418104240** |

***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)**

**MAY 2022**

## PANIMALAR ENGINEERING COLLEGE

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

### BONAFIDE CERTIFICATE

Certified that this project report **“CRYPTO CURRENCY MARKET PRICE PREDICTION USING DATA SCIENCE PROCESS”** is the bonafide work of **SASHI KIRAN S (211418104234), SATISH R (211418104236), SENTHIL RAJ S**

**(211418104240)** who carried out the project work under my supervision.

#### SIGNATURE SIGNATURE

**Dr. S. MURUGAVALLI, M.E., Ph.D., S.A.K. JAINULABUDEEN, M. Tech., PROFESSOR SUPERVISOR**

#### HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE, NAZARATHPETTAI, NAZARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above-mentioned students were examined in End Semester project viva-voce held on .

#### INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

Our sincere, humble and heartfelt thanks and praises to our honorable Founder and Chairman Late **Dr. JEPPIAAR, M.A., B.L., Ph.D.,** for his sincere Endeavor in educating us in his reputed institution.

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **TMT.C. VIJAYARAJESWARI**, **Dr. C. SAKTHI KUMAR, M.E., Ph.D.,** and **Dr. SARANYA SREE SAKTHI**

**KUMAR, B.E., MBA, Ph.D.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.,** for his timely concern and encouragement provided to us throughout the course.

We thank the Head of the CSE Department, **Dr. S. MURUGAVALLI, M.E., Ph.D.,**

for the support extended throughout the project.

We would like to thank my **Project Coordinator Dr. N. PUGAZHENDI, M.E., Ph.D.,** and **Supervisor Mr. S.A.K. JAINULABUDEEN, M. Tech.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

## SASHI KIRAN S SATISH R SENTHIL RAJ S

**DECLARATION BY THE STUDENT**

#### We SASHI KIRAN S [211418104234], SATISH R [211418104236], SENTHIL

**RAJ S [211418104240]** hereby declare that this project report titled “**CRYPTO CURRENCY MARKET PRICE PREDICTION USING DATA SCIENCE**

**PROCESS**” under the guidance of **Mr. S.A.K. JAINULABUDEEN** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

**SASHI KIRAN S SATISH R SENTHIL RAJ S**

**ABSTRACT**

A cryptocurrency is a tradable digital asset or digital money, built on blockchain technology that exists online. Nowadays crypto currency is in large scale and there's a sudden rise or decrease in their share and it's difficult to predict the worth of the crypto currency. during this project a machine learning model is made to predict the value of crypto currency. the appliance of data science process is applied for getting the higher model for predicting the result. Variable identification and data understanding is that the main process in building the successful model. Different machine learning algorithms are applied on the pre-processed data and also the accuracy are compared to work out which algorithm performed better other performance metrics like precision, recall, score also are taken in consideration for evaluating the model. The machine learning model is to predict the crypto currency outcome.

### TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | v |
|  | **LST OF FIGURES** | viii |
|  | **LIST OF SYMBOLS** | ix |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 Problem Definition | 2 |
|  | 1.2 Scope of the Project | 2 |
| **2.** | **LITERATURE SURVEY** | 3 |
| **3.** | **SYSTEM ANALYSIS** | 9 |
|  | 3.1 Existing System | 10 |
|  | 3.2 Proposed system | 10 |
|  | 3.3 Feasibility Study | 11 |
|  | 3.4 Software Requirements | 12 |
|  | 3.5 Hardware Requirements | 12 |
| **4.** | **SYSTEM DESIGN** | 13 |
|  | 4.1 ER diagram | 14 |
|  | 4.2 UML Diagram | 15 |
|  | 4.3 Work Flow Diagram | 20 |
| **5.** | **SYSTEM ARCHITECTURE** | 21 |
|  | 5.1 Architecture Diagram | 22 |
|  | 5.2 Module Design Specification | 23 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | 5.3 Tools & Packages | 40 |
| **6.** | **SYSTEM TESTING** | 42 |
|  | 6.1 Unit Testing | 43 |
|  | 6.2 Integration Testing | 43 |
|  | 6.3 Test Case & Report | 44 |
| **7.** | **CONCLUSION** | 46 |
|  | 7.1 Conclusion | 47 |
|  | 7.2 Future Enhancements | 47 |
|  | **APPENDICES** | 48 |
|  | A.1 Coding | 48 |
|  | A.2 Sample Screenshots | 72 |
|  | **PLAGIARISM REPORT** | 76 |
|  | **REFERENCES** | 77 |

#### LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIG. NO** | **TITLE** | **PAGE NO** |
| **4.1** | **ER – Diagram** | **14** |
| **4.2.1** | **Use case Diagram** | **15** |
| **4.2.2** | **Class Diagram** | **16** |
| **4.2.3** | **Activity Diagram** | **17** |
| **4.2.4** | **Sequence Diagram** | **18** |
| **4.2.5** | **Collaboration Diagram** | **19** |
| **4.3** | **Workflow Diagram** | **20** |
| **5.1** | **System Architecture** | **22** |

**LIST OF SYMBOLS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NOTATION**  **NAME** | **NOTATION** | **DESCRIPTION** |
| 1. | Class | *Class Name*  *-attribute*  *-attribute*  *+ public +operation*  *-private +operation*  *# protected +operation* | Represents a  collection of similar entities grouped together. |
| 2. | Association | Class A NAME Class B Class A Class B | Associations represents static relationships between classes. Roles represents the way the two classes see each other. |
| 3. | Actor |  | It aggregates several classes into a single classes. |

|  |  |  |  |
| --- | --- | --- | --- |
| 4. | Aggregation | Class A Class A    Class B Class B | Interaction between the system and external  environment |
| 5. | Relation(uses) | uses | Used for additional process communication. |
| 6. | Relation (extends) | extends | Extends relationship is used when one use case is similar to another use case but does a  bit more. |
| 7. | Communication |  | Communication between various use cases. |
| 8. | State | State | State of the process. |
| 9. | Initial State |  | Initial state of the object |



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10. | Final state |  | | | | | Final state of the object |
| 11. | Control flow |  | | | | | Represents various control flow between the states. |
| 12. | Decision box |  | | | | | Represents decision making process from a constraint |
| 13. | Use case | Uses case | | | | | Interaction between the system and external environment. |
| 14. | Component |  | | | | | Represents physical modules which is a  collection of components. |
|  | |  | |  |
|  |  | |  |
|  |  |  |  |
|  |  | |  |
|  |  |  | |
|  | | | | |



|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 15. | Node |  | Represents physical modules which are a  collection of components |
| 16. | Data Process/State |  | A circle in DFD represents a state or process which has been triggered due to some event or action. |
| 17. | External entity |  | Represents external entities such as keyboard, sensors etc. |
| 18. | Transition |  | Represents communication that occurs between  processes. |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 19. | Object Lifeline |  | Represents the vertical dimensions that the object communications. |
| 20. | Message | Message | Represents the message exchanged. |

# CHAPTER 1 INTRODUCTION

### INTRODUCTION

* 1. **Problem Definition**

Crypto currency is a digital currency where the coin ownership records are stored in a very ledger existing in an exceedingly sort of a computerized database using strong cryptography to secure transaction records and to regulate the creation of additional coins, and to verify the transfer of coin ownership. Nowadays crypto currency are in large scale and there's a sudden rise or decrease in their share and it's difficult to predict the value of the crypto currency. during this project a machine learning model is constructed to predict the worth of crypto currency. the appliance of information science process is applied for getting the higher model for predicting the result. Variable identification and data understanding is that the main process in building the successful model.

### Scope of the Project

Cryptocurrency is an internet-based medium of exchange within the type of digital assets which uses cryptographic functions to conduct financial transactions. Cryptocurrencies leverage blockchain technology to achieve decentralization, transparency, and immutability. the most scope of the project is to finding the accuracy and getting result from the flask framework deployment.

# CHAPTER 2 LITERATURE SURVEY

### LITERATURE SURVEY

#### General

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period.

Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them.

Loan default trends have been long studied from a socio-economic stand point. Most economics surveys believe in empirical modeling of these complex systems in order to be able to predict the loan default rate for a particular individual. The use of machine learning for such tasks is a trend which it is observing now. Some of the survey’s to understand the past and present perspective of loan approval or not.

#### Review of Literature Survey

**Title** : Identifying and Analyzing Cryptocurrency Manipulations in Social Media [1]

**Author:** Mehrnoosh Mirtaheri, Fred Morstatter

#### Year : 2021

Interest surrounding cryptocurrencies, digital or virtual currencies that are used as a medium for financial transactions, has grown tremendously in the recent years. The anonymity surrounding these currencies makes investors particularly susceptible to fraudity—such as “pump and dump” scams— where the goal is to artificially inflate the perceived worth of a currency, luring victims into investing before the fraudsters can sell their holdings. Because of the speed and relative anonymity offered by social platforms such as Twitter and Telegram, social media has become a preferred platform for scammers who wish to spread false hype about the cryptocurrency they are trying to pump. This work proposes and evaluates a computational approach that can automatically identify pump and dump scams as they unfold by combining information across social media platforms. It also develop a multi-modal approach for predicting whether a particular pump attempt will succeed or not. Finally, It analyze the prevalence of bots in cryptocurrency related tweets, and observe a significant increase in bot activity during the pump attempts.

**Title :** A Research On Bitcoin Price Prediction Using Machine Learning Algorithms [2]

**Author:** Lekkala Sreekanth Reddy, Dr.P. Sriramya

#### Year : 2020

This paper was proposed to predict the Bitcoin price accurately taking into consideration various parameters that affect the Bitcoin value. By gathering information from different reference papers and applying in real time, advantages and

disadvantages of bitcoin price prediction was found . Each and every paper has its own

set of methodologies of bitcoin price prediction. Many papers has accurate price but some other don’t, but the time complexity is higher in those predictions, so to reduce the time complexity this paper use an algorithm linked to artificial intelligence named LASSO(least absolute shrinkage selection operator. The other papers used different algorithms like SVM(support vector machine),coinmarkupcap, Quandl, GLM, CNN(Convolutional Neural Networks)and RNN(Recurrent neural networks) etc.. which do not have a great time management, but in LASSO finding of the results from a larger database is quick and fast..so for this purpose this paper draw a comparison between other algorithms and the LASSO algorithm, this survey paper helps the upcoming researchers to make an impact in the their papers. The process happens in the paper is first moment of the research, it aim to understand and find daily trends in the Bitcoin market while gaining insight into optimal features surrounding Bitcoin price. Our data set consists of various features relating to the Bitcoin price and payment network over the course of every years, recorded daily. By preprocessing the dataset, it applies the some data mining techniques to reduce the noise of data. Then the second moment of our research, using the available information, it will predict the sign of the daily price change with highest possible accuracy.

**Title** : Bitcoin Price Prediction Using Machine Learning [3]

**Author:** Neha Mangla

**Year :** 2019

In this paper, Neha Mangala tried to estimate the Bitcoin price precisely taking into consideration various parameters that affect the Bitcoin value. In the work, Neha Mangala pointed to understand and identify daily changes in the Bitcoin market while obtaining insight into most appropriate features surrounding Bitcoin price. This will predict the daily price change with highest possible accuracy. The market capitalization of publicly traded cryptocurrencies is currently above $230 billion. Bitcoin, the most valuable cryptocurrency, serves primarily as a digital store of value, and its price

predictability has been well-studied. These characteristics are outlined in the following

subsection; the underlying details of Bitcoin,as it is described in depth in the cited papers.

**Title :** MODELING AND PREDICTION OF CRYPTOCURRENCY PRICES USING MACHINE LEARNING TECHNIQUES [4]

**Author:** Alireza Ashayer

#### Year : 2019

The contents of this chapter have been submitted to IEEE International Conference on Blockchain. The motivation for this review is to understand the trend of Blockchain research with respect to the machine learning field by studying and reviewing published articles. This understanding can help other researchers and practitioners with insight into the current state and future direction of research in this field. Given this motivation, it will review and verify the distribution of research papers by their year of publication and classify the research papers by the machine learning techniques used. To provide a comprehensive review of research papers.

**Title :** Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning [5]

**Author:** Franco Valencia, Alfonso Gómez-Espinosa and Benjamín Valdés-Aguirre

**Year :** 2019

Cryptocurrencies are becoming increasingly relevant in the financial world and can be considered as an emerging market. The low barrier of entry and high data availability of the cryptocurrency market makes it an excellent subject of study, from which it is possible to derive insights into the behavior of markets through the application of sentiment analysis and machine learning techniques for the challenging task of stock market prediction. While there have been some previous studies, most of them have focused exclusively on the behavior of Bitcoin. In this paper, we propose the usage of common machine learning tools and available social media data for

predicting the price movement of the Bitcoin, Ethereum, Ripple and Litecoin

cryptocurrency market movements. We compare the utilization of neural networks (NN), support vector machines (SVM) and random forest (RF) while using elements from Twitter and market data as input features. The results show that it is possible to predict cryptocurrency markets using machine learning and sentiment analysis, where Twitter data by itself could be used to predict certain cryptocurrencies and that NN outperform the other models.

**Title :** AUTOMATED CRYPTOCURRENCIES PRICES PREDICTION USING MACHINE LEARNING [6]

**Athor:** Ruchi Mittal, Shefali Arora and M.P.S Bhatia

**Year :** 2018

Currently, Cryptocurrency is one of the trending areas of research among researchers. Many researchers may analyze the cryptocurrency features in several ways such as market price prediction, the impact of cryptocurrency in real life and so on. In this paper, we focus on market price prediction of the number of cryptocurrencies based on their historical trend. For our study, we tried to understand and identify the daily trends in the cryptocurrency market which analyzing the features related to the price of cryptocurrency. Our dataset consists of over nine features relating to the cryptocurrency price recorded daily over the period of 6 months. We applied some machine-learning algorithms to predict the daily price change of cryptocurrencies.

# CHAPTER 3 SYSTEM ANALYSIS

### SYSTEM ANALYSIS

* 1. **Existing System**

It presented a computational approach for identifying and characterizing crypto currency pump and dump operations that are carried out in social media. It had used financial and Twitter data pertaining to a particular coin, the method used was able to detect, with reasonable accuracy, whether there is an unfolding attack on that coin on Telegram, and whether or not the resulting pump operation will succeed in terms of meeting the anticipated price targets. It also analysed activities of users involved in pump operations, and observe a prevalence of Twitter bots in cryptocurrency-related tweets in close proximity to the attack. Telegram was a popular choice for scammers to organize and coordinate pump and dump operations. To analyse such activities, let us define the following two notions one was Pump Attempt which is the act of targeting a coin on Telegram by posting a pump message mentioning the coin as a “pump attempt. The second was Successful Pump Attempt which is the pump attempt is successful if the actual price approaches the target price within a time window after the first pump message has been posted.

### Disadvantages:

1. It did not use any specific algorithms for predicting the crypto-currency price
2. The data which they found are only used to consider to see the only the impact from social media.

