TREND ANALYSIS OF DOGE, RIPPLE_XRP, ETHEREUM CRYPTO CURRENCIES BY REGRESSION MODEL

A PROJECT REPORT

Submitted by

SRIMATHI S (211420104262)

SRI VARSSHINI S (211420104270)

YOHAMALAR R (211420104314)

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 "**TREND ANALYSIS OF DOGE, RIPPLE_XRP, ETHEREUM CRYPTO CURRENCIES BY REGRESSION MODEL"** is the bonafide work of **SRIMATHI S (211420104262), SRI VARSSHINI S (211420104270), YOHAMALAR R (211420104314)** who carried out the project work under my supervision.

Signature of the HOD with date	**Signature of the Supervisor with date**
Dr.L.JABA SHEELA, M.E., Ph.D	**Dr.M.SANGEETHA, M.Tech., Ph.D**
PROFESSOR AND HEAD,	**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 Srimathi S (211420104262), Sri Varsshini S (211420104270), Yohamalar R (211420104314) hereby declare that this project report titled "Trend Analysis of Doge, Ripple_XRP, Ethereum Crypto Currencies by Regression Model" , under the guidance of Dr.M.Sangeetha M.Tech., Ph.D., is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

SRIMATHI S

SRI VARSSHINI S

YOHAMALAR R

ACKNOWLEDGEMENT

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

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

We would like to express our sincere thanks to **Project Coordinator Dr.Valarmathi, M.E., Ph.D.,** and **Project Guide Dr.M.Sangeetha, M.Tech., Ph.D.,** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

SRIMATHI S

SRI VARSSHINI S

YOHAMALAR R

ABSTRACT

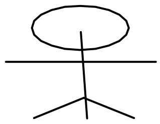
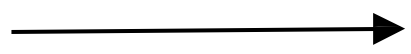
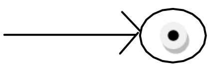
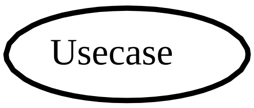
In recent years, crypto currencies have drawn a lot of interest and become increasingly popular and their value and market trends have become subjects of intense speculation and analysis. This study focuses on analysing the trends of the three prominent crypto currencies are Doge coin (DOGE), Ripple (XRP), and Ethereum (ETH). The objective is to predict and understand price fluctuations based on historical data. To achieve this, a comprehensive dataset consisting historical price data, transactions, price appreciation, and other relevant information features will be collected for the selected crypto currencies. The dataset will cover a significant timeframe, allowing for a thorough analysis of trends patterns. The gathered data will be split into training and testing sets in order to precisely assess the regression model's performance. Following that, the model will be trained using the historical data to determine how the independent variables and crypto currency prices relate to one another. Subsequently, the predictive capabilities will be assessed by relating the predicted prices with the actual prices obtained from the testing set.

Keywords— Trend analysis, Doge coin, Ripple (XRP), Ethereum (ETH), Crypto currency, Regression model.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

S.NO	NAME	NOTATION	DESCRIPTION
1.	Actor		It aggregates several classes into a single classes.
2.	Relation (extends)	extends 	Extends relationship is used when one use case is similar to another use case but does a bit more.
3.	Communication		Communication between various use cases.
4.	State		State of the process.
5.	Initial State		Initial state of the object
6.	Final state		Final state of the object
7.	Usecase		Interact ion between the system and external environment.

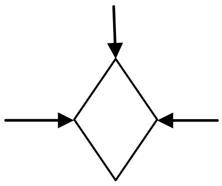
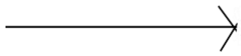
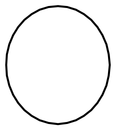
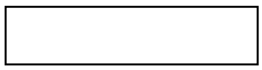
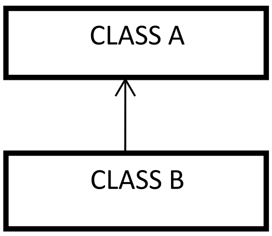
8.	Decision box		Represents decision making process from a constraint
9.	Message	Message 	Represents the message exchanged.
10	Data Process/ State		A circle in DFD represents a state or process which has been triggered due to some event or action.
11.	External entity		Represents external entities such as keyboard, sensors, etc.
12.	Aggregation		Interaction between the system and external environment

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

OVERVIEW OF THE PROJECT

The traditional financial system is characterized by a number of government rules, frameworks and policies, and central banks play a key role. In such a system, individuals must rely on a third party, known as the central bank. However, the traditional financial system is limited, leading to the development of digital currencies. Crypto currencies, as decentralized digital or virtual currencies, have attracted the attention. While thousands of digital currencies exist today, some of them have gained popularity, while others have not. The crypto currency market is dominated by a select group of coins, with the top 80% primarily covered by 10 prominent crypto currencies, such as Bitcoin, Ethereum, Tether, Binance Coin, USD Coin, and others. Due to key its characteristics such as decentralization, the digital market trend is rapidly expanding accessibility and market diversity facilitated by Blockchain technology. The prominent crypto currencies have completely transformed the financial landscape with their secure and decentralized digital alternatives to physical currency.

Although it may be difficult to pinpoint the exact reasons for price movements, machine learning algorithms have proven effective in providing accurate price predictions. In conclusion, the crypto currency market is characterized by a few dominant players, and predicting price fluctuations of these digital assets is a challenging yet vital task. The use of machine learning algorithms, in particular regression methods, is a promising approach to gain insights into the dynamic and volatile nature of crypto currency prices. Machine learning algorithms are used to predict the opening price, closing price, and market capitalization of crypto currencies for the following day, based on previous historical data. The goal is to improve decision-making while providing a comprehensive view of our study without obscuring its potential shortcomings. This research analyzes three prominent currencies: Doge coin (Doge), Ripple_XRP (XRP), and Ethereum (ETH), employing Bayesian Ridge, Huber Regression, and Elastic Net models. A comprehensive dataset consisting of historical price data, trading volumes, market capitalization, and other relevant information features will be collected for the selected crypto currencies. The dataset will cover a significant timeframe, allowing for a thorough analysis of trends and patterns.

DESCRIPTION

Machine learning predicts the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data, the basics of machine learning, the implementation of a simple python machine learning algorithm. The training and prediction process involves the use of specialized algorithms. It feeds the training data to an algorithm as follows and the algorithm uses this training data to predict a new test dataset. Machine learning can be roughly divided into three categories. There are supervised learning, unsupervised learning, and reinforcement learning. The supervised learning program is given both input data and the corresponding labeling to learn data must be labeled beforehand by a human being. Unsupervised learning has no labels. It provided information to the learning algorithm. This algorithm must determine the clustering of the input data. Finally, Reinforcement learning interacts dynamically with its environment and it receives positive or negative feedback to improve its performance. Supervised machine learning learns patterns and relationships between input and output data. It is defined on the basis of its use of labeled data. Labeled data is a dataset that contains numerous examples of features and targets. Supervised learning uses algorithms that learn the relationships between features and targets from the dataset. This process is referred to as training and fitting.

Regression is a type of supervised machine learning in which algorithms learn continuous values from the data .A regression model is able to show whether changes observed in the dependent variable are associated with changes in one or more explanatory variables. This is done by essentially fitting a best-fit line and observing how the data are dispersed around this line. Regression helps economists and financial analysts, from asset valuation to forecasts. There are several main reasons to use regression analysis such as predicting future economic conditions, trends, or values ,determining the relationship between two or more variables and to understand how a variable changes when another changes.

PROBLEM DEFINITION

Crypto currencies have emerged as a popular investment option, and understanding their trends is crucial for traders and investors. Crypto currencies have gained significant attention and popularity in recent years, and their value and market trends have become the subject of intense speculation and analysis. Bitcoin, the world's first crypto currency, began with a mission to challenge the existing global financial order. Alternative coins other than Bitcoin with power-packed features that can be used in decentralized finance do not have a standard prediction model. Most existing studies focus mainly on Bitcoin and Ethereum, while other alternatives such as Doge and Ripple are not considered. Analysis of crypto currencies using regression models to gain insight into their price movements and make informed predictions. Regression analysis, a statistical modeling technique, can provide valuable insights into the factors that influence the crypto-currency prices and enable the development of predictive models. The proposed system aims to enhance the understanding of crypto currency trends and provide valuable insights for traders, investors, and researchers. It can contribute to the development of more accurate price prediction models and assist in crypto currency market risk management strategy.