### Proposed System

Cryptocurrency behaves differently and it is little difficult to predict the possibility. The proposed model is build a model where the model is able to predict the price. The steps involved in the proposed model is a process used in data science from

variable identification to building a model. The process start from variable identification like dependent and independent variable where we find the target column.

Then the pre-processing techniques are applied like dealing with the missing values the pre-processed data then used to build a model by dividing the dataset into 7:3 ratio where 70% of the data is used for training purpose that is model learns the pattern and the remaining testing data is used to test the performance of data. The regression model also can be used to predict the price of the cryptocurrency.

### Advantages:

1. The machine learning algorithms are compared and the performance metric are also calculated for better prediction.
2. Machine learning model predictions allow businesses to make highly accurate guesses.

### Feasibility Study

#### Data Wrangling

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

#### Data collection

The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest, logistic, Decision tree algorithms and Support vector classifier (SVC) are applied on the Training set and based on the test result accuracy, Test set prediction is done.

#### Preprocessing

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done.

#### Building the Regression model

The prediction of cryptocurrency market price, A Random Forest Algorithm prediction model is effective because of the following reasons: It provides better results in regression problem.

It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.

It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

### Software Requirements:

Operating System : Windows 7,8,10,11

Tool : Anaconda with Jupyter Notebook

### Hardware requirements:

Processor : Pentium IV/III

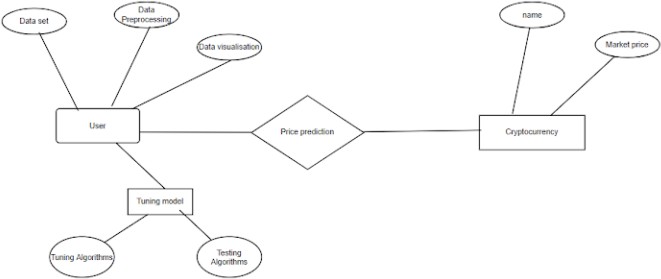
Hard disk : minimum 10 GB

RAM : minimum 4 GB

# CHAPTER 4 SYSTEM DESIGN

### SYSTEM DESIGN

* 1. **ER Diagram**

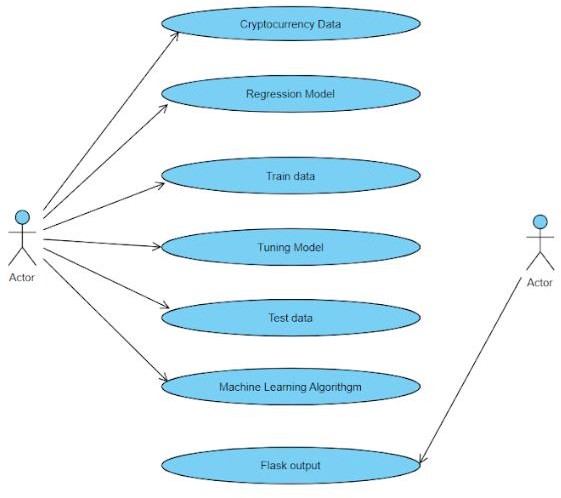


#### Figure 4.1 ER diagram

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a [data modeling](https://searchdatamanagement.techtarget.com/definition/data-modeling) technique that can help define business processes and be used as the foundation for a [relational database](https://searchdatamanagement.techtarget.com/definition/relational-database). Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

### UML DIAGRAM

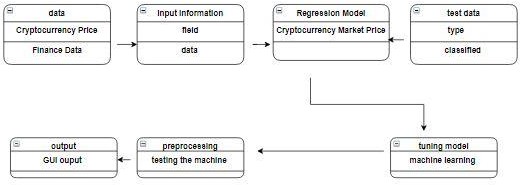
* + 1. **Use Case Diagram**



**Figure 4.2.1** 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.

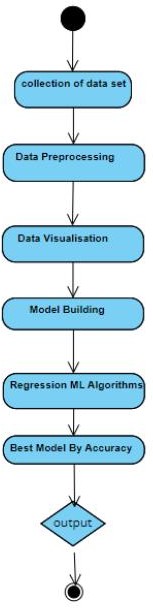
### Class Diagram:



**Figure 4.2.2** 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. Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

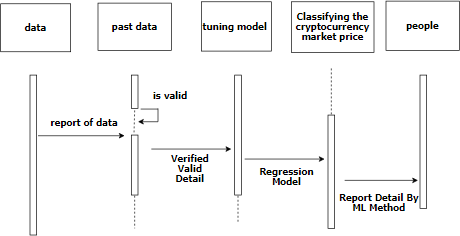
### Activity Diagram:



**Figure 4.2.3** Activity Diagram

Activity is a particular operation of the system. Activity diagrams are not only used for 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 missing thing in activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

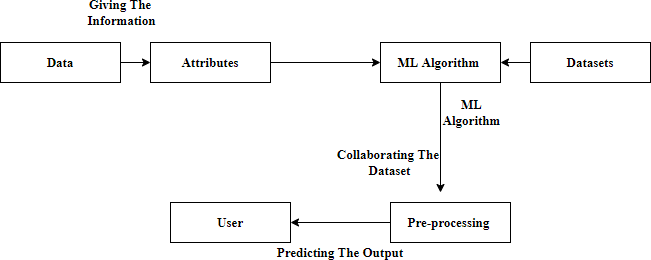
### Sequence Diagram:



**Figure 4.2.4** Sequence Diagram

Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system. Other dynamic modeling techniques include [activity](http://agilemodeling.com/artifacts/activityDiagram.htm) [diagramming](http://agilemodeling.com/artifacts/activityDiagram.htm), [communication diagramming](http://agilemodeling.com/artifacts/communicationDiagram.htm), [timing diagramming](http://agilemodeling.com/artifacts/timingDiagram.htm), and [interaction](http://agilemodeling.com/artifacts/interactionOverviewDiagram.htm) [overview diagramming](http://agilemodeling.com/artifacts/interactionOverviewDiagram.htm). Sequence diagrams, along with [class diagrams](http://agilemodeling.com/artifacts/classDiagram.htm) and [physical](http://agiledata.org/essays/dataModeling101.html) [data models](http://agiledata.org/essays/dataModeling101.html) are in my opinion the most important design-level models for modern business application development.

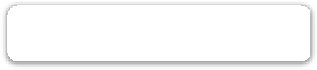
### Collaboration Diagram:



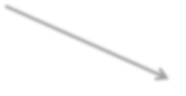
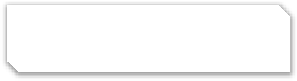
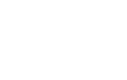
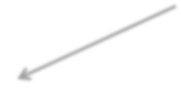
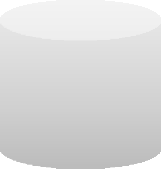
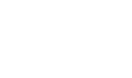
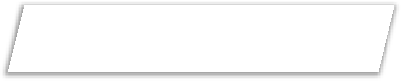
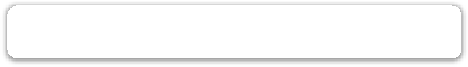
**Figure 4.2.5** Collaboration Diagram

A collaboration diagram is a type of visual presentation that shows how various software objects interact with each other within an overall IT architecture and how users (like doctor or patient) can benefit from this collaboration. A collaboration diagram often comes in the form of a visual chart that resembles a flow chart. It can show, at a glance, how a single piece of software complements other parts of a greater system.

### Workflow Diagram



Source Data



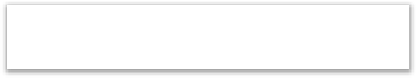
Data Processing and Cleaning

Training Dataset

Testing Dataset

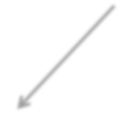
Regression ML Algorithms

Best Model by Accuracy



Finding Crypto-currency Market Price

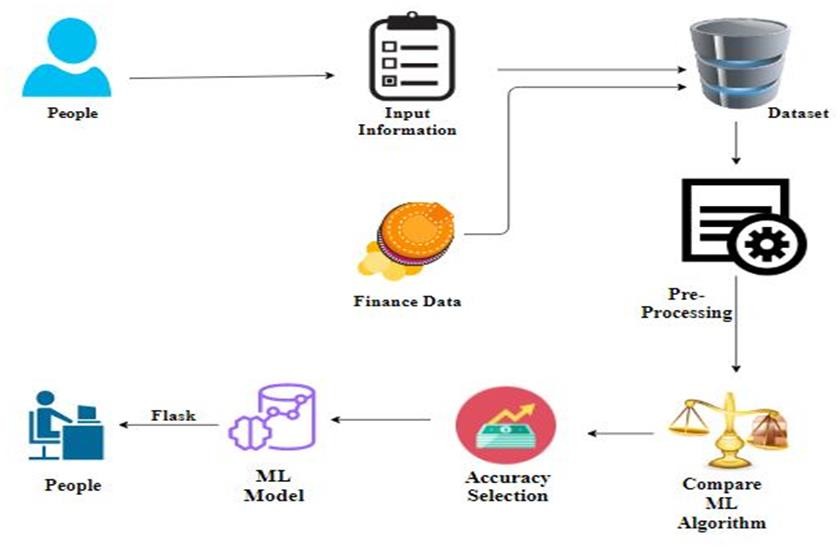
**Figure 4.3** Workflow Diagram



# CHAPTER 5 SYSTEM ARCHITECTURE

### SYSTEM ARCHITECTURE

* 1. **Architecture Diagram**



**Figure 5.1** Architecture Diagram

### MODULE DESIGN SPECIFICATION

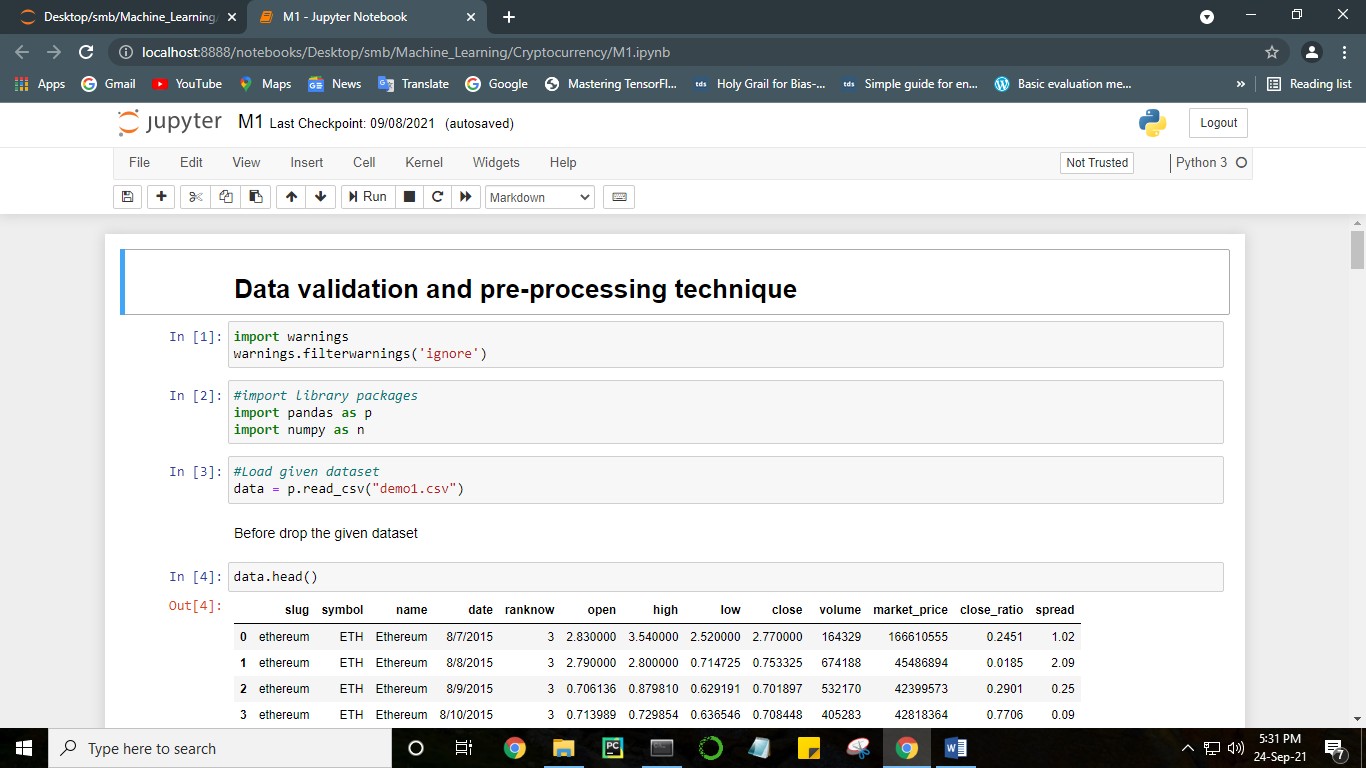
#### MODULE 1 :

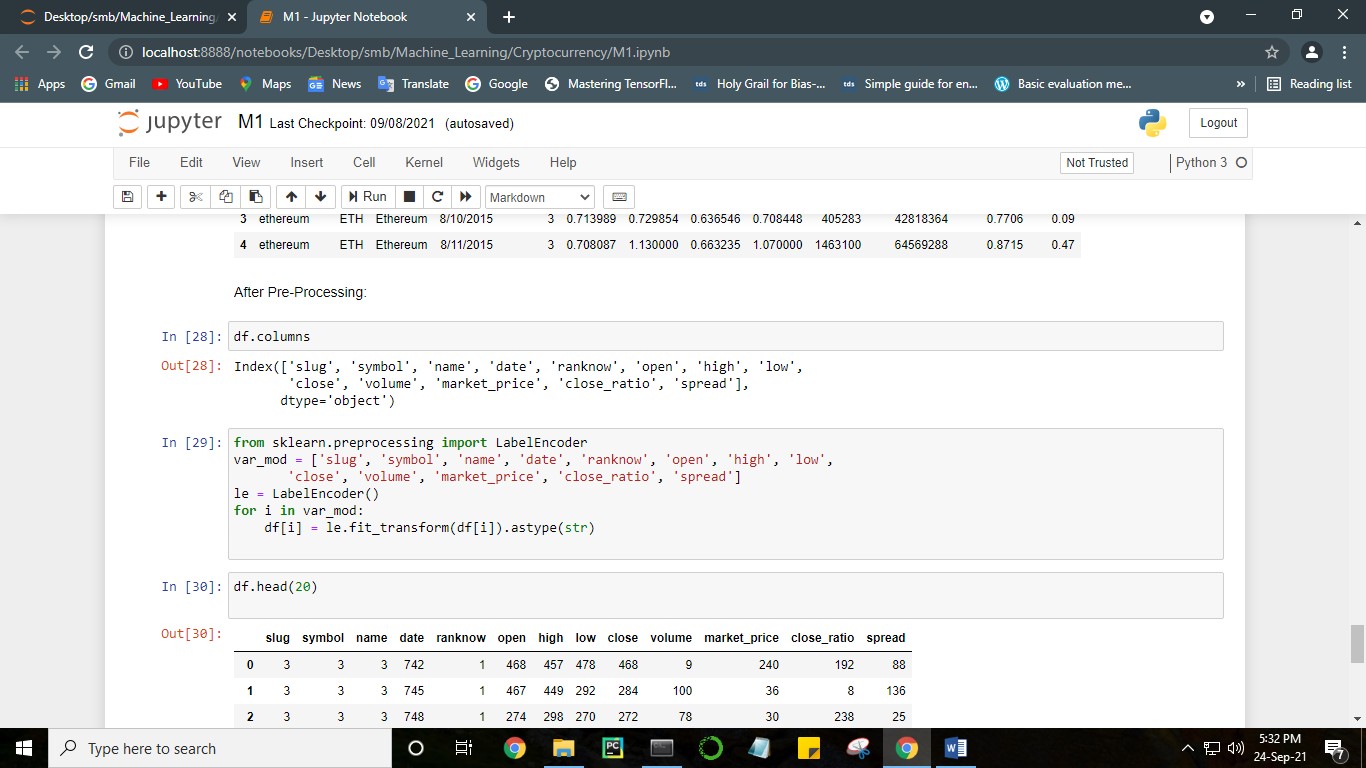
**Data Pre-processing**

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

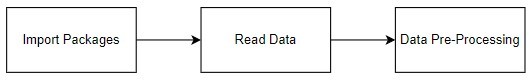
#### Data Validation/ Cleaning/Preparing Process

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.