OBJECTIVE

The scope of a trend analysis project using a regression model for DOGE, XRP, and ETH includes collection, preprocessing, and feature selection. The aim of this project is to predict future price trends and assess risk, and analyze the impact of external factors on these crypto currencies. It provides decision-support to traders and investors, with a focus on ongoing model refinement and education of its capabilities and limitations.

The dynamic and speculative nature of crypto currency markets should be considered when defining the scope of the project. The analytical process started with data cleaning and processing, missing values, exploratory analysis, and model building and evaluation. The algorithm will be evaluated using model evaluation techniques such as R-squared (R2). R2 measures the goodness of fit that determines how the data fits the regression model.

The best accuracy on a public test set will be determined by a higher accuracy score. The aim is to automate this process by showing the prediction result in a web application or a desktop application.

This application can be used to find the DOGE-RIPPLE XRP-ETHEREUM prediction by connecting with the AI model and optimizing the implementation in an artificial intelligence environment.

CHAPTER 2

LITERATURE SURVEY

INTRODUCTION

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 discusses published information in a particular subject area and sometimes information in a particular subject area within a certain time. Its 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 Novel LSTM based Approach for Crypto Currency Price Prediction

Authors : R. Vijaya Saraswathi, Swathi Bindhu Bollina, Rachana Bongu, Ankitha Cherukuru, Nikitha Chintala

Year 2023

The decentralized nature of crypto currencies makes it difficult to predict their pricing, making reliable price forecasts essential for investors. A proposed LSTM-based approach will create machine learning models using open-source libraries and cross-validation. The aim is to create a machine learning algorithm that can forecast Bitcoin values using regression and deep learning algorithms. The end result will be an effective and accurate model with R2 score of 0.96 for train data and 0.97 test data for forecasting crypto currency values.

Unraveling the Landscape of Bitcoin Research Using Machine Learning: A Topic Modelling Approach

Authors : Sarveshwar Kumar Inani, Gaurav Kabra, Gaurav Nagpal, Peterson Owusu

Year 2023

The study analyzes the evolution of Bitcoin research using bibliometric analysis and structural topic modeling techniques on 1,937 articles from the Scopus database. Six thematic clusters are identified, including Blockchain-based digital currency, volatility modeling, portfolio diversification, Bitcoin futures trading, return forecasting, and crypto currency regulations. The study outlines future research directions in finance, economics, and management and aims to support effective risk management strategies, regulatory frameworks, and business approaches. It serves as a valuable resource for industry practitioners, academics, regulators, and policymakers navigating the evolving landscape of crypto currencies.

A Two-phase Approach to Determine User-Preference and Feature Importance in Pricing of Cryptocurrencies using Twitter Data

Authors : Saptashwa Maity, Soujatya Khan, Sobhan Sarkar

Year 2023

The study proposes a novel two-phase robust hybrid approach for determining user-preference and feature importance of cryptocurrencies. The approach is based on subjectivity and polarity score obtained from the sentiment polarity classification of the tweets. The proposed approach outperformed conventional machine learning techniques with accuracy, precision, recall, F-1 score, and AUC-ROC values of 92%, 88%, 85%, 95%, and 94%.

Statistical Analysis to Reveal Factors Affecting Cryptocurrency Pricesin Commodity Futures Trading and Stock Market

Authors : Adhi Dharma Wibawa, M Sadewa Wicaksana, Yuri Pamungkas

Year 2023

Crypto currency, a virtual currency with a market capitalization of 18 billion USD in 2017, is widely used worldwide. However, it carries high risk due to fast price changes. This study uses Pearson Correlation to examine the association of crypto currencies with gold, oil, natural gas, and the Nasdaq stock market. The Nasdaq stock market dataset has the highest correlation, with Ethereum having the highest value compared to crude oil.

Prediction Of Crypto currencies Prices Using Long Short Term Memory and Technical Indicators

Authors : Omar Al–Qudah, Hamzah Zureigat, Amer Al–Badarneh, Saleh Abu–Soud

Year 2023

Bitcoin, the most popular digital currency, is the most dominant crypto currency in trading volume, accounting for 0.47 of the total market volume. This study focuses on predicting prices of Ethereum and Litecoin using Long-short-term memory. The study explores the impact of technical indicators and Bitcoin's correlation with altcoins. Results show a significant improvement in price prediction when using technical indicators and correlation with Bitcoin. The error decreases, particularly in Lite coin, where MAPE decreases by as much as 0.41.

Price Prediction of Digital Currencies using Machine Learning

Authors : Ashutosh Dhar Dwivedi, Subhayu Dutta, Subhrangshu Adhikary, Jens Myrup Pedersen

Year 2023

Crypto currencies have gained significant popularity, making it challenging for users to choose the right one. Accurately predicting future prices is crucial for profitable investments in digital currencies, but it presents unique challenges as it lacks physical goods or services. Machine learning is a pivotal tool for addressing this issue. This research analyzes five prominent currencies, Monero, Bitcoin, Ethereum, IOTA, and Zcash, using five models: SVR, LRG, Huber, RANSAC, MLP, and AdaBoost. The experimental results show promising results, with an impressive R2 score of 1.0 for specific machine learning algorithms. This advancement opens new opportunities for informed decision-making and profitable ventures in the dynamic world of digital currencies.

A Comparative Performance Evaluation of Bitcoin Price Prediction Using Machine Learning Techniques

Authors : Forhad Uddin Ahmed, Mamun Ahmed, Fahamida Hossain Mahi, Syed Hasnut Abdullah, Sayma Alam Suha

Year 2023

The rising value of bitcoin necessitates accurate forecasts for investment decisions due to its extreme volatility. Factors such as adoption, regulatory developments, geopolitical events, and macroeconomic factors influence its price. However, limited studies have focused on machine learning techniques in this domain. This study aims to analyze multiple machine learning regression models to identify the most effective system for estimating Bitcoin values. The dataset was meticulously analyzed and preprocessed, and the top findings were a 99.5 percent R-squared, 0.01281 RMSE, and 0.005755 MAE using the gradient boosting regressor model.

Next Day Bitcoin Price Prediction: Performance Comparison of Various Statistical and Machine Learning Algorithms

Authors : Lubna Noor AhmadChitkara, Pradeepta Kumar Sarangi, Merry Saxena, Ashok Kumar Sahoo, Amandeep Singh

Year 2023

This research paper analyzes five statistical and machine learning algorithms for predicting the next day Bitcoin price using daily traded data from November 2021 to February 2023. The LSTM and Lasso regression models predict the same value with an accuracy of 97.88%, followed by Random Forest and Support Vector models with 94.65% and 94.40% respectively. The Ridge regression model has the highest accuracy at 98.02%, making it the most accurate option for predicting the next day Bitcoin price.

Cryptocurrency Trading Using Machine Learning

Authors : Mayank Puri, Aman Garg, Lekha Rani

Year 2023

Crypto currency education is crucial for informed foreign investment decisions. The price of crypto currency has increased exponentially due to easy trading, especially in developing countries. Financial institutions have been adding crypto currency to their portfolios, leading to economic growth. Traders can forecast price swings due to social media's impact on crypto currency prices. Advanced machine learning techniques can help traders capitalize on the Bitcoin market's inefficiency for high profits. Basic computational processes can predict the near-term development of the Bitcoin market, and SVM and Random Forest algorithms can predict and compare prices.

Impact of Machine Learning on Digital Pricing And Crypto currency Markets

Authors : Arun Kumar Marandi, Gordhan Jethava

Year 2023

Machine learning (ML) is a forecasting technique used in various sectors, including manufacturing, advertising, finance, travel, and transportation. It has significantly impacted digital pricing strategies and crypto currencies. ML allows companies to trade money over virtual channels more easily and at a lower cost. Researchers have used surveys and probability sampling to explore the effectiveness of ML in digital pricing and crypto currencies. This research provides insight into the effects of ML methods on digital pricing and Bitcoin trading.

CHAPTER 3
SYSTEM ANALYSIS

EXISTING SYSTEM

Bitcoin the world's first crypto currency started off with the mission to challenge the existing global financial order. Bayesian Regression, a binary classification algorithm, was used to predict price variation in Bitcoin and the prediction gave almost 200% returns in less than 60 days when used with a trading strategy. The objective is to understand how features of Bitcoin (such as transaction volume, cost per transaction) can affect the next day change in price level of Bitcoin through the use of an Artificial Neural Network (ANN). Additionally, alternative methodologies like time-scale multifractal methods have been deployed to analyze Bitcoin's volume and price patterns, offering valuable insights into its market behavior.

DISADVANTAGES

- Alternative coins other than Bitcoin with power packed features that can be used in the decentralized finance does not have a standard prediction model.
- Long-term Bitcoin price predictions are done at an accuracy of upto 55% using machine learning models.
- Most of the existing works focuses mainly on Bitcoin and Ethereum while other alternatives such as Doge, Ripple are not taken into account.

PROPOSED SYSTEM

System for analysing crypto currencies using regression models to gain insights into their price movements and make informed predictions. Crypto currencies have emerged as a popular investment option, and understanding their trends is crucial for traders and investors. Regression analysis, a statistical modelling technique, can provide valuable insights into the factors influencing crypto currency prices and enable the development of predictive models. The proposed system aims to enhance the understanding of crypto currency trends and provide valuable insights for traders, investors, and researchers. It can contribute to the development of more accurate price prediction models and assist in risk management strategies in the crypto currency market. Thus the proposed work is to provide users to look into other alternative coins and find out which ones besides Bitcoin are doing well in the market by finding trends and predictions using Machine learning Algorithms. System for analyzing crypto currencies using regression models is to gain insights into their price movements and make informed predictions. Regression analysis, a statistical modeling technique, can provide valuable insights into the factors influencing crypto currency prices and enable the development of predictive models. The proposed system aims to enhance the understanding of crypto currency trends and provide valuable insights for traders, investors, and researchers. It can contribute to the development of more accurate price prediction models and assist in risk management strategies in the crypto currency market.

MERITS

- Our suggested method will forecast bitcoin alternatives, generating greater returns for investors.
- Regression is a supervised machine learning technique which is used to predict continuous values. Thus the ultimate goal of the regression algorithm is to plot a best-fit line or a curve between the data.
- The proposed regression algorithms allows a useful mechanism to deal with insufficient data, or poor distributed data.
- The deployment procedure was put into effect by using Frontend codes like Html, Css, Bootstrap and Python Framework like Django or Flask.

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 Machine Learning Regression techniques 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 CLASSIFICATION MODEL

A high accuracy 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 the continuous variables.
- It produces an out-of-bag estimate problem which has proven to be unbiased in many tests and it is relatively easy to tune with.

CONSTRUCTION OF A PREDICTIVE MODEL

Machine learning needs data gathering have lot of past data's. Data gathering have sufficient historical data and raw data. Before data pre-processing, raw data can't be used directly. It's used to pre-process then, what kind of algorithm with model. Training and testing this model working and predicting correctly with minimum errors. Tuned model involved by tuned time to time with improving the accuracy.

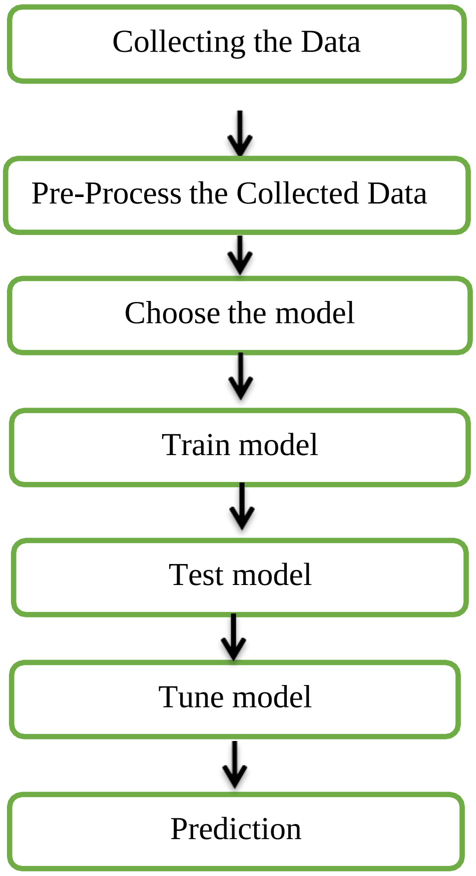


Fig 3.3.5.1 Process of dataflow diagram

SYSTEM REQUIREMENTS

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements

3. Environment requirements
    a. Hardware requirements
    b. software requirements

FUNCTIONAL REQUIREMENTS

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

NON FUNCTIONAL REQUIREMENTS

Process of functional steps,

1. Problem definition
2. Preparing dataset
3. Evaluating the Algorithm
4. Prediction the result

ENVIRONMENTAL REQUIREMENTS

1. Software Requirements:

Operating System	: Windows 7 or above
Tool	: Anaconda with Jupyter Notebook

2. Hardware requirements:

Processor	: Pentium IV/III
Hard disk	: minimum 80 GB
RAM	: minimum 2 GB

PLATFORM DESCRIPTION

ANACONDA

Anaconda is a free and open-source 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. Package versions are managed by the package management system “Conda”. The Anaconda distribution is used by over 12 million users and includes more 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 packages can be individually installed from the Anaconda repository with the conda install command or using the pip install command that is installed with Anaconda. Pip packages provide many of the features of conda 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. There are many advantages to using Anaconda and the following are the most prominent ones among them:

- Anaconda is free and open-source. This means we can use it without spending any money.
- In the data science sector, Anaconda is an industry staple. It is open-source too, which has made it widely popular.

- If we want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects we to have this skill. It is a must-have for data science.

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, 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.

Installation: The easiest way to install the *Jupyter Notebook App* is installing a scientific python distribution which also includes scientific python packages. The most common distribution is called **Anaconda.**

Running the Jupyter Notebook

Launching *Jupyter Notebook App:* The Jupyter Notebook App can be launched by clicking on the *Jupyter Notebook* icon installed by Anaconda in the start menu (Windows) or by typing in a terminal (*cmd* on Windows): “jupyter notebook”. This will launch a new browser window (or a new tab) showing the Notebook Dashboard, a sort of control panel that allows (among other things) to select which notebook to open. When started, the Jupyter Notebook App can access only files within its start-up folder (including any sub-folder).

No configuration is necessary if you place your notebooks in your home folder or subfolders. Otherwise, you need to choose a Jupyter Notebook App start-up folder which will contain all the notebooks.

Save notebooks: Modifications to the notebooks are automatically saved every few minutes. To avoid modifying the original notebook, make a copy of the notebook document (menu file -> make a copy…) and save the modifications on the copy.

Executing a notebook: Download the notebook you want to execute and put it in your notebook folder (or a sub-folder of it).

- ❖ Launch the jupyter notebook app
- ❖ In the Notebook Dashboard navigate to find the notebook: clicking on its name will open it in a new browser tab.
- ❖ Click on the menu *Help -> User Interface Tour* for an overview of the Jupyter Notebook App user interface.
- ❖ You can run the notebook document step-by-step (one cell a time) by pressing *shift + enter*.
- ❖ You can run the whole notebook in a single step by clicking on the menu *Cell -> Run All*.
- ❖ To restart the kernel (i.e. the computational engine), click on the menu *Kernel -> Restart*. This can be useful to start over a computation from scratch (e.g. variables are deleted, open files are closed, etc…).

File Extension: An **IPYNB** file is a notebook document created by Jupyter Notebook, an interactive computational environment that helps scientists manipulate and analyze data using Python.

CHAPTER 4

SYSTEM DESIGN

ER DIAGRAM

ER diagram illustrates the logical structure of a database by defining entities, their attributes, and their relationships.

ENTITY:

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent. The entities used here are model selection and trained model.

ATTRIBUTES:

The attributes are represented in the form of ellipse. Every attribute is connected to its entity. The attributes represented here are tuning model and supervised machine learning.

The purpose of ER Diagram is to represent the entity framework infrastructure. An entity can be place, person, object, event or a concept, which stores data in the database.