#### MODULE DIAGRAM:



**GIVEN INPUT EXPECTED OUTPUT**

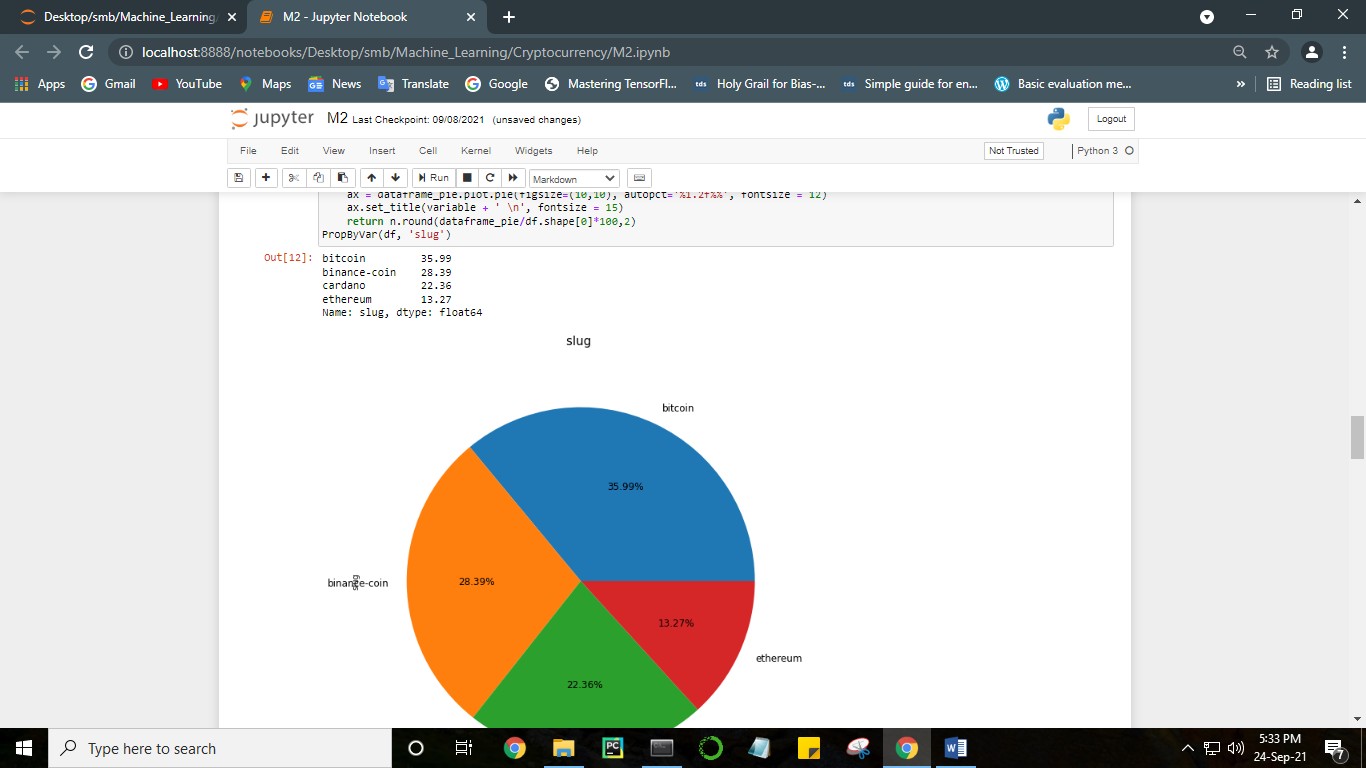
input : data

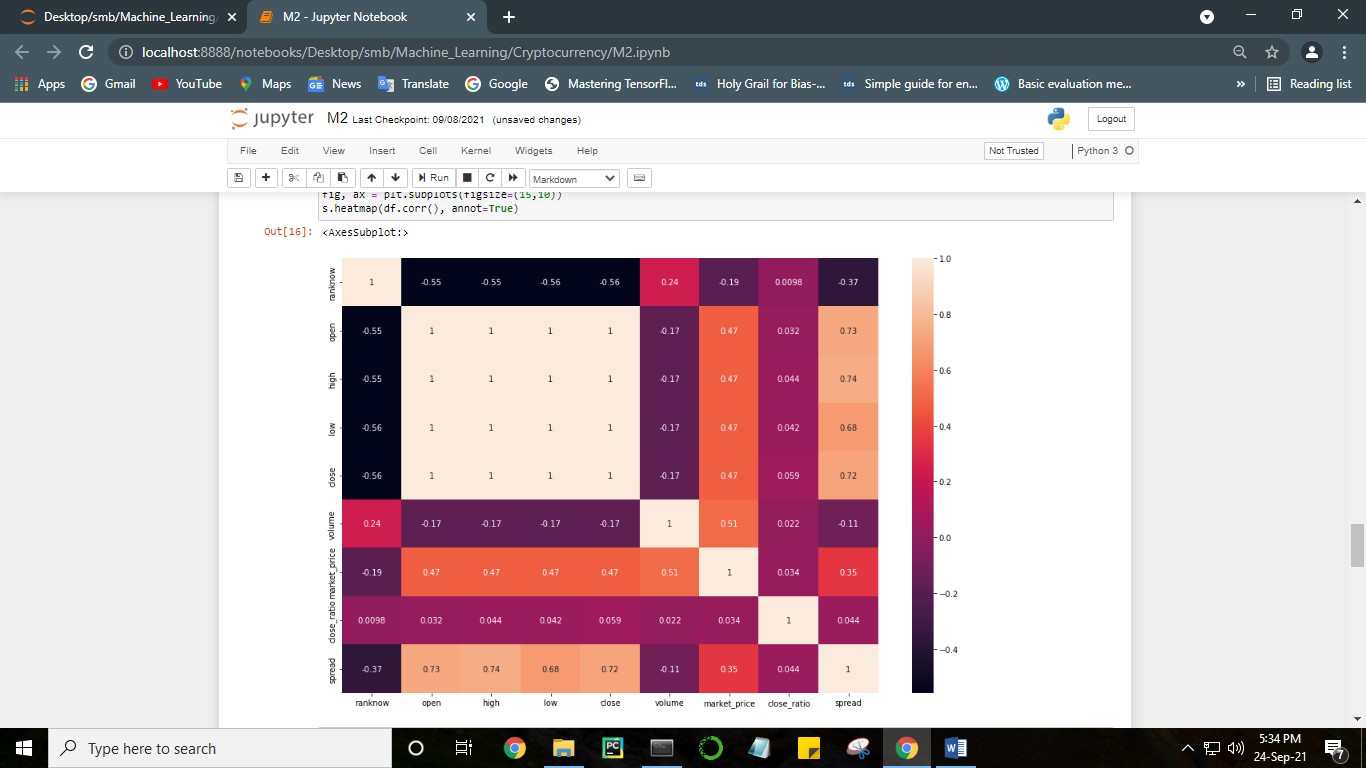
output : removing noisy data

#### MODULE 2 :

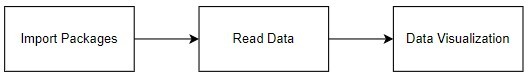
**Exploration data analysis of visualization**

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.





#### MODULE DIAGRAM:



**GIVEN INPUT EXPECTED OUTPUT**

input : data

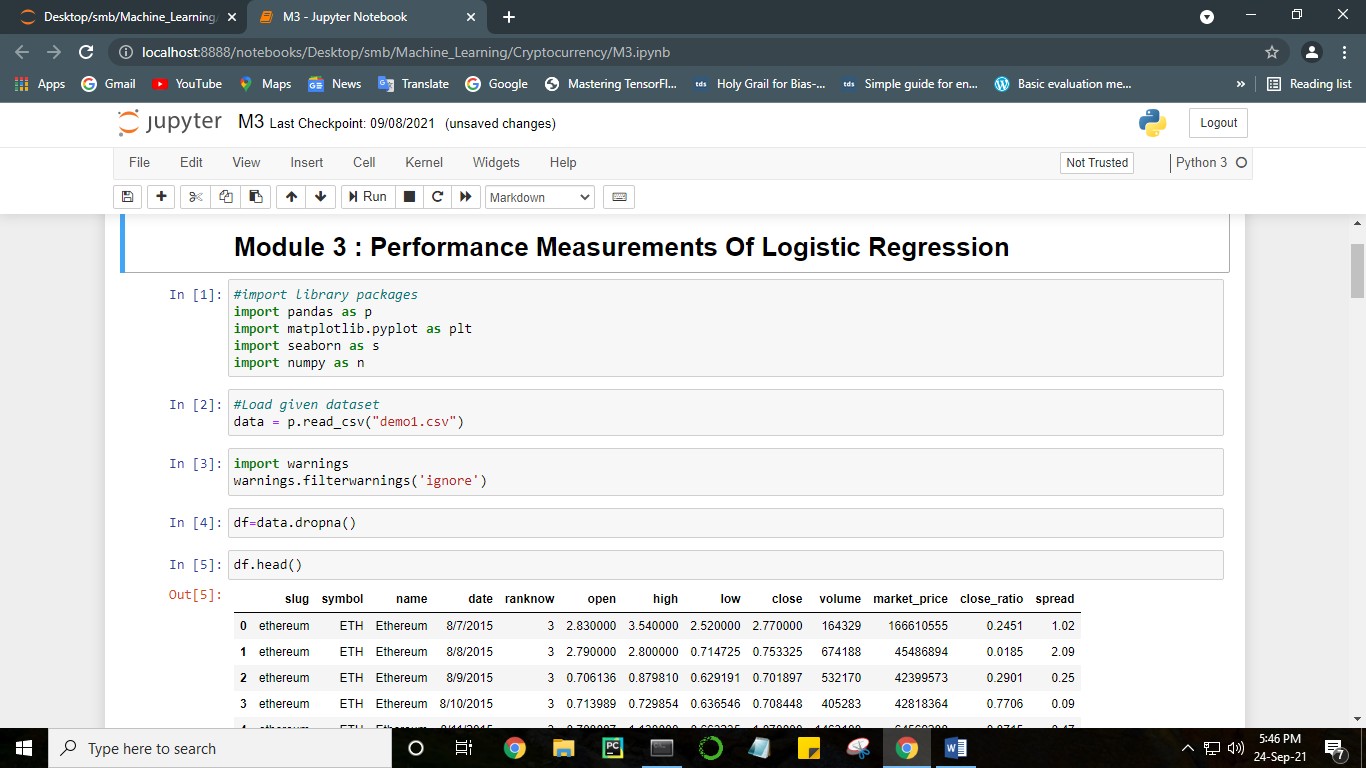
output : visualized data

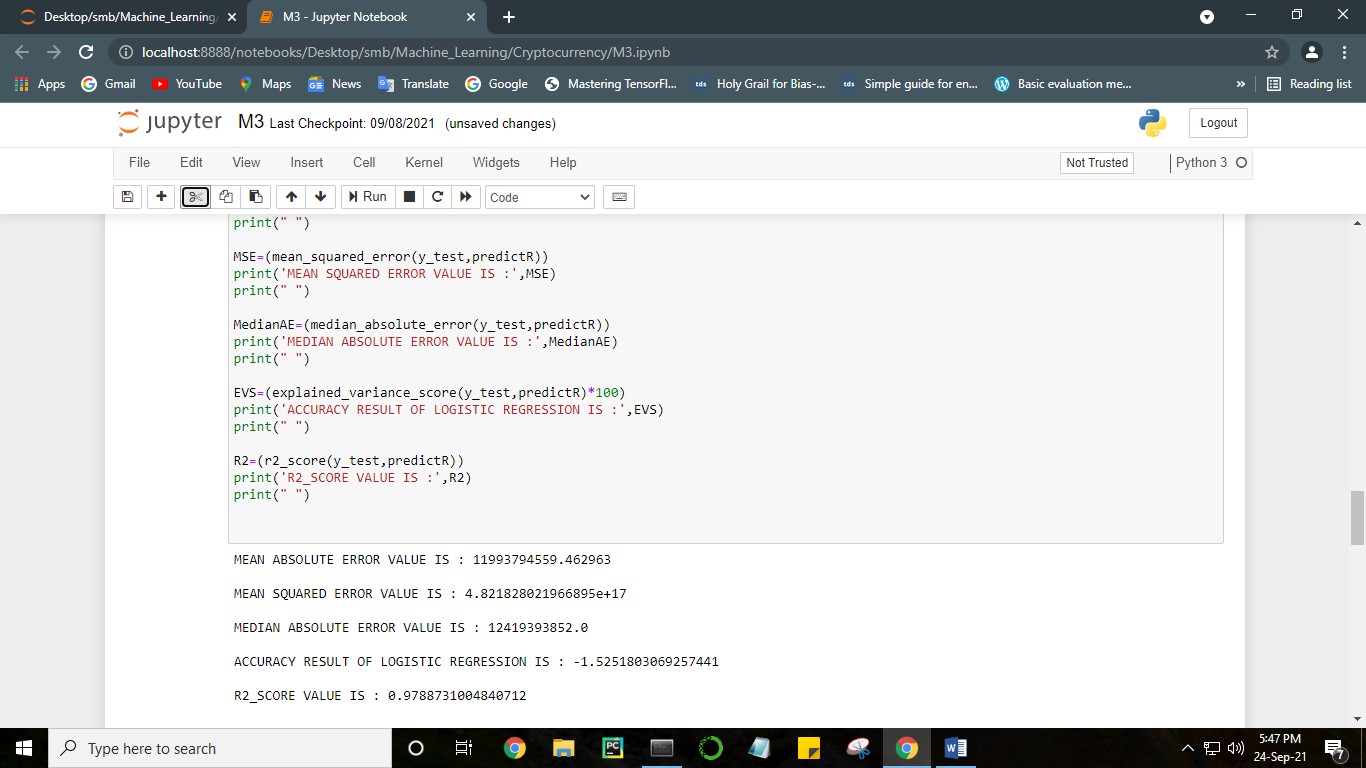
#### MODULE 3:

[**Logistic Regression**](https://en.wikipedia.org/wiki/Logistic_regression)

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables.

Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).





#### Linear Regression:

Linear Regression is a machine learning algorithm based on supervised learning. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output).

It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

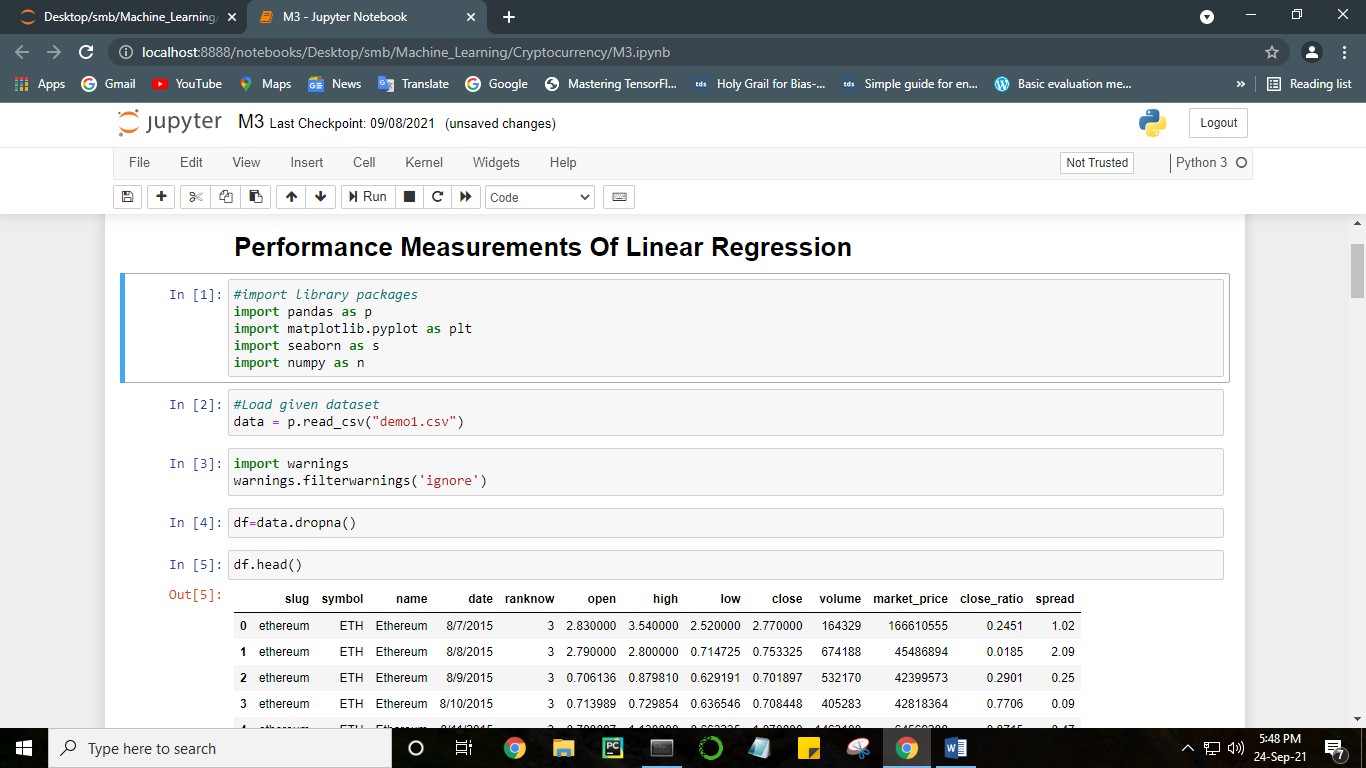
Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.

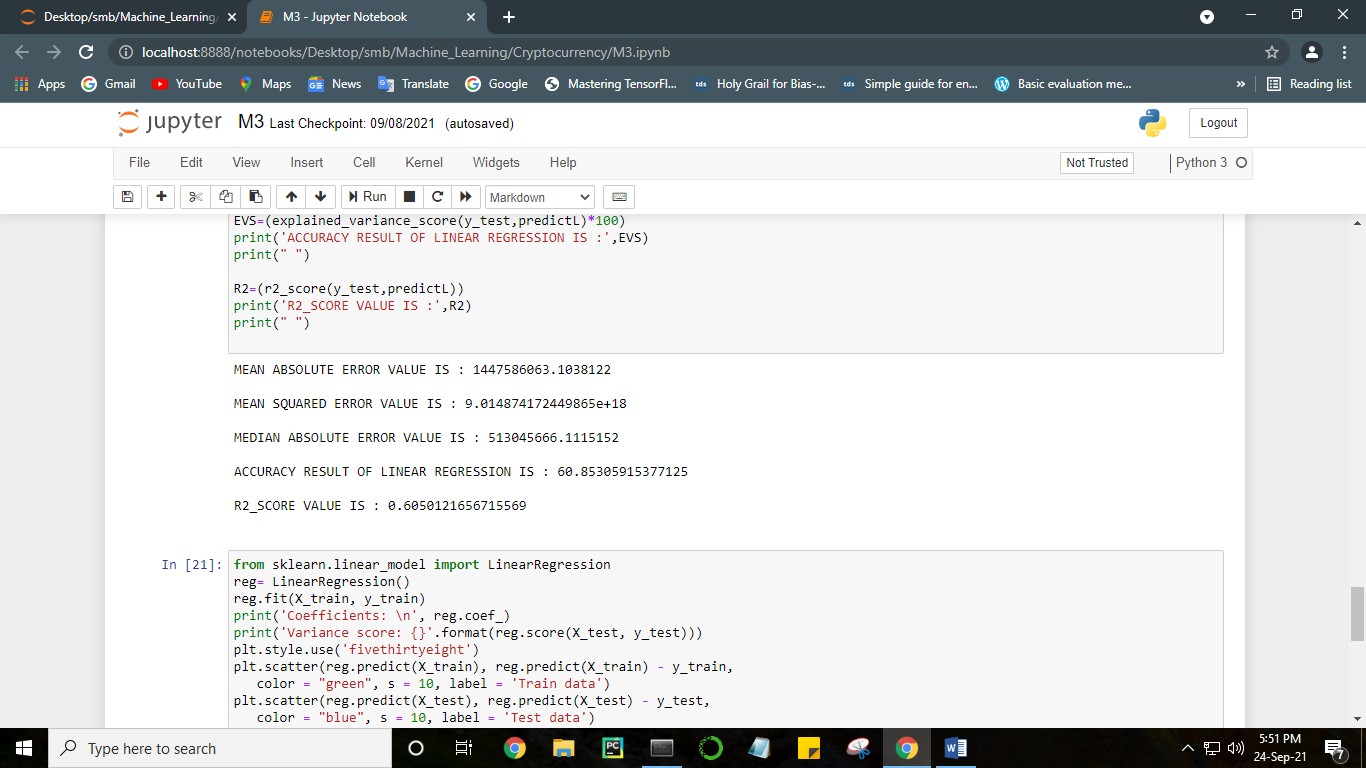
In the figure above, X (input) is the work experience and Y (output) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

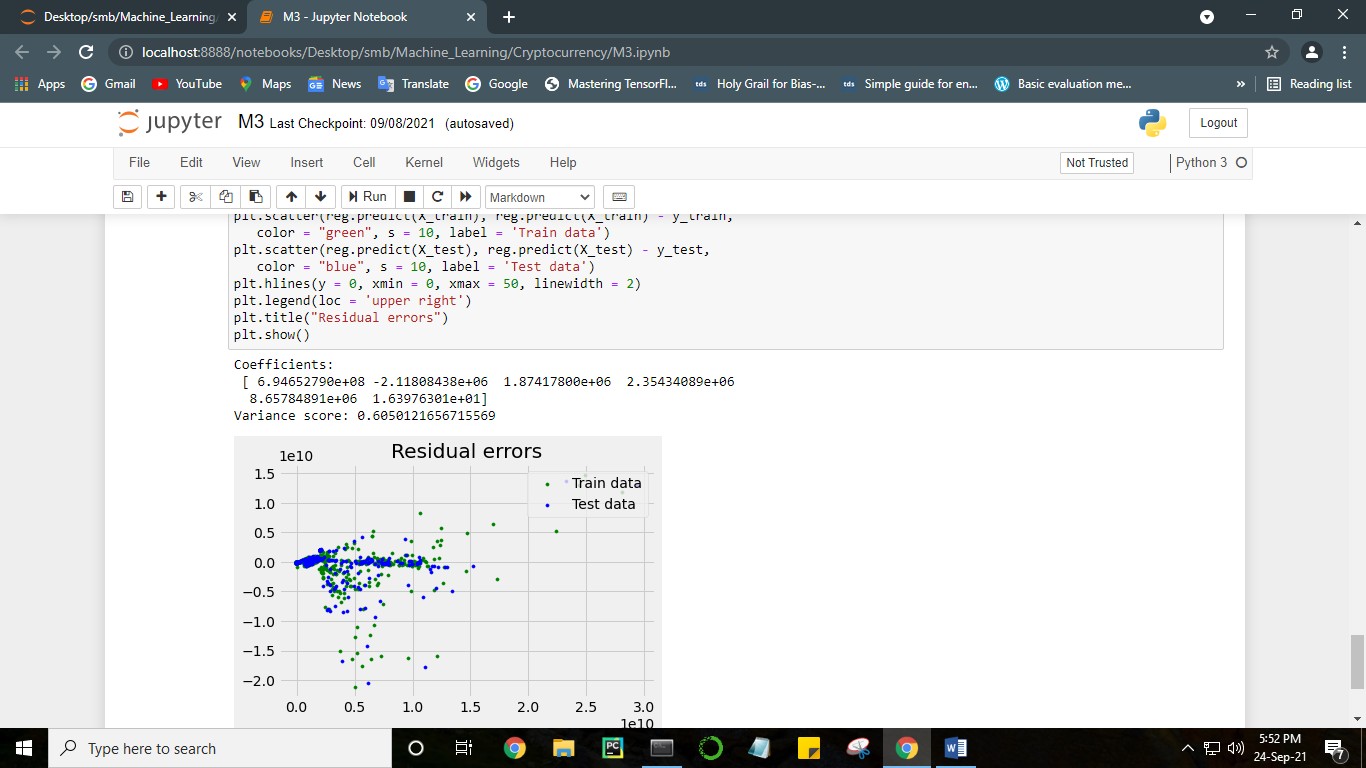
Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

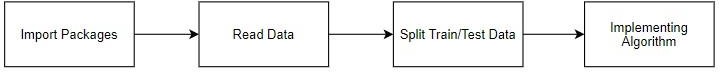
The linear regression model provides a sloped straight line representing the relationship between the variables.







#### MODULE DIAGRAM



**GIVEN INPUT EXPECTED OUTPUT**

Input : Data

Output : Getting Accuracy

#### MODULE 4:

**Decision Tree Regression:**

Decision Tree - Regression. Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

31

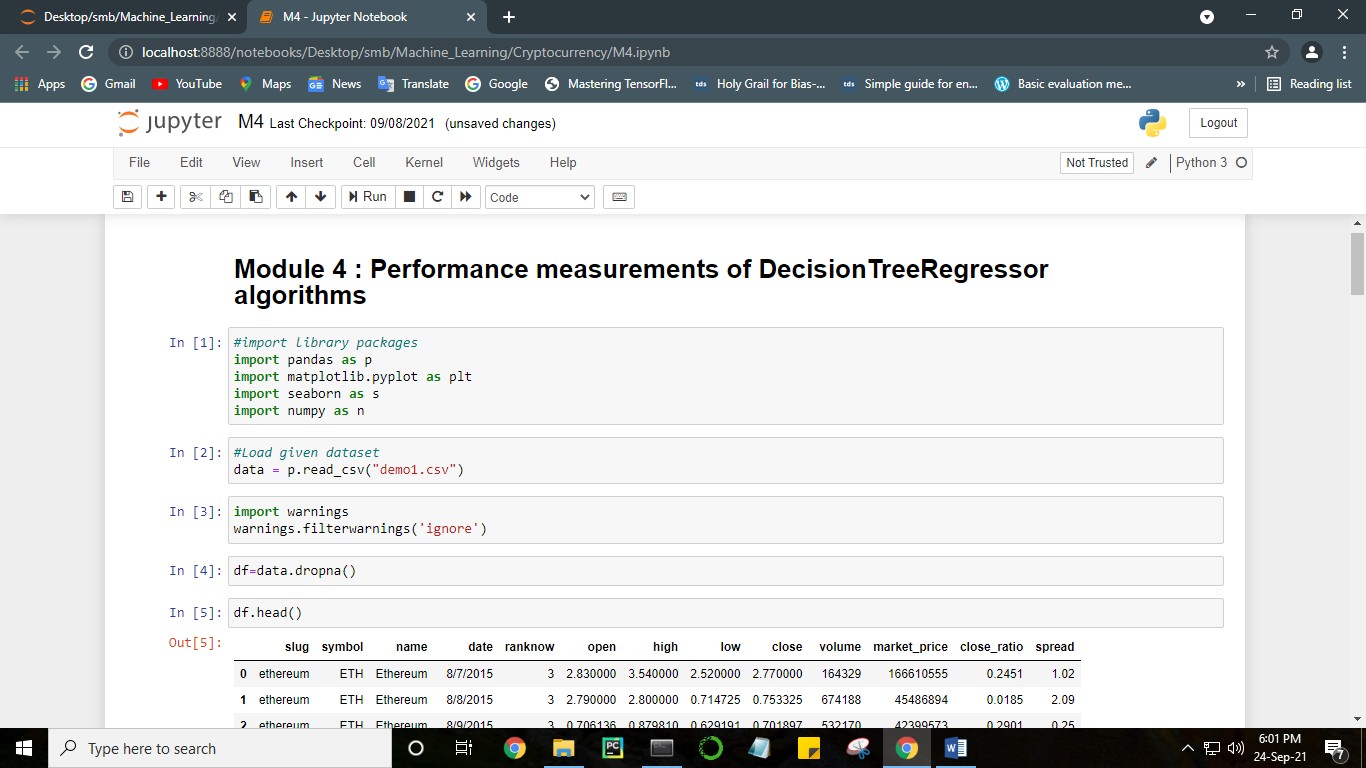
Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