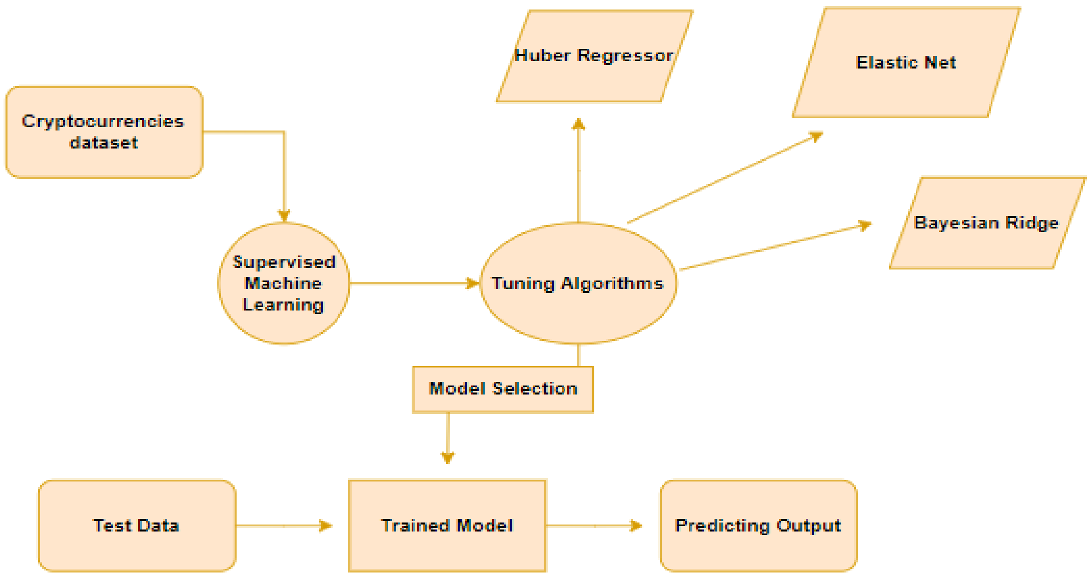


Fig 4.1 ER diagram for Trend Analysis of crypto currencies

USECASE DIAGRAM

A use case represents a particular functionality of a system. Use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors. Use case diagrams are considered for high level requirement analysis of a system.

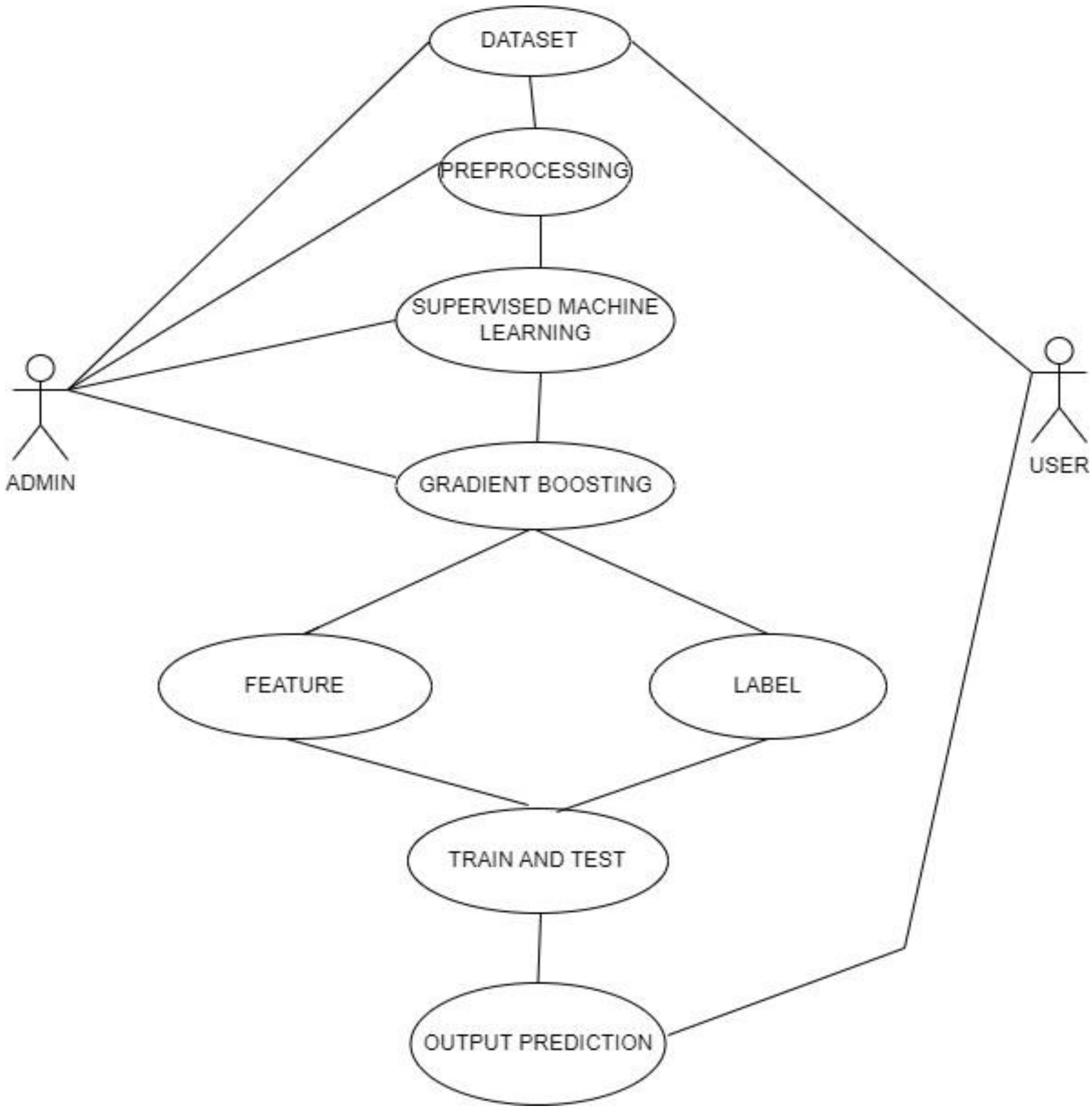


Fig 4.2 Usecase diagram for Trend Analysis of crypto currencies

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.

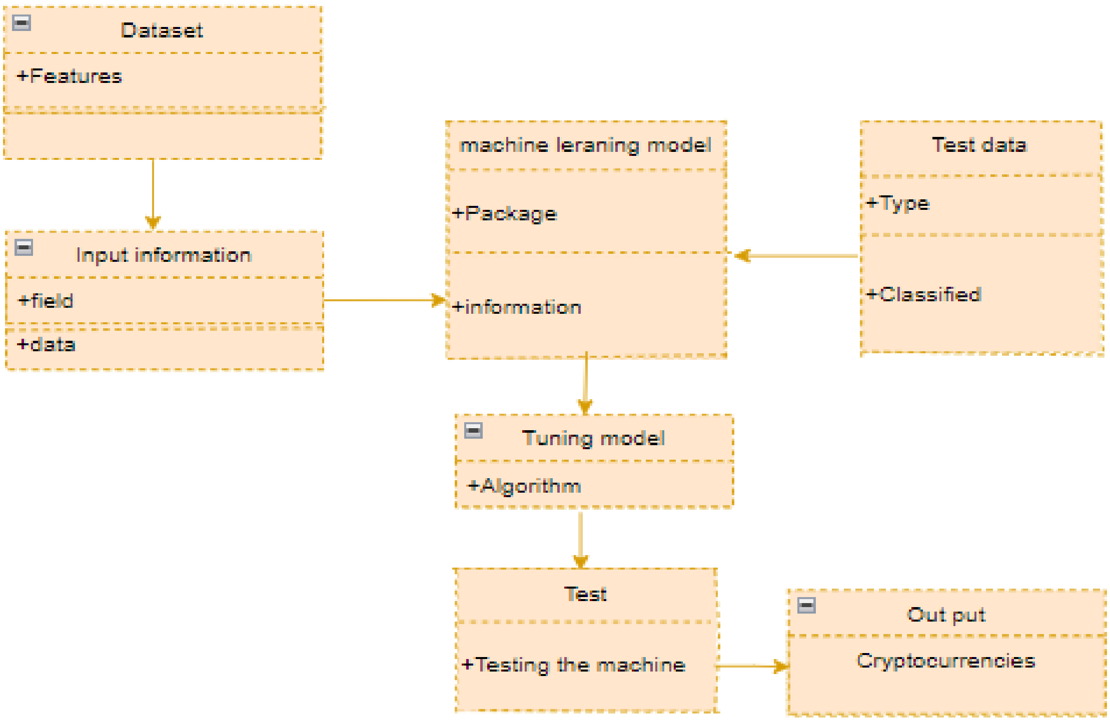


Fig 4.3 Class diagram for Trend Analysis of crypto currencies

ACTIVITY DIAGRAM

Activity diagram represents the flow of activities from one to another. The activity can be described as an operation of the system. An activity diagram captures the dynamic behavior of the system. It is also known as an object-oriented flowchart. Activity diagrams include swim lanes, branching, parallel flow, control nodes, expansion nodes, and object nodes.

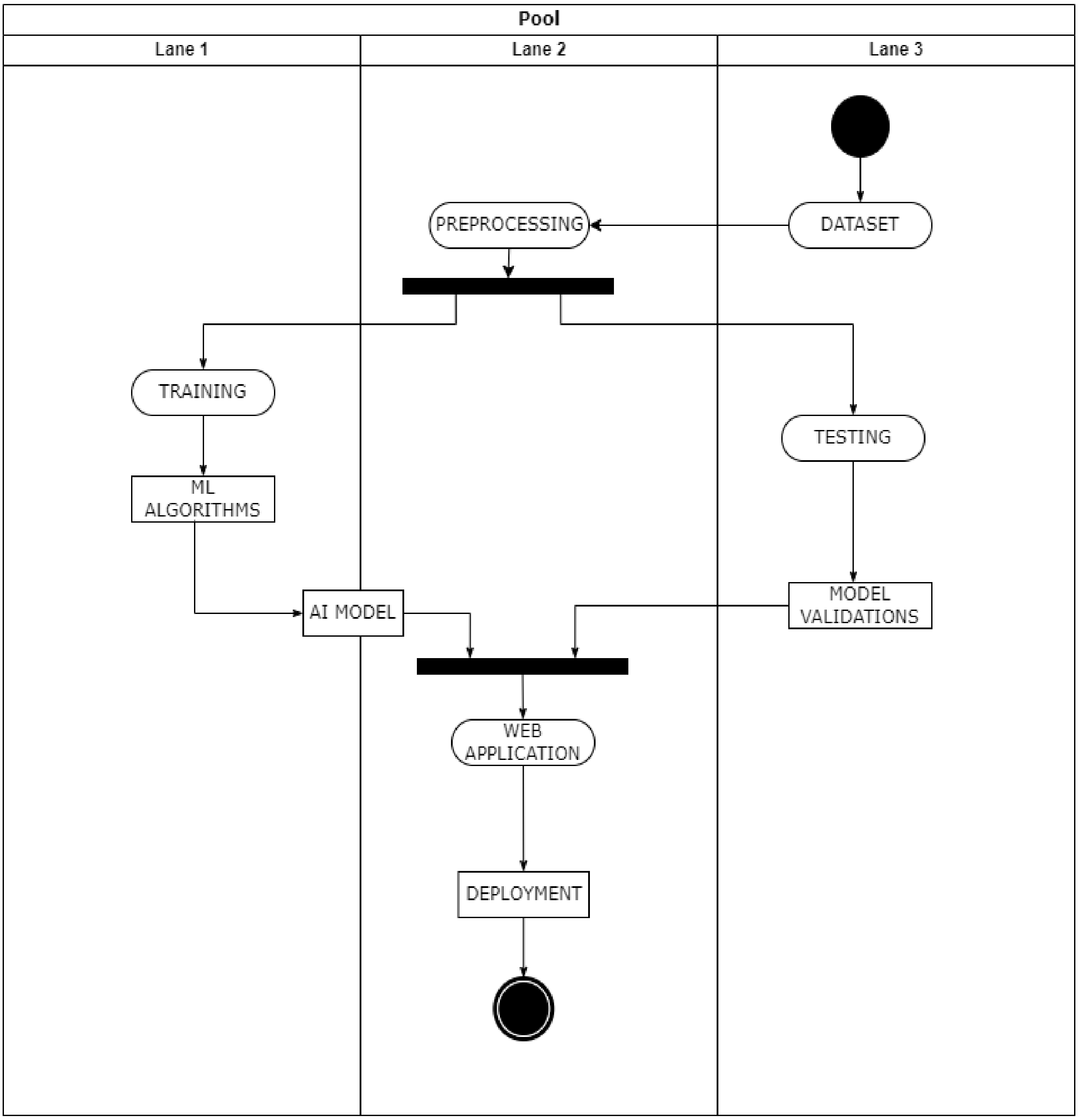


Fig 4.4 Activity diagram for Trend Analysis of crypto currencies

SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another. The sequence diagram includes a group of objects which are represented by lifelines, and the messages they exchange over time during their interaction.

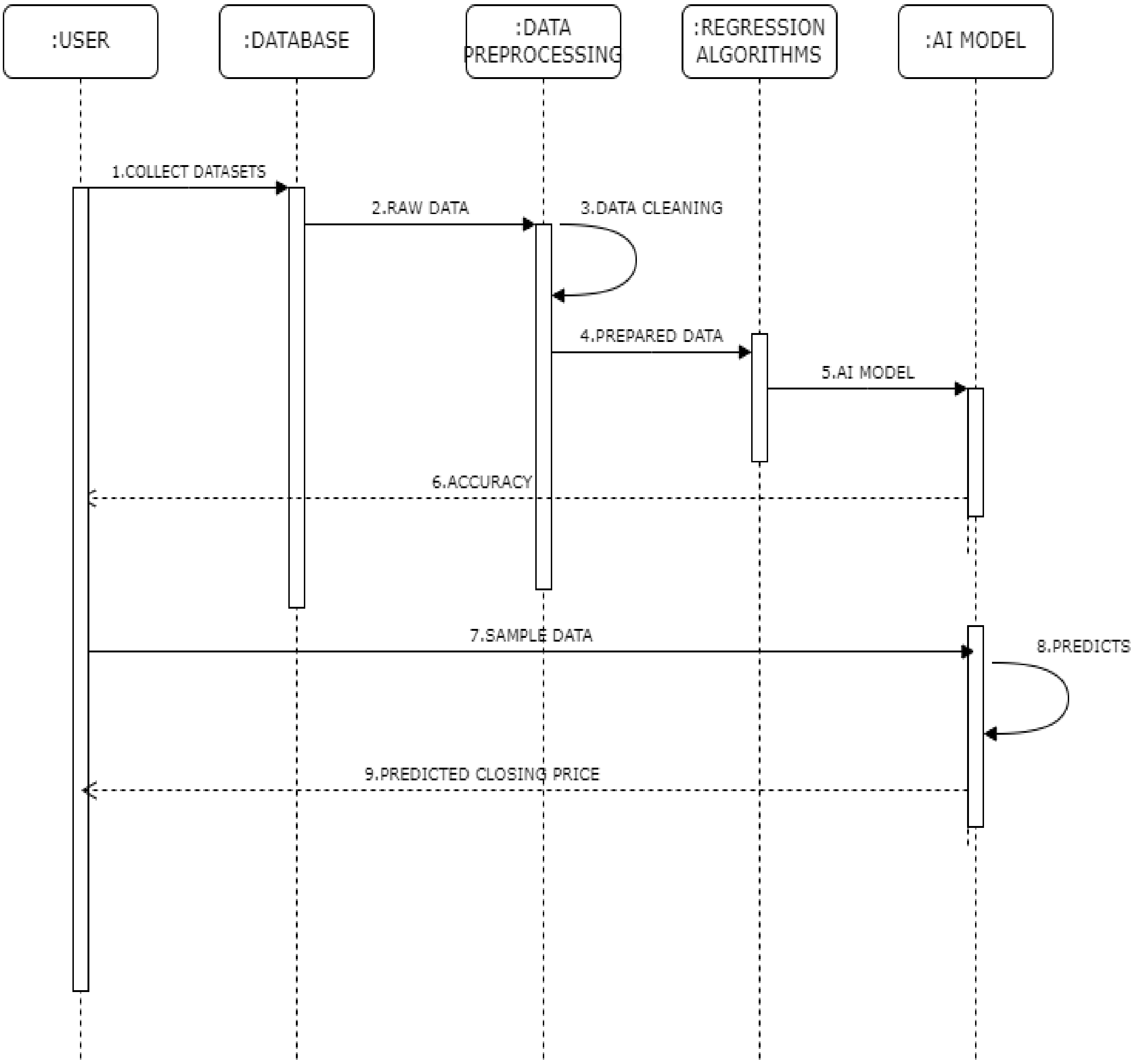


Fig 4.5 Sequence diagram for Trend Analysis of crypto currencies

DATA FLOW DIAGRAM

A data flow diagram shows how information flows through a system or process. These include data inputs and outputs, data stores, and the various processes through which the data moves. DFDs use standard symbols and notations to describe the relationships between entities. The four components of data flow diagrams are

- External entity
- Process
- Data store
- Data flow.

LEVEL - 0:

This is basically a contextual diagram, also referred to as a "context diagram". It only represents the top level or the 0 Level in the whole process. It gives an abstraction kind of view and shows the whole system as a single process and its relationship to externalities. Thus the system-level 0 data flow diagram is intended to be a high-level overview of the system. Here a user gives the input variables which are used to predict the closing price by testing them with the developed AI model. The predicted results will be returned to the user by the UI.