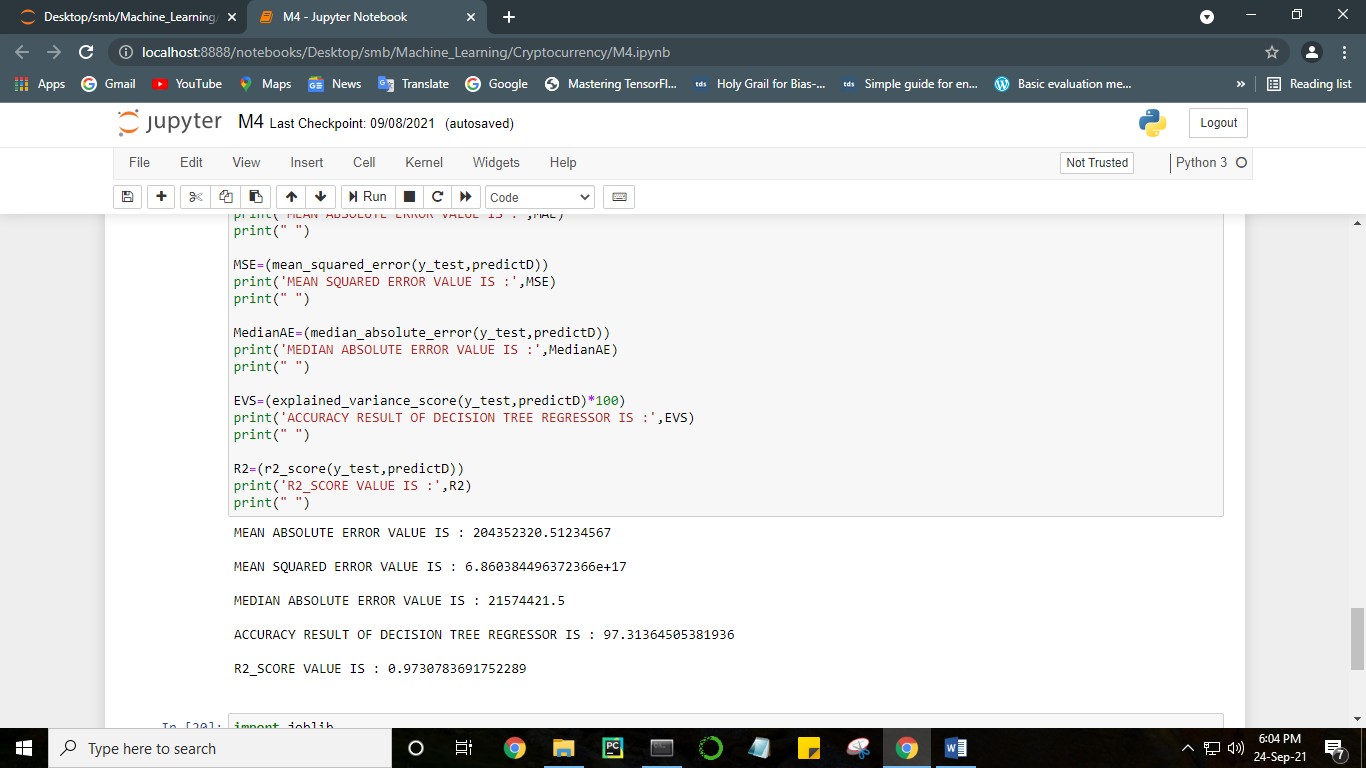
32

The final result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

Decision Tree is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application. It is a tree-structured classifier with three types of nodes.

Decision trees where the target variable can take continuous values (typically [real](https://en.wikipedia.org/wiki/Real_numbers) [numbers](https://en.wikipedia.org/wiki/Real_numbers)) are called [regression](https://en.wikipedia.org/wiki/Regression_analysis) [trees](https://en.wikipedia.org/wiki/Decision_tree). Decision trees are among the most popular machine learning algorithms given their intelligibility and simplicity.





#### Random Forest Regression:

**Random Forest Regression** is a supervised learning algorithm that uses **ensemble learning** method for regression.

Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

we have a basic understanding of how the Random Forest Regression model works, we can assess its performance on a real-world dataset. Similar to my previous posts, I will be using data on House Sales in King County, USA.

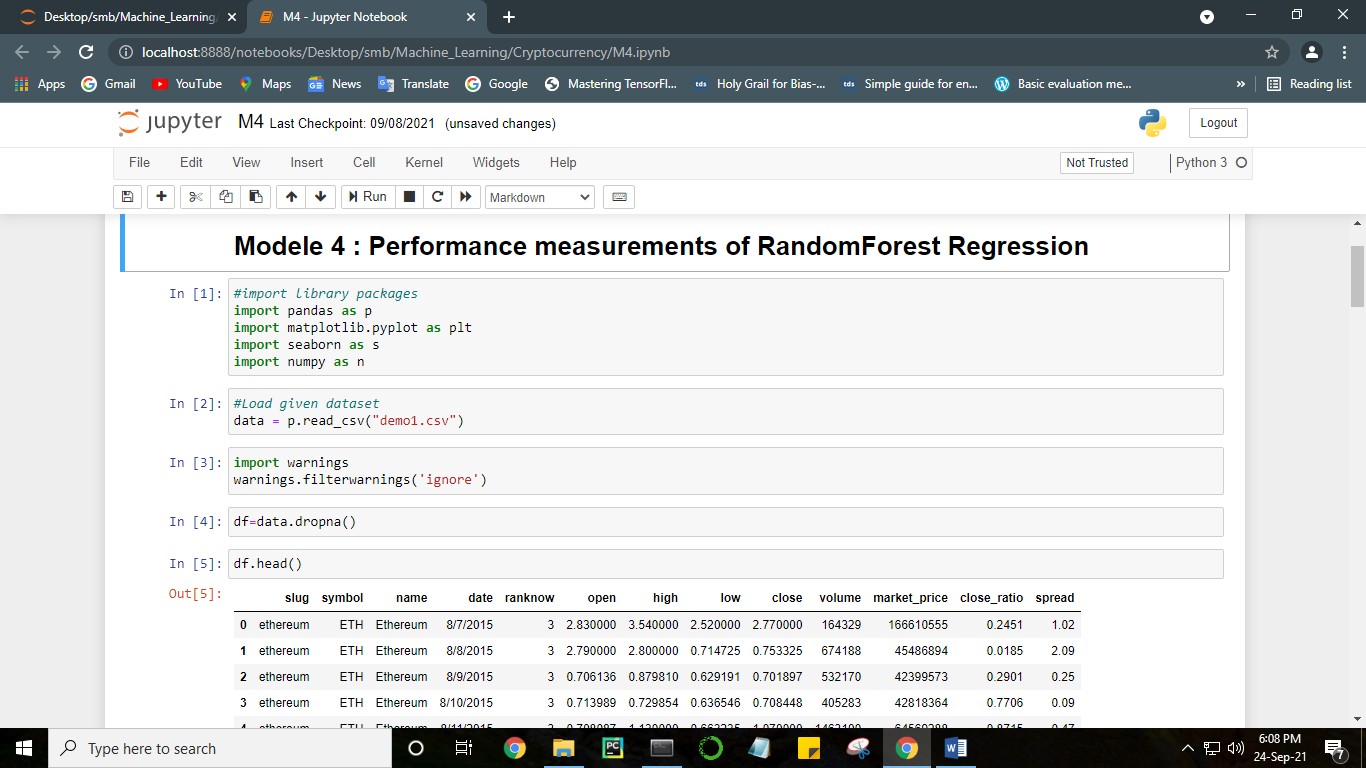
After importing the libraries, importing the dataset, addressing null values, and dropping any necessary columns, we are ready to create our Random Forest Regression model

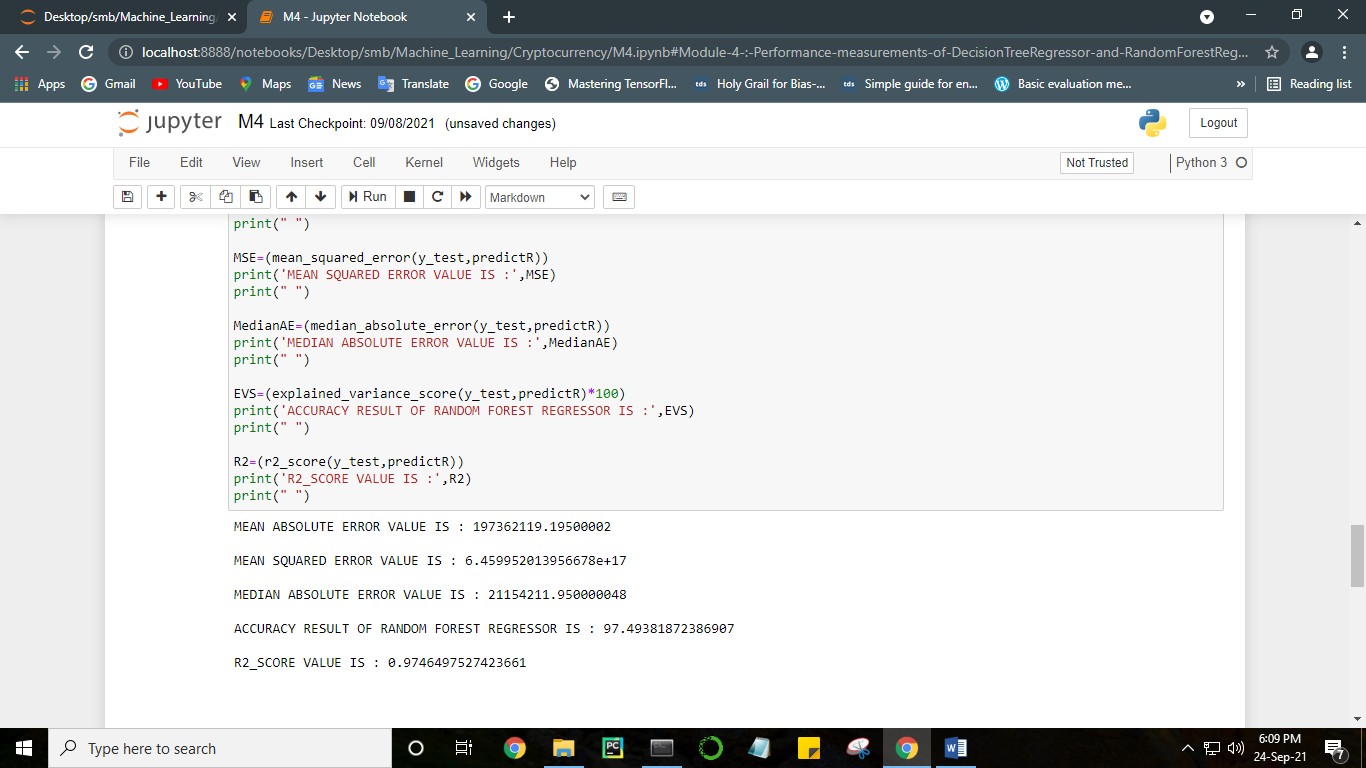
From the sklearn package containing ensemble learning, we import the class **Random Forest Regressor**, create an instance of it, and assign it to a variable.

The parameter *n\_estimators* creates *n* number of trees in your random forest, where *n* is the number you pass in. We passed in 10.

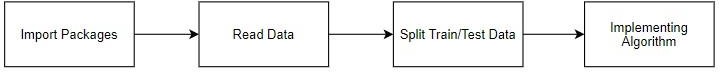
The .*fit()* function allows us to train the model, adjusting weights according to the data values in order to achieve better accuracy. After training, our model is ready to make predictions, which is called by the .*predict()* method.

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max\_samples parameter if Bootstrap = True (default), otherwise the whole dataset is used to build each tree.





#### MODULE DIAGRAM:



**GIVEN INPUT EXPECTED OUTPUT**

Input : Data

Output : Getting Accuracy

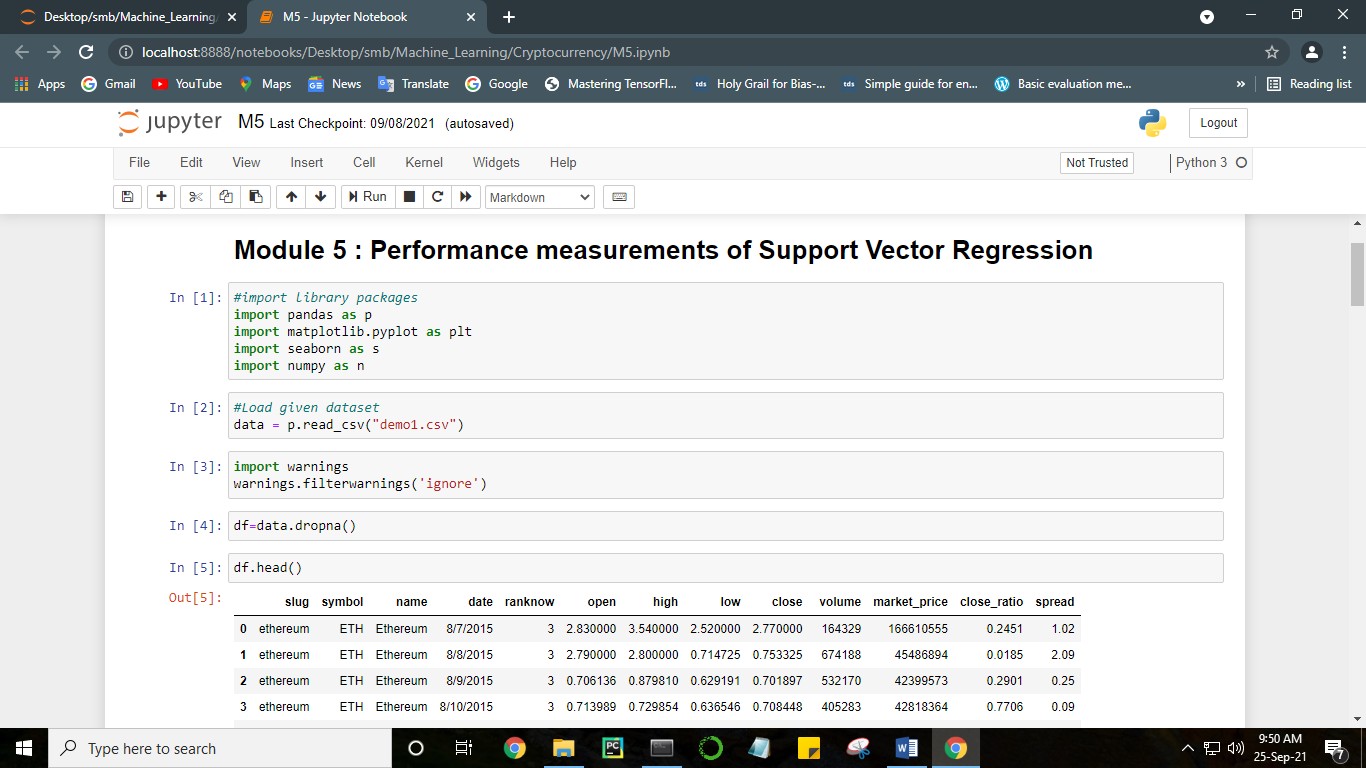
#### MODULE 5 :