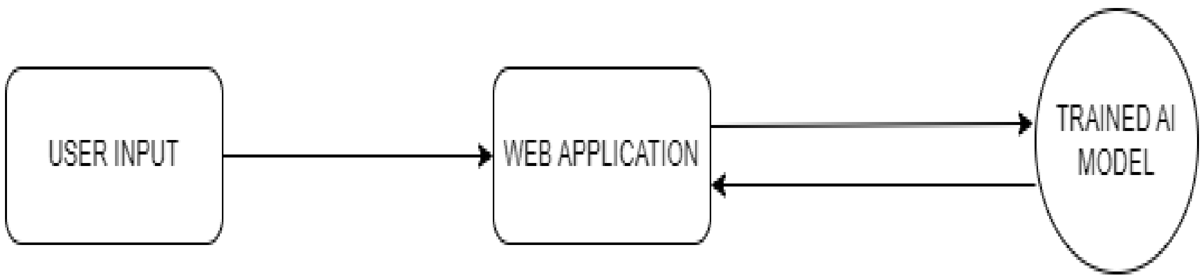


Fig.4.6.1 Level – 0 DFD Diagram for Trend analysis of Digital currencies

LEVEL – 1:

A level 1 DFD notates each of the main sub-processes that together form the complete system.

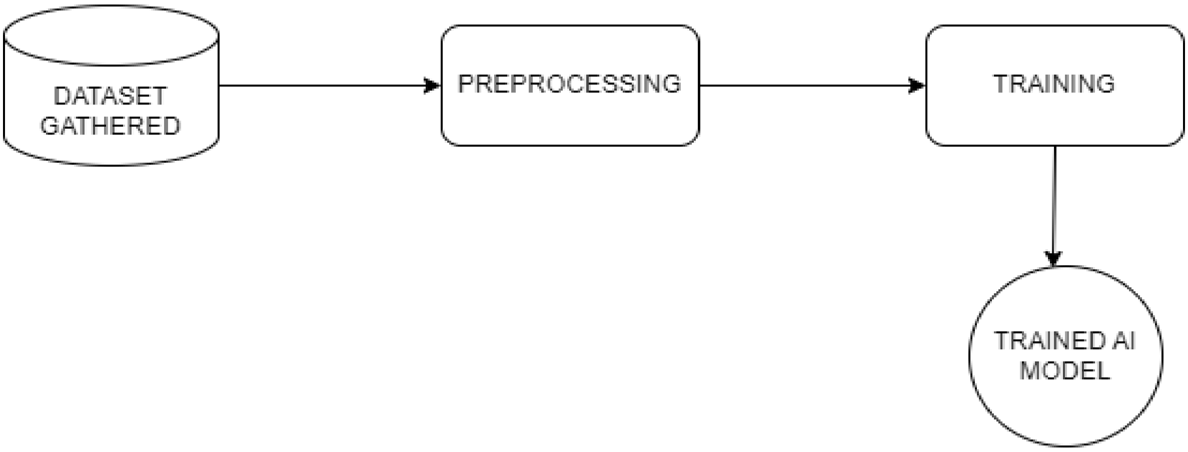


Fig.4.6.2 Level – 1 DFD Diagram for Trend analysis of Digital currencies

LEVEL - 2

This level provides an even more detailed view of the system by breaking down the sub-processes identified in the level 1 DFD into further sub-processes. Each sub-process is depicted as a separate process on the level 2 DFD. The data flows and data stores associated with each sub-process are also shown.

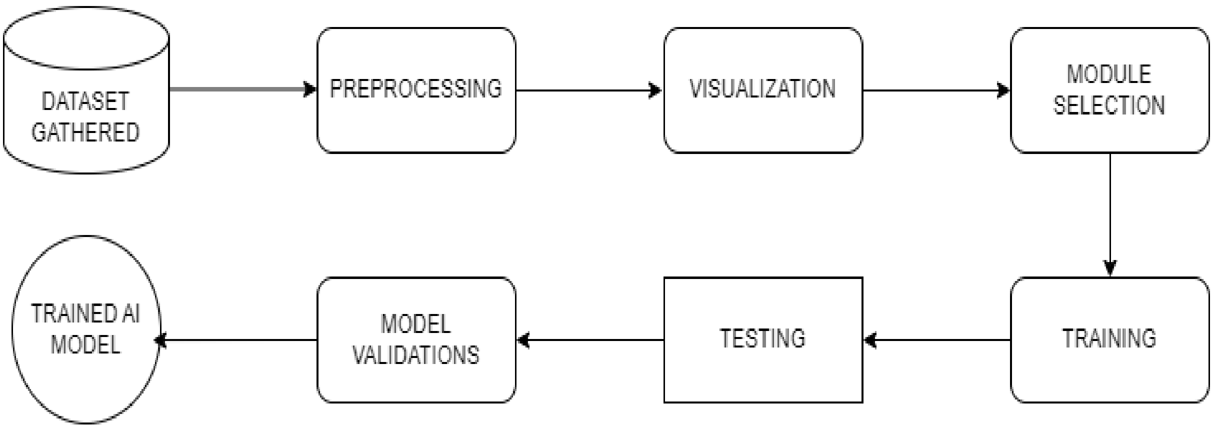


Fig.4.6.3 Level – 2 DFD Diagram for Trend analysis of Digital currencies

LEVEL – 3

This is the most detailed level of DFDs, which provides a detailed view of the processes, data flows, and data stores in the system. This level is typically used for complex systems, where a high level of detail is required to understand the system.

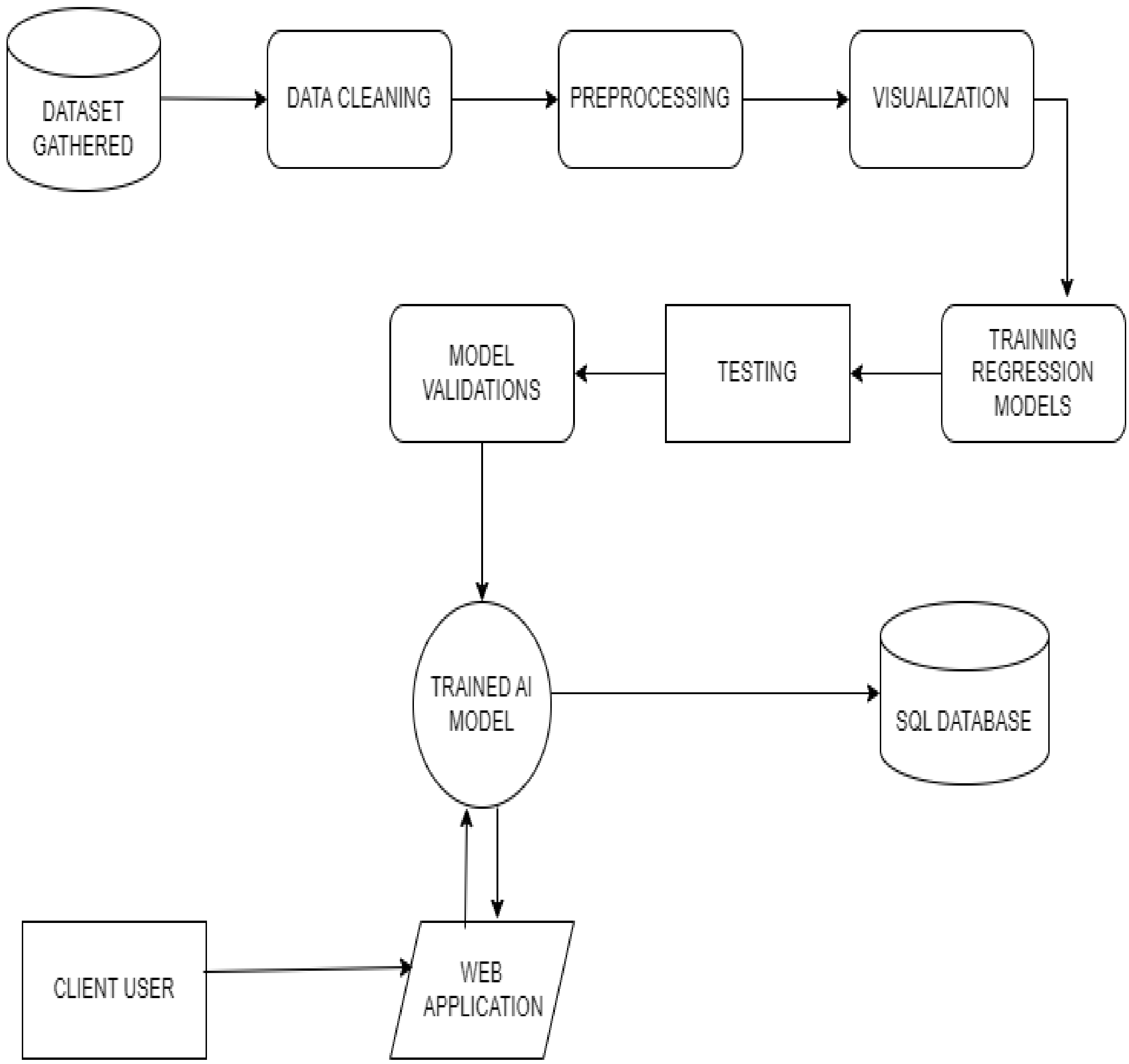


Fig.4.6.4 Level – 3 DFD Diagram for Trend analysis of Digital currencies

CHAPTER 5

SYSTEM ARCHITECTURE

SYSTEM OVERVIEW

The system architecture is the basic structure of the system. The diagram represents the relations between each of them and involves a sequence of decision-making processes and steps. The datasets of three different digital currencies are collected, preprocessed and trained to predict the closing price. The trained AI models are deployed as a web application.