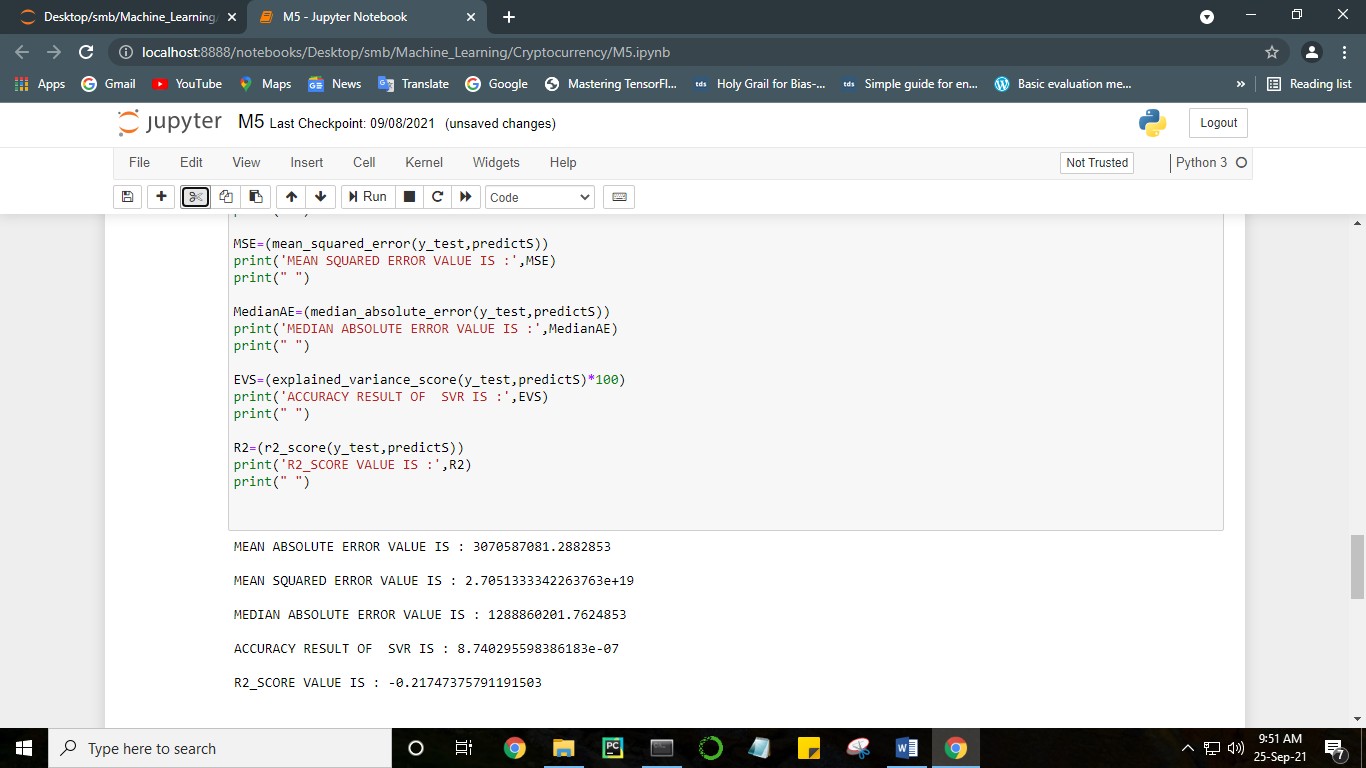
**Support Vector Regression:**

Support Vector Machines (SVM) are popularly and widely used for classification and regression problems in machine learning.

The regression problem is a generalization of the classification problem, in which the model returns a continuous-valued output, as opposed to an output from a finite set. In other words, a regression model estimates a continuous-valued multivariate function.

SVMs solve binary classification problems by formulating them as convex optimization problems. The optimization problem entails finding the maximum margin separating the hyperplane, while correctly classifying as many training points as possible. SVMs represent this optimal hyperplane with support vectors. The sparse solution and good generalization of the SVM lend themselves to adaptation to regression problems. SVM generalization to SVR is accomplished by introducing an *ε*-insensitive region around the function, called the *ε*-tube. This tube reformulates the optimization problem to find the tube that best approximates the continuous- valued function, while balancing model complexity and prediction error. More specifically, SVR is formulated as an optimization problem by first defining a convex *ε*-insensitive loss function to be minimized and finding the flattest tube that contains most of the training instances. Hence, a multi-objective function is constructed from the loss function and the geometrical properties of the tube. Then, the convex optimization, which has a unique solution, is solved, using appropriate numerical optimization algorithms. The hyperplane is represented in terms of support vectors, which are training samples that lie outside the boundary of the tube. As in SVM, the support vectors in SVR are the most influential instances that affect the shape of the tube, and the training and test data are assumed to be independent and identically distributed (iid), drawn from the same fixed but unknown probability distribution function in a supervised-learning context.





#### Lasso Regression:

Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values

are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters).

The word “LASSO” stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator. It is a statistical formula for the regularisation of data models and feature selection.

Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean.

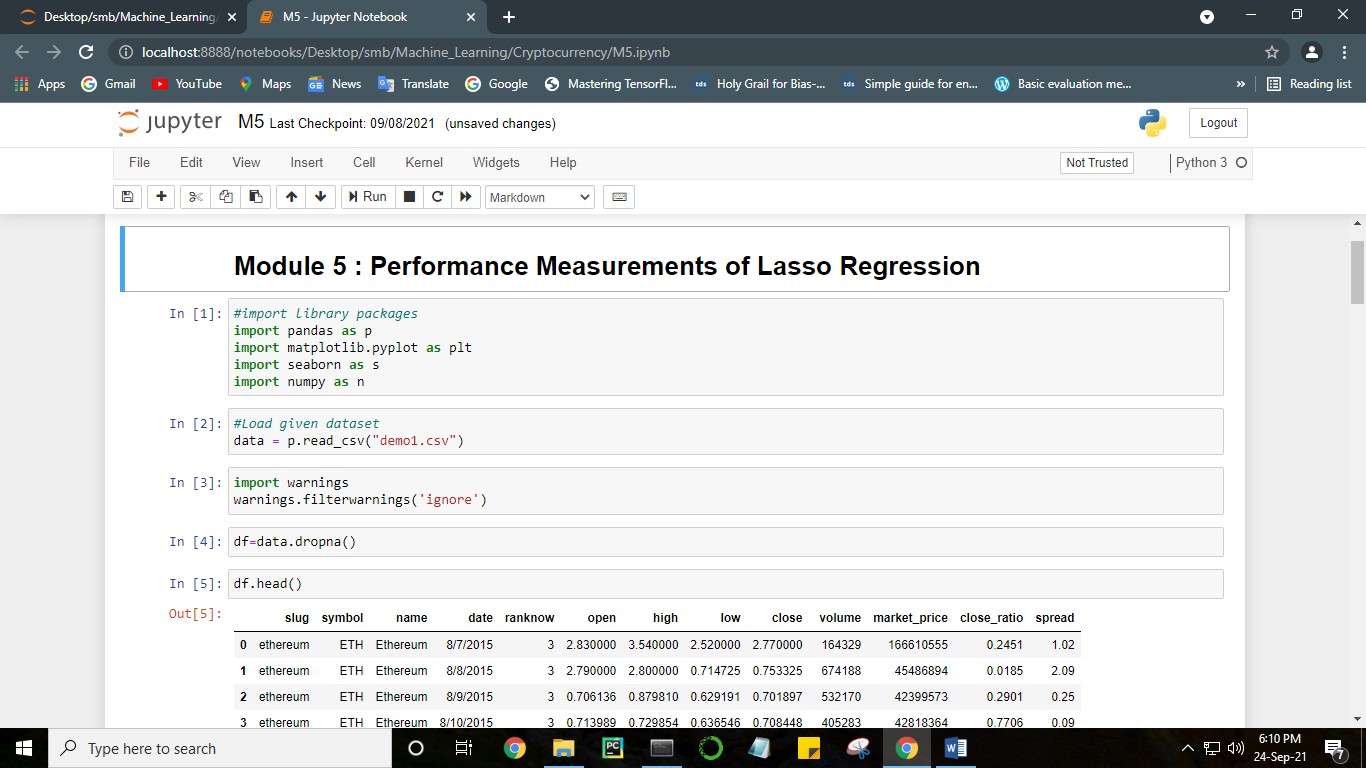
The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

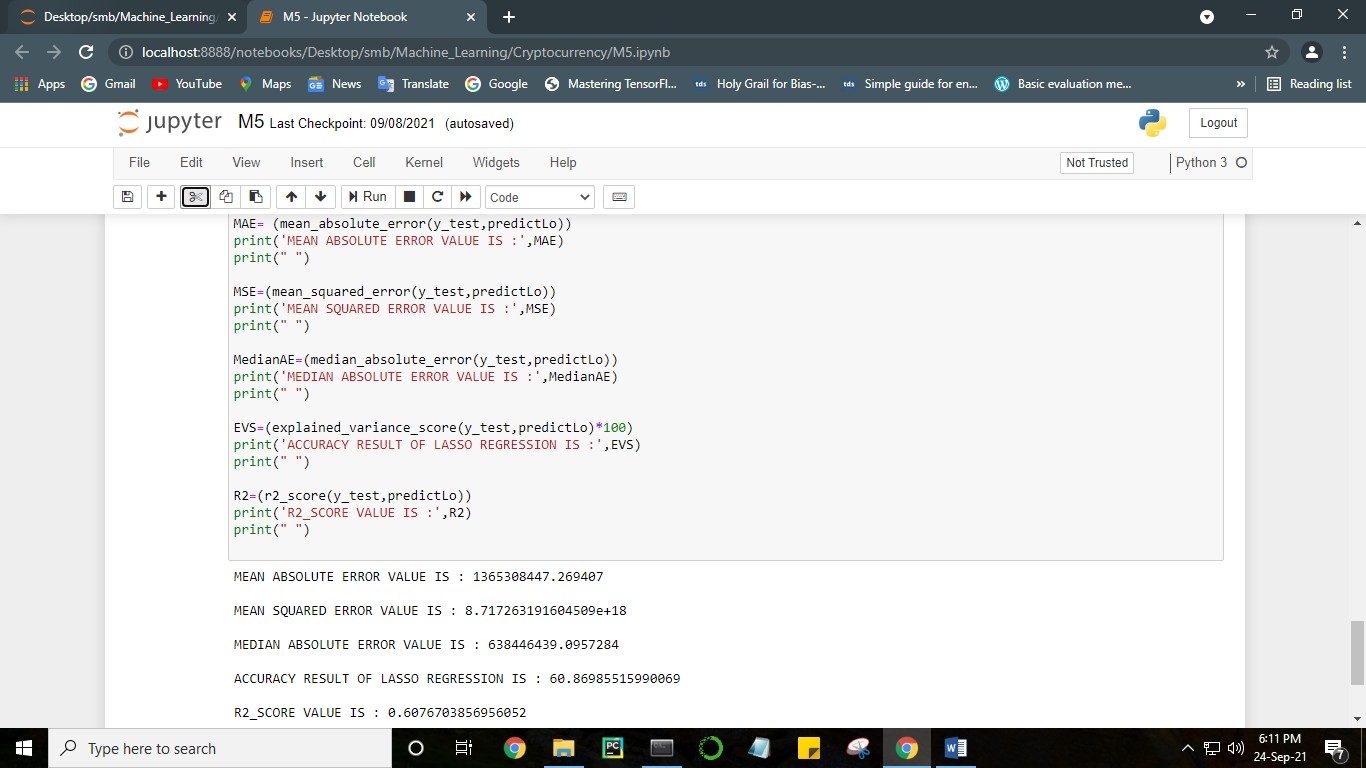
Lasso Regression uses L1 regularization technique (will be discussed later in this article). It is used when we have more number of features because it automatically performs feature selection.

The L1 regularization performed by Lasso, causes the regression coefficient of the less contributing variable to shrink to zero or near zero.

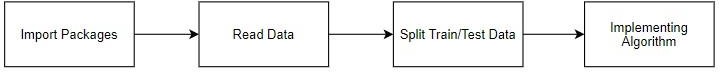
Lasso regression uses shrinkage, where the data values are shrunk towards a central point such as the mean value.

The Lasso penalty shrinks or reduces the coefficient value towards zero. The less contributing variable is therefore allowed to have a zero or near-zero coefficient.





#### MODULE DIAGRAM:



**GIVEN INPUT EXPECTED OUTPUT**

input : data

output : getting accuracy

### TOOLS AND PACKAGES

#### ANACONDA NAVIGATOR

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

Anaconda Now, if you are primarily doing data science work, Anaconda is also a great option. Anaconda is created by Continuum Analytics, and it is a Python distribution that comes preinstalled with lots of useful python libraries for data science. 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.

#### JUPYTER NOTEBOOK

This website acts as “meta” documentation for the Jupyter ecosystem. It has a collection of resources to navigate the tools and communities in this ecosystem, and to help you get started.

Project Jupyter is a project and community whose goal is to "develop open- source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Perez. Notebook documents are documents produced by the [Jupyter Notebook App](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#notebook-app), which contain both computer code (e.g. python) and rich text elements (paragraph,

equations, figures, links, etc…).

Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc.) as well as executable documents which can be run to perform data analysis.

#### Used Python Packages:

**sklearn:**

* In python, sklearn is a machine learning package which include a lot of ML algorithms.
* Here, we are using some of its modules like train\_test\_split, DecisionTreeClassifier or Logistic Regression and accuracy\_score.

#### NumPy:

* It is a numeric python module which provides fast maths functions for calculations.
* It is used to read data in numpy arrays and for manipulation purpose.

#### Pandas:

* Used to read and write different files.
* Data manipulation can be done easily with data frames.

#### Matplotlib:

* Data visualization is a useful way to help with identify the patterns from given dataset.
* Data manipulation can be done easily with data frames.

41

# CHAPTER 6 SYSTEM TESTING

### SYSTEM TESTING

#### UNIT TESTING

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process.

Unit testing is software verification and validation method in which the individual units of source code are tested fit for use. A unit is the smallest testable part of an application. In this testing, each class is tested to be working satisfactorily.

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

#### INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when put together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an

integration test plan to those aggregates, and delivers as its output the integrated system ready. All the errors found in the system are corrected for the next phase.

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctlyfor example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing.

* 1. **TEST CASE & REPORT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TESTCASE ID | INPUT | ACTION | EXPECTED OUTPUT | ACTUAL OUTPUT | RESULT (PASS/F AIL) |
| TC\_ID\_001 | Name: Bitcoin Open:763.28  close:735.07 High:777.51 Low:713.6  Volume:  46862700 | Machine learning algorithms trains the dataset and predicts the appropriate result | 8955394564 | 8955394564 | PASS |
| TC\_ID\_002 | Name: Bitcoin Open:737.98  close: 727.83  High: 747.06 Low:705.35  Volume: 32505800 | Machine learning algorithms trains the dataset and predicts the appropriate result | 8869918644 | 8869918644 | PASS |
| TC\_ID\_003 | Name: Bitcoin Open: 728.05  close: 745.05  High: 748.61  Low: 714.44  Volume: 19011300 | Machine learning algorithms trains the dataset and predicts the appropriate result | 9082103621 | 9074361342 | FAIL |
| TC\_ID\_004 | Name: Ethereum Open: 2.83 close:2.77  High: 3.54  Low: 2.52  Volume: 164329 | Machine learning algorithms trains the dataset and predicts the appropriate result | 166610555 | 166610555 | PASS |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC\_ID\_005 | Name: Ethereum Open: 2.79 close:0.753325 High:2.8  Low: 0.714725  Volume: 674188 | Machine learning algorithms trains the dataset and predicts the appropriate result | 45486894 | 45486894 | PASS |
| TC\_ID\_006 | Name: Cardano Open: 0.024607  close: 0.025932  High: 0.030088  Low: 0.019969  Volume: 57641300 | Machine learning algorithms trains the dataset and predicts the appropriate result | 628899052 | 628899052 | PASS |
| TC\_ID\_007 | Name: Cardano Open: 0.025757  close: 0.020816  High: 0.027425  Low: 0.020864  Volume: 16997800 | Machine learning algorithms trains the dataset and predicts the appropriate result | 539692715 | 546475673 | FAIL |
| TC\_ID\_008 | Name: binance- coin  Open: 0.115203  close: 0.10587  High: 0.115841  Low: 0.101531  Volume: 145269 | Machine learning algorithms trains the dataset and predicts the appropriate result | 10587000 | 104579343 | FAIL |
| TC\_ID\_009 | Name: binance- coin  Open: 0.105893  close: 0.10587  High: 0.109013  Low: 0.099266  Volume:200395 | Machine learning algorithms trains the dataset and predicts the appropriate result | 10513800 | 10513800 | PASS |

# CHAPTER 7 CONCLUSION

### CONCLUSION

#### CONCLUSION

The process started from data cleaning and processing, missing value, exploratory analysis ,eventually model building and evaluation. the simplest accuracy on public test set is higher accuracy score is are learn. This application can help to seek out the Cryptocurrency marketvalue.

#### FUTURE ENHANCEMENT

Cryptocurrency market value prediction to attach with AI model.To automate this process by show the prediction lead to web application or desktop application.To optimize the work to implement in computer science environment

### APPENDICES

* 1. **Coding**

#### Module – 1

**Data Validation And Pre-Processing Technique import** warnings

warnings**.**filterwarnings('ignore')

#import library packages **import** pandas **as** p **import** numpy **as** n

#Load given dataset

data **=** p**.**read\_csv("demo1.csv")

Before drop the given dataset data**.**head()

data**.**shape

After drop the given dataset df**=**data**.**dropna()

df**.**head()

df**.**shape

#show columns df**.**columns

df['slug']**.**unique()

df['ranknow']**.**unique()

df["symbol"]**.**unique()

df["open"]**.**unique()

df["high"]**.**unique()

df["low"]**.**unique()

df["close"]**.**unique()

#To describe the dataframe df**.**describe()

#Checking datatype and information about dataset df**.**info()

Checking duplicate values of dataframe

#Checking for duplicate data df**.**duplicated()

#find sum of duplicate data sum(df**.**duplicated())

#Checking sum of missing values df**.**isnull()**.**sum()

df**.**columns df['market\_price']**.**value\_counts()

print("Minimum market\_price value is:", df**.**market\_price**.**min()) print("Maximum market\_price value is:", df**.**market\_price**.**max()) p**.**Categorical(df['slug'])**.**describe() p**.**Categorical(df['market\_price'])**.**describe()

df**.**corr()

Before Pre-Processing:

df**.**head()

After Pre-Processing:

df**.**columns

**from** sklearn.preprocessing **import** LabelEncoder

var\_mod **=** ['slug', 'symbol', 'name', 'date', 'ranknow', 'open', 'high', 'low', 'close', 'volume', 'market\_price', 'close\_ratio', 'spread']

le **=** LabelEncoder()

**for** i **in** var\_mod:

df[i] **=** le**.**fit\_transform(df[i])**.**astype(str) df**.**head(20)

#### Module - 2

**Exploration data analysis of visualization and training a model by given attributes**

**import** warnings warnings**.**filterwarnings('ignore')

#import library packages

**import** pandas **as** p

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** s

**import** numpy **as** n

data **=** p**.**read\_csv("demo1.csv") df**=**data**.**dropna()

df**.**columns

#Histogram Plot

df['slug']**.**hist(figsize**=**(7,6), color**=**'green', alpha**=**0.7) plt**.**xlabel('slug')

plt**.**ylabel('No of slug') plt**.**title('Number of slug')

#Histogram Plot

df['symbol']**.**hist(figsize**=**(7,6), color**=**'green', alpha**=**0.7) plt**.**xlabel('symbol')

plt**.**ylabel('No of symbol') plt**.**title('Number of symbol')

#Histogram Plot

df['ranknow']**.**hist(figsize**=**(7,6), color**=**'green', alpha**=**0.7) plt**.**xlabel('ranknow')

plt**.**ylabel('No of ranknow') plt**.**title('Number of ranknow')

#Histogram Plot

df['open']**.**hist(figsize**=**(7,6), color**=**'green', alpha**=**0.7) plt**.**xlabel('open')

plt**.**ylabel('No of open') plt**.**title('Number of open')

#Histogram Plot

df['close']**.**hist(figsize**=**(7,6), color**=**'green', alpha**=**0.7) plt**.**xlabel('close')

plt**.**ylabel('No of close') plt**.**title('Number of close')

#Propagation by variable

**def** PropByVar(df, variable):

dataframe\_pie **=** df[variable]**.**value\_counts()

ax **=** dataframe\_pie**.**plot**.**pie(figsize**=**(10,10), autopct**=**'%1.2f%%', fontsize **=** 12) ax**.**set\_title(variable **+** ' \n', fontsize **=** 15)

**return** n**.**round(dataframe\_pie**/**df**.**shape[0]**\***100,2) PropByVar(df, 'slug')

#Propagation by variable

**def** PropByVar(df, variable):

dataframe\_pie **=** df[variable]**.**value\_counts()

ax **=** dataframe\_pie**.**plot**.**pie(figsize**=**(10,10), autopct**=**'%1.2f%%', fontsize **=** 12) ax**.**set\_title(variable **+** ' \n', fontsize **=** 15)

**return** n**.**round(dataframe\_pie**/**df**.**shape[0]**\***100,2) PropByVar(df, 'ranknow')

# Heatmap plot diagram

fig, ax **=** plt**.**subplots(figsize**=**(15,10)) s**.**heatmap(df**.**corr(), annot**=True**)

fig, ax **=** plt**.**subplots(figsize**=**(16,8)) ax**.**scatter(df['slug'],df['market\_price']) ax**.**set\_xlabel('slug') ax**.**set\_ylabel('price')

plt**.**show() df**.**columns

**from** sklearn.preprocessing **import** LabelEncoder

var\_mod **=** ['slug', 'symbol', 'name', 'date', 'ranknow', 'open', 'high', 'low',

'close', 'volume', 'market\_price', 'close\_ratio', 'spread'] le **=** LabelEncoder()

**for** i **in** var\_mod:

df[i] **=** le**.**fit\_transform(df[i])**.**astype(int)

df**.**head()

Spliting Train/Test:

#preprocessing, split test and dataset, split response variable X **=** df**.**drop(labels**=**'market\_price', axis**=**1)

#Response variable

y **=** df**.**loc[:,'market\_price']

#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.

**from** sklearn.model\_selection **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.3) print("Number of training dataset: ", len(X\_train))

print("Number of test dataset: ", len(X\_test))

print("Total number of dataset: ", len(X\_train)**+**len(X\_test))

plt**.**boxplot(df['open']) plt**.**show() plt**.**boxplot(df['close']) plt**.**show()

#### Module – 3

**Performance measurements of Logistic regression and Linear regression algorithms**

#import library packages

**import** pandas **as** p

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** s

**import** numpy **as** n

#Load given dataset

data **=** p**.**read\_csv("demo1.csv")

**import** warnings warnings**.**filterwarnings('ignore')

df**=**data**.**dropna() df**.**head() df**.**columns df**.**info()

**del** df['symbol']

**del** df['name']

**del** df['date']

**del** df['ranknow']

**del** df['close\_ratio']

**del** df['spread']

df**.**columns df["slug"]**.**unique()

**from** sklearn.preprocessing **import** LabelEncoder var\_mod **=** ['slug']

le **=** LabelEncoder()

**for** i **in** var\_mod:

df[i] **=** le**.**fit\_transform(df[i])**.**astype(int) df["slug"]**.**unique()

df**.**head()

df**.**tail()

**from** sklearn.metrics **import**

mean\_absolute\_error,mean\_squared\_error,r2\_score,explained\_variance\_score,median\_absolute

\_error

X **=** df**.**drop(labels**=**'market\_price', axis**=**1) #Response variable

y **=** df**.**loc[:,'market\_price']

#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.

**from** sklearn.model\_selection **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.3)

#### Logistic Regression :

**from** sklearn.linear\_model **import** LogisticRegression logR**=** LogisticRegression()

logR**.**fit(X\_train,y\_train) predictR **=** logR**.**predict(X\_test)

MAE**=** (mean\_absolute\_error(y\_test,predictR)) print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)

print(" ")

MSE**=**(mean\_squared\_error(y\_test,predictR)) print('MEAN SQUARED ERROR VALUE IS :',MSE)

print(" ")

MedianAE**=**(median\_absolute\_error(y\_test,predictR)) print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)

print(" ")

EVS**=**(explained\_variance\_score(y\_test,predictR)**\***100) print('ACCURACY RESULT OF LOGISTIC REGRESSION IS :',EVS)

print(" ")

R2**=**(r2\_score(y\_test,predictR)) print('R2\_SCORE VALUE IS :',R2)

print(" ")

#### LinearRegression

**from** sklearn.linear\_model **import** LinearRegression linR**=** LinearRegression()

linR**.**fit(X\_train,y\_train) predictL **=** linR**.**predict(X\_test)

MAE**=** (mean\_absolute\_error(y\_test,predictL)) print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)

print(" ")

MSE**=**(mean\_squared\_error(y\_test,predictL)) print('MEAN SQUARED ERROR VALUE IS :',MSE)

print(" ")

MedianAE**=**(median\_absolute\_error(y\_test,predictL)) print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)

print(" ")

EVS**=**(explained\_variance\_score(y\_test,predictL)**\***100) print('ACCURACY RESULT OF LINEAR REGRESSION IS :',EVS)

print(" ")

R2**=**(r2\_score(y\_test,predictL)) print('R2\_SCORE VALUE IS :',R2)

print(" ")

**from** sklearn.linear\_model **import** LinearRegression reg**=** LinearRegression()

reg**.**fit(X\_train, y\_train) print('Coefficients: \n', reg**.**coef\_)

print('Variance score: {}'**.**format(reg**.**score(X\_test, y\_test)))

plt**.**style**.**use('fivethirtyeight')

plt**.**scatter(reg**.**predict(X\_train), reg**.**predict(X\_train) **-** y\_train, color **=** "green", s **=** 10, label **=** 'Train data')

plt**.**scatter(reg**.**predict(X\_test), reg**.**predict(X\_test) **-** y\_test, color **=** "blue", s **=** 10, label **=** 'Test data')

plt**.**hlines(y **=** 0, xmin **=** 0, xmax **=** 50, linewidth **=** 2) plt**.**legend(loc **=** 'upper right')

plt**.**title("Residual errors") plt**.**show()

#### Module – 4

**Performance measurements of DecisionTreeRegressor and RandomForestRegressor algorithms**

#import library packages

**import** pandas **as** p

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** s

**import** numpy **as** n

#Load given dataset

data **=** p**.**read\_csv("demo1.csv")

**import** warnings warnings**.**filterwarnings('ignore')

df**=**data**.**dropna()

df**.**head() df**.**columns df**.**info()

**del** df['symbol']

**del** df['name']

**del** df['date']

**del** df['ranknow'] **del** df['close\_ratio'] **del** df['spread']

df**.**columns df["slug"]**.**unique()

**from** sklearn.preprocessing **import** LabelEncoder var\_mod **=** ['slug']

le **=** LabelEncoder()

**for** i **in** var\_mod:

df[i] **=** le**.**fit\_transform(df[i])**.**astype(int) df["slug"]**.**unique()

df**.**head()

df**.**tail()

**from** sklearn.metrics **import**

mean\_absolute\_error,mean\_squared\_error,r2\_score,explained\_variance\_score,median\_absolute

\_error

X **=** df**.**drop(labels**=**'market\_price', axis**=**1) #Response variable

y **=** df**.**loc[:,'market\_price']

#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.

**from** sklearn.model\_selection **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.3)

#### RandomForestRegressor

**from** sklearn.ensemble **import** RandomForestRegressor RF**=** RandomForestRegressor()

RF**.**fit(X\_train,y\_train) predictR **=** RF**.**predict(X\_test)

MAE**=** (mean\_absolute\_error(y\_test,predictR)) print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)

print(" ")

MSE**=**(mean\_squared\_error(y\_test,predictR)) print('MEAN SQUARED ERROR VALUE IS :',MSE)

print(" ")

MedianAE**=**(median\_absolute\_error(y\_test,predictR)) print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)

print(" ")

EVS**=**(explained\_variance\_score(y\_test,predictR)**\***100)

print('ACCURACY RESULT OF RANDOM FOREST REGRESSOR IS :',EVS)

print(" ")

R2**=**(r2\_score(y\_test,predictR)) print('R2\_SCORE VALUE IS :',R2)

print(" ")

#### DecisionTreeRegressor

**from** sklearn.tree **import** DecisionTreeRegressor DT**=** DecisionTreeRegressor() DT**.**fit(X\_train,y\_train)

predictD **=** DT**.**predict(X\_test)

MAE**=** (mean\_absolute\_error(y\_test,predictD)) print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)

print(" ")

MSE**=**(mean\_squared\_error(y\_test,predictD)) print('MEAN SQUARED ERROR VALUE IS :',MSE)

print(" ")

MedianAE**=**(median\_absolute\_error(y\_test,predictD)) print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)

print(" ")

EVS**=**(explained\_variance\_score(y\_test,predictD)**\***100)

print('ACCURACY RESULT OF DECISION TREE REGRESSOR IS :',EVS)

print(" ")

R2**=**(r2\_score(y\_test,predictD)) print('R2\_SCORE VALUE IS :',R2)

print(" ")

**import** joblib joblib**.**dump(RF,'rf.pkl')

**import** joblib joblib**.**dump(RF,'dt.pkl')

#### Module – 5

**Performance measurements of SVR and Lasso algorithms**

#import library packages

**import** pandas **as** p

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** s

**import** numpy **as** n

#Load given dataset

data **=** p**.**read\_csv("demo1.csv")

**import** warnings

warnings**.**filterwarnings('ignore') df**=**data**.**dropna()

df**.**head() df**.**columns df**.**info()

**del** df['symbol']

**del** df['name']

**del** df['date']

**del** df['ranknow'] **del** df['close\_ratio'] **del** df['spread']

df**.**columns df["slug"]**.**unique()

**from** sklearn.preprocessing **import** LabelEncoder var\_mod **=** ['slug']

le **=** LabelEncoder()

**for** i **in** var\_mod:

df[i] **=** le**.**fit\_transform(df[i])**.**astype(int)

df["slug"]**.**unique()

df**.**head()

df**.**tail()

**from** sklearn.metrics **import**

mean\_absolute\_error,mean\_squared\_error,r2\_score,explained\_variance\_score,median\_absolute

\_error

X **=** df**.**drop(labels**=**'market\_price', axis**=**1) #Response variable

y **=** df**.**loc[:,'market\_price']

#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.