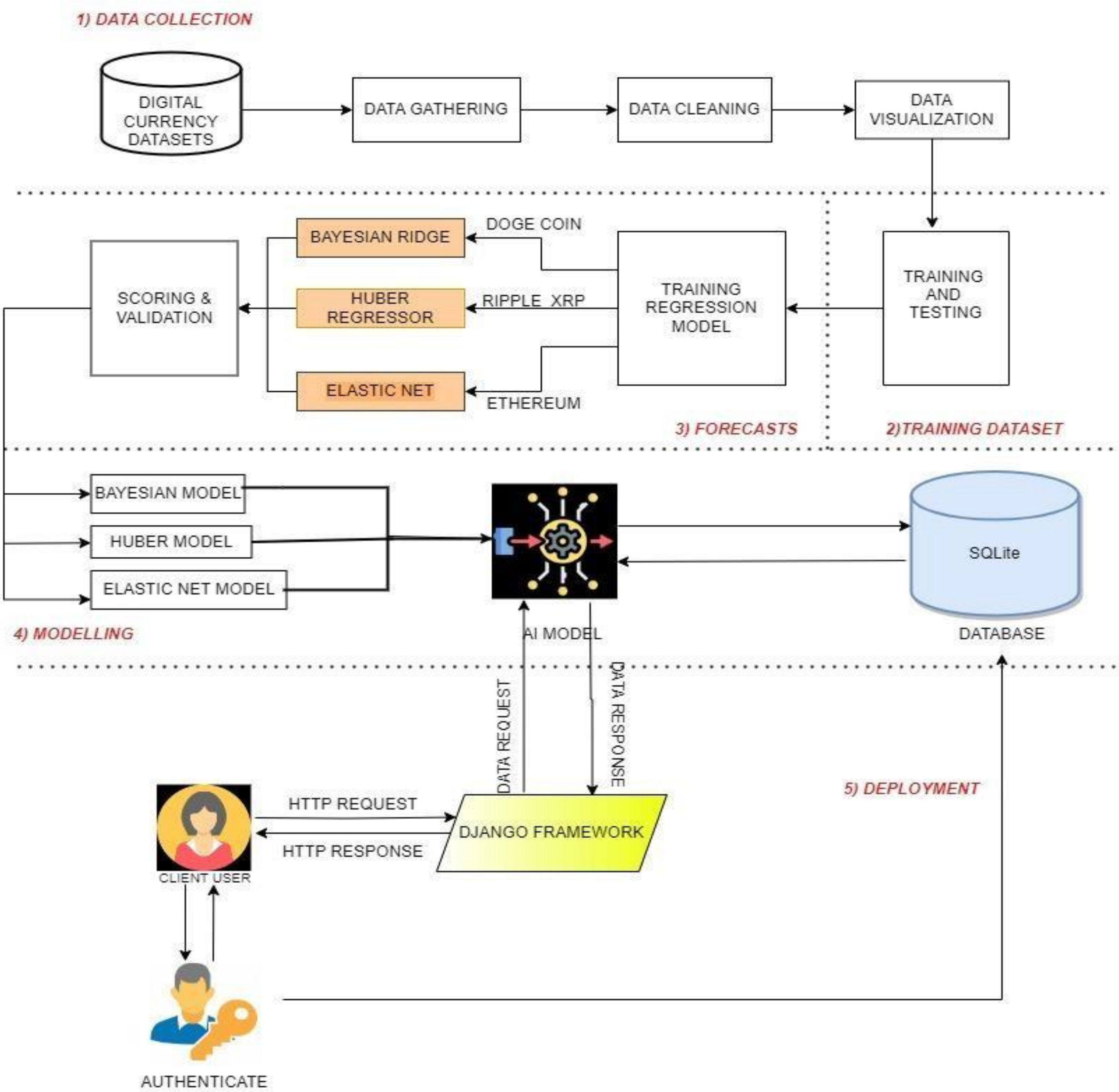


Fig.5.1 Architecture Diagram for Trend analysis of Digital currencies

MODULE DESCRIPTION

The following are the list of modules implemented in prediction of closing price for the digital currencies

- Data Preprocessing
- Data Visualization
- Huber Regressor model
- Bayesian Ridge model
- Elastic Net model
- Deployment

DATA PREPROCESSING:

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.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers uses this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties. This knowledge will help you choose which algorithm to use to build your model. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Data Validation/ Cleaning/Preparing Process

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 models 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 analyse 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.

	Date	Open	High	Low	Close	Adj Close	Volume
0	2017-11-09	0.001207	0.001415	0.001181	0.001415	0.001415	6259550.0
1	2017-11-10	0.001421	0.001431	0.001125	0.001163	0.001163	4246520.0
2	2017-11-11	0.001146	0.001257	0.001141	0.001201	0.001201	2231080.0
3	2017-11-12	0.001189	0.001210	0.001002	0.001038	0.001038	3288960.0
4	2017-11-13	0.001046	0.001212	0.001019	0.001211	0.001211	2481270.0

Table 5.1.1: Sample dataset

	Open	High	Low	Close	Volume
count	1760.000000	1760.000000	1760.000000	1760.000000	1.760000e+03
mean	0.059575	0.063096	0.056126	0.059619	1.016258e+09
std	0.101325	0.109152	0.093695	0.101379	3.563999e+09
min	0.001046	0.001210	0.001002	0.001038	1.431720e+06
25%	0.002550	0.002616	0.002500	0.002548	2.307671e+07
50%	0.003476	0.003603	0.003356	0.003495	8.981855e+07
75%	0.070633	0.075035	0.068478	0.070657	6.565853e+08
max	0.687801	0.737567	0.608168	0.684777	6.941068e+10

Table 5.1.2: Descriptive statistics of the sample dataset

DATA 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.

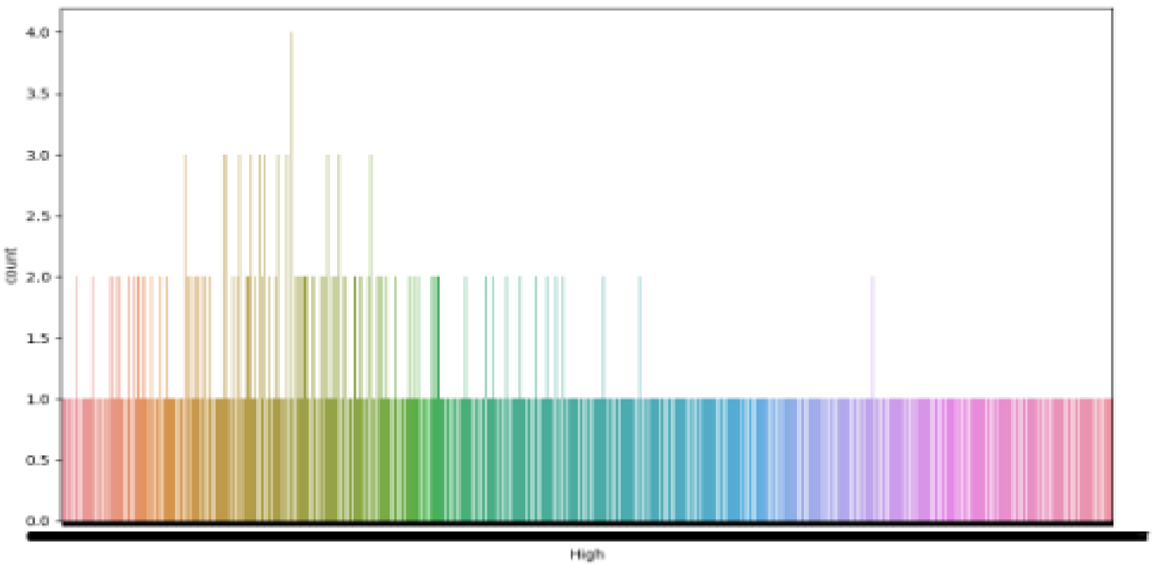


Fig 5.2.2 Visual representation of Data Preprocessing

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in given dataset.