**from** sklearn.model\_selection **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.3)

#### SVR

**from** sklearn.svm **import** SVR svmA**=** SVR() svmA**.**fit(X\_train,y\_train)

predictS **=** svmA**.**predict(X\_test)

MAE**=** (mean\_absolute\_error(y\_test,predictS)) print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)

print(" ")

MSE**=**(mean\_squared\_error(y\_test,predictS))

print('MEAN SQUARED ERROR VALUE IS :',MSE)

print(" ")

MedianAE**=**(median\_absolute\_error(y\_test,predictS)) print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)

print(" ")

EVS**=**(explained\_variance\_score(y\_test,predictS)**\***100) print('ACCURACY RESULT OF SVR IS :',EVS)

print(" ")

R2**=**(r2\_score(y\_test,predictS)) print('R2\_SCORE VALUE IS :',R2)

print(" ")

#### Lasso

**from** sklearn.linear\_model **import** Lasso la**=** Lasso()

la**.**fit(X\_train,y\_train) predictLo **=** la**.**predict(X\_test)

MAE**=** (mean\_absolute\_error(y\_test,predictLo)) print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)

print(" ")

MSE**=**(mean\_squared\_error(y\_test,predictLo)) print('MEAN SQUARED ERROR VALUE IS :',MSE)

print(" ")

MedianAE**=**(median\_absolute\_error(y\_test,predictLo)) print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)

print(" ")

EVS**=**(explained\_variance\_score(y\_test,predictLo)**\***100) print('ACCURACY RESULT OF LASSO REGRESSION IS :',EVS)

print(" ")

R2**=**(r2\_score(y\_test,predictLo)) print('R2\_SCORE VALUE IS :',R2)

print(" ")

#### HTML Code:

<!DOCTYPE html>

<html >

<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>

<meta charset="UTF-8">

<title>TITLE</title>

<link rel="stylesheet" href="{{ url\_for('static', filename='css/bootstrap.min.css') }}">

<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'>

<style>

.back{

background-image: url("{{ url\_for('static', filename='image/cry.gif') }}");

}

.white{ color:white;

}

.space{ margin:10px 30px; padding:15px 10px;

background: palegreen; width:500px

}

.gap{

padding:10px 20px;

}

</style>

</head>

<body class="back">

<div>

<div class="jumbotron">

<h1 style="text-align:center">CRYPTOCURRENCY MARKET PRICE PREDICTIONS

</h1>

</div>

<!-- Main Input For Receiving Query to our ML -->

<form class="form-group" action="{{ url\_for('predict')}}"method="post">

<div class="row">

<div class="gap col-md-6 ">

<label class="white" for="">SLUG</label>

<select class="nspace form-control" name="SLUG" id="SLUG">

<option value=0>binance-coin</option>

<option value=1>bitcoin</option>

<option value=2>cardano</option>

<option value=3>ethereum</option>

</select>

<label class="white" for="">OPEN</label>

<input type="number" class="space form-control" step="0.01" name="OPEN" placeholder="OPEN" required="required" /><br>

<label class="white" for="">HIGH</label>

<input type="number" class="space form-control" step="0.01" name="HIGH" placeholder="HIGH" required="required" /><br>

</div>

<div class="gap col-md-6">

<label class="white" for="">LOW</label>

<input type="number" class="space form-control" step="0.01" name="LOW" placeholder="LOW" required="required" /><br>

<label class="white" for="">CLOSE</label>

<input type="number" class="space form-control" step="0.01" name="CLOSE" placeholder="CLOSE" required="required" /><br>

<label class="white" for="">VOLUME</label>

<input type="number" class="space form-control" step="0.01" name="VOLUME" placeholder="VOLUME" required="required" /><br>

</div>

</div>

<div style="padding:2% 35%">

<button type="submit" class="btn btn-success btn-block" style="width:350px;padding:20px">Predict</button>

</div>

</form>

<br>

<br>

<div style="background:skyblue;padding:2% 40%">

{{ prediction\_text }}

</div>

</div>

</body>

</html>

#### Flask Deploy:

import numpy as np

from flask import Flask, request, jsonify, render\_template import pickle

import joblib

app = Flask( name ) model = joblib.load('rf.pkl') @app.route('/')

def home():

return render\_template('index.html') @app.route('/predict',methods=['POST']) def predict():

'''

For rendering results on HTML GUI '''

int\_features = [(x) for x in request.form.values()] final\_features = [np.array(int\_features)] print(final\_features)

prediction = model.predict(final\_features) output = prediction[0]

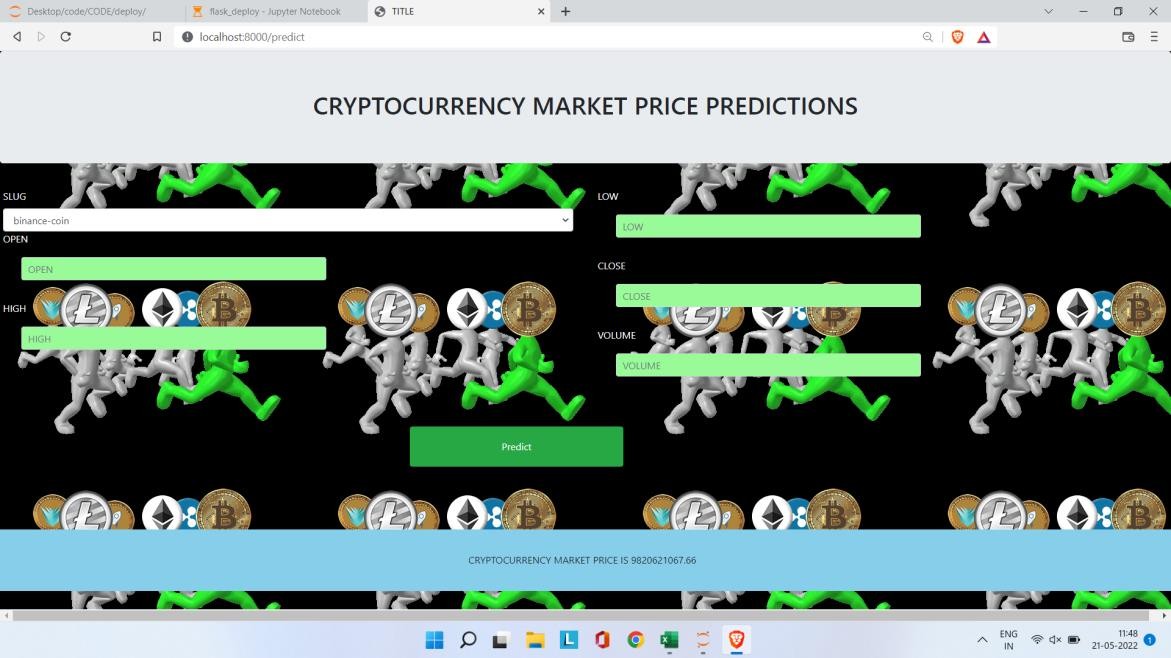
print(output)

return render\_template('index.html', prediction\_text='CRYPTOCURRENCY MARKET PRICE IS {}'.format(output))

if name == " main ": app.run(host="localhost", port=8000)

* 1. **SAMPLE SCREENS**





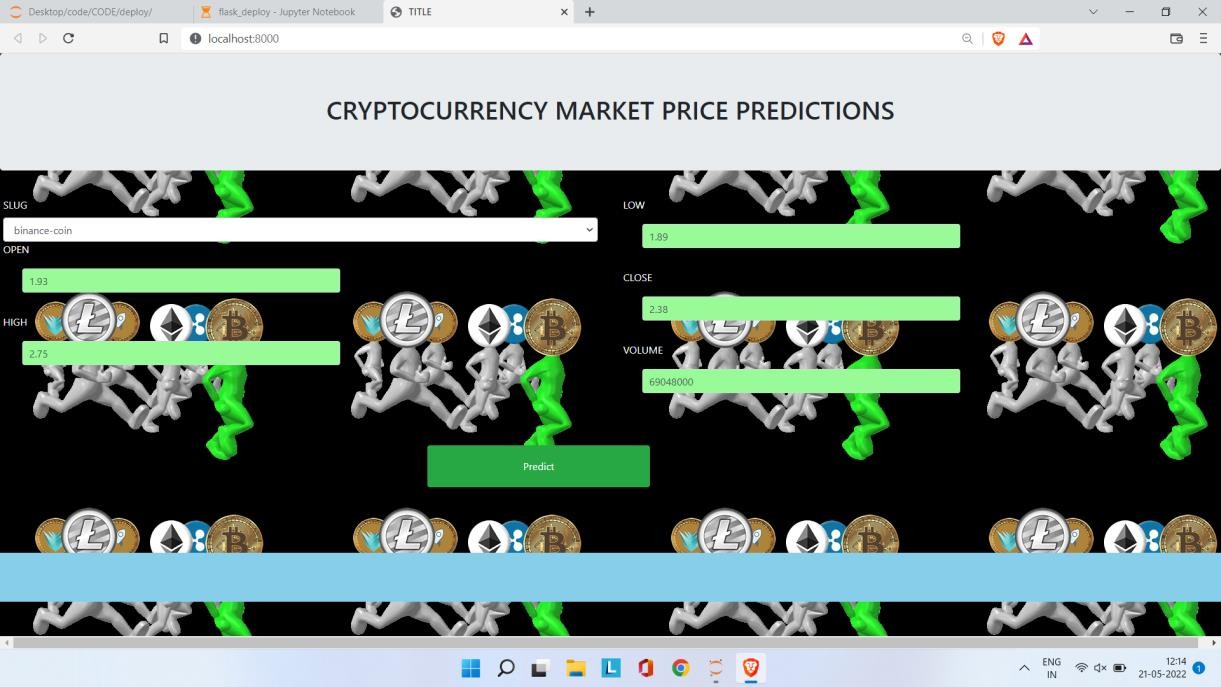


73



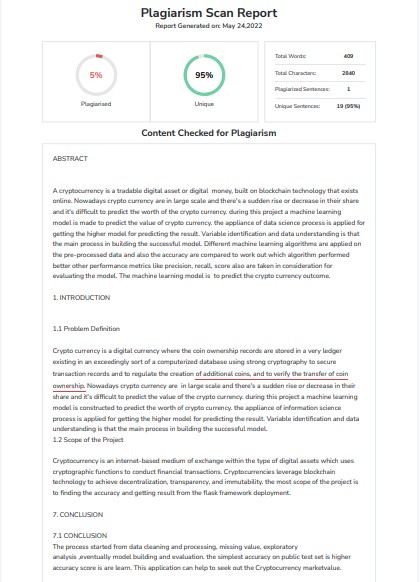








**PALGIARISM**



**REFERENCES**

1. Mehrnoosh Mirtaheri , Fred Morstatter “Identifying and Analyzing Cryptocurrency Manipulations in Social Media” in IEEE transactions on computational Social System,vol.8,No.3,June 2021
2. Lekkala Sreekanth Reddy, Dr.P. Sriramya“ A Research On Bitcoin Price Prediction Using Machine Learning Algorithms“ in International Journal of Scientific & Technology research volume 9, issue 04, april 2020
3. Neha Mangla “Bitcoin Price Prediction Using Machine Learning” in International Journal of Information and Computing science Volume 6, Issue 5, May 2019
4. Alireza Ashayer “Modeling and prediction of cryptocurrency prices using machine learning techniques**”** in Computer, vol. 49, no. 6, pp. 38–46, April, 2019
5. Franco Valencia, Alfonso Gómez-Espinosa and Benjamín Valdés-Aguirre “ Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning**”** in Proc. IEEE 13th Int. Conf. Data Mining, pp. 1103–1108 Year 2019
6. Ruchi Mittal, Shefali Arora and M.P.S Bhatia **“**Automated cryptocurrencies prices prediction using machine learning **”** in ICTACT journal on soft computing, volume: 08, issue: 04 **,** JULY 2018
7. O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini, “Online human-bot interactions: Detection, estimation, and characterization,” in *Proc. 11th Int. AAAI Conf. Web Social Media*, 2017
8. L. Wu, F. Morstatter, X. Hu, and H. Liu, “Mining misinformation in social media,” in *Big Data in Complex and Social Networks*. Boca Raton, FL, USA: CRC Press, 2016,
9. K. Starbird, J. Maddock, M. Orand, P. Achterman, and R. M. Mason, “Rumors, false flags, and digital vigilantes: Misinformation on Twitter after the 2013 Boston marathon bombing,” in *Proc. IConference*, 2014.
10. J. Ratkiewicz, M. Conover, M. R. Meiss, B. Gonçalves, A. Flammini, and F. Menczer, “Detecting and tracking political abuse in social media,” in *Proc. ICWSM*, vol. 11, 2011.
11. S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, “Prominent features of rumor propagation in online social media,” in *Proc. IEEE 13th Int.*

*Conf. Data Mining*, Dec. 2013.

1. V. S. Subrahmanian *et al.*, “The DARPA Twitter bot challenge,” *Computer*,vol. 49, no. 6, Jun. 2016.
2. D. M. J. Lazer *et al.,* “The science of fake news,” *Science*, vol. 359, no. 6380.
3. M. Forelle, P. Howard, A. Monroy-Hernández, and S. Savage, “Politicalbots and the manipulation of public opinion in venezuela,” 2015,

*arXiv:1507.07109*.

1. X. Hu, J. Tang, H. Gao, and H. Liu, “Social spammer detection with sentiment information,” in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014