HUBER REGRESSOR:

The Huber Regressor is a robust linear regression algorithm used for modelling and analysing data, particularly when there are potential outliers or errors in the dataset. It is designed to be less sensitive to extreme values, making it more robust compared to traditional least squares linear regression. Unlike ordinary least squares (OLS) regression, which minimizes the sum of squared residuals, the Huber Regressor minimizes a different cost function.

In Huber regression, the cost function is a combination of a quadratic term for small errors and a linear term for large errors. This combination allows the algorithm to provide a compromise between the robustness of the model and its efficiency. Essentially, it can handle outliers while still providing meaningful insights into the data's underlying linear relationship.

The Huber Regressor introduces a tuning parameter called the "delta" that determines the threshold at which the loss function transitions from the quadratic to linear part. Smaller delta values make the model more robust, while larger delta values allow the algorithm to behave more like OLS regression.

Selecting an appropriate delta is an important aspect of using the Huber Regressor effectively, as it balances the trade-off between robustness and efficiency.

In summary, the Huber Regressor is a valuable tool for regression analysis when dealing with noisy data or data containing outliers. It strikes a balance between the robustness of methods like the Least Absolute Deviations (LAD) regression and the efficiency of OLS regression, making it a useful choice in situations where traditional linear regression models may not perform well due to the presence of extreme values.

$$\text{Huber Loss}(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{for } |y - \hat{y}| \leq \epsilon \\ \epsilon(|y - \hat{y}| - \frac{1}{2}\epsilon) & \text{otherwise} \end{cases}$$

MODULE DIAGRAM

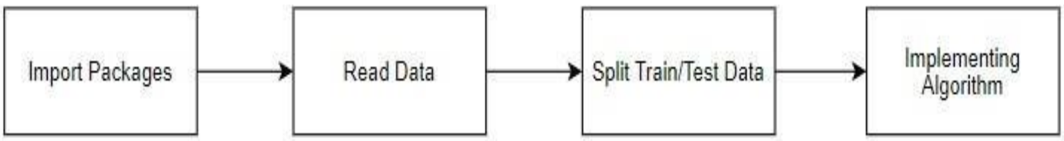


Fig 5.2.3 Huber Regressor Module

BAYESIAN RIDGE:

The Bayesian Ridge regression is a probabilistic and Bayesian approach to linear regression that combines the advantages of ridge regression with a probabilistic framework. This regression method aims to provide more robust and stable estimates of the regression coefficients, particularly when multicollinearity is present, and can handle cases where the number of features exceeds the number of data points.

In Bayesian Ridge regression, a prior distribution is assigned to the regression coefficients, typically a Gaussian (normal) distribution. This prior represents our initial beliefs about the coefficients' values, and it is updated based on the observed data through Bayes' theorem to form a posterior distribution. This posterior distribution provides a range of possible coefficient values along with their associated uncertainties.

One of the key advantages of Bayesian Ridge regression is that it naturally handles multicollinearity by shrinking the regression coefficients towards zero. The degree of shrinkage is determined by a hyper parameter, alpha, which controls the amount of regularization applied. A smaller alpha results in less regularization, while a larger alpha results in stronger regularization.

Bayesian Ridge also provides estimates of the predictive uncertainty, making it useful in scenarios where you not only want point estimates of the coefficients but also a measure of their reliability. This is particularly beneficial when making predictions, as it allows you to quantify the uncertainty associated with your predictions.

In summary, Bayesian Ridge regression is a Bayesian approach to linear regression that addresses multicollinearity and provides a probabilistic framework for estimating regression coefficients and their associated uncertainties. It's a powerful tool in situations where traditional linear regression models might struggle due to multicollinearity or when a measure of predictive uncertainty is crucial.

$$p(w \mid \lambda) = \mathcal{N}(w \mid 0, \lambda^{-1} I_p)$$

ELASTIC NET:

Linear regression refers to a model that assumes a linear relationship between input variables and the target variable. With a single input variable, this relationship is a line, and with higher dimensions, this relationship can be thought of as a hyper plane that connects the input variables to the target variable. The coefficients of the model are found via an optimization process that seeks to minimize the sum squared error between the predictions (*yhat*) and the expected target values (*y*).

loss = sum i=0 to n (y_i – yhat_i)^2

A problem with linear regression is that estimated coefficients of the model can become large, making the model sensitive to inputs and possibly unstable. This is particularly true for problems with few observations (*samples*) or more samples (*n*) than input predictors (*p*) or variables (so-called *p >> n problems*).

One approach to addressing the stability of regression models is to change the loss function to include additional costs for a model that has large coefficients. Linear regression models that use these modified loss functions during training are referred to collectively as penalized linear regression.

One popular penalty is to penalize a model based on the sum of the squared coefficient values. This is called an L2 penalty. An L2 penalty minimizes the size of all coefficients, although it prevents any coefficients from being removed from the model.

l2_penalty = sum j=0 to p beta_j^2

Another popular penalty is to penalize a model based on the sum of the absolute coefficient values. This is called the L1 penalty. An L1 penalty minimizes the size of all coefficients and allows some coefficients to be minimized to the value zero, which removes the predictor from the model.

l1_penalty = sum j=0 to p abs(beta_j)

Elastic net is a penalized linear regression model that includes both the L1 and L2 penalties during training. Using the terminology from "*The Elements of Statistical Learning*," a hyper parameter "*alpha*" is provided to assign how much weight is given to each of the L1 and L2 penalties. Alpha is a value between 0 and 1 and is used to weight the contribution of the L1 penalty and one minus the alpha value is used to weight the L2 penalty.

*elastic_net_penalty = (alpha * l1_penalty) + ((1 – alpha) * l2_penalty)*

For example, an alpha of 0.5 would provide a 50 percent contribution of each penalty to the loss function. An alpha value of 0 gives all weight to the L2 penalty and a value of 1 gives all weight to the L1 penalty. The benefit is that elastic net allows a balance of both penalties, which can result in better performance than a model with either one or the other penalty on some problems.

Another hyper parameter is provided called "lambda" that controls the weighting of the sum of both penalties to the loss function. A default value of 1.0 is used to use the fully weighted penalty; a value of 0 excludes the penalty. Very small values of lambda, such as 1e-3 or smaller, are common.

*elastic_net_loss = loss + (lambda * elastic_net_penalty)*

DEPLOYMENT – WEB APPLICATION

DJANGO(WEB FRAMEWORK):

Django is an extremely popular and fully featured server-side web framework, written in Python. This module shows you why Django is one of the most popular web server frameworks, how to set up a development environment, and how to start using it to create your own web applications. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development.

Start Using an API

1. Most APIs require an API key.

2. The easiest way to start using an API is by finding an HTTP client online, like REST-Client, Postman, or Paw.

3. The next best way to pull data from an API is by building a URL from existing API documentation.

The flask object implements a WSGI application and acts as the central object. It is passed the name of the module or package of the application. Once it is created it will act as a central registry for the view functions, the URL rules, template configuration and much more.

The name of the package is used to resolve resources from inside the package or the folder the module is contained in depending on if the package parameter resolves to an actual python package (a folder with an_init_.py file inside) or a standard module (just a .py file).

Parameters

rule (*str*) – The URL rule string.

endpoint (Optional[str]) – The endpoint name to associate with the rule and view function. Used when routing and building URLs. Defaults to view_func._name_.

view_func (Optional[Callable]) – The view function to associate with the endpoint name.

provide_automatic_options (Optional[bool]) – Add the OPTIONS method and respond to OPTIONS requests automatically.

options (Any) – Extra options passed to the Rule object.

Return type -- None

After_Request(f)

Register a function to run after each request to this object.

The function is called with the response object, and must return a response object. This allows the functions to modify or replace the response before it is sent. If a function raises an exception, any remaining after request functions will not be called. Therefore, this should not be used for actions that must execute, such as to close resources. Use *teardown_request()* for that.

Parameters:

f (Callable[[Response], Response])

Return type

Callable[[Response], Response]

after_request_funcs: t.Dict[AppOrBlueprintKey,

t.List[AfterRequestCallable]]

A data structure of functions to call at the end of each request, in the format {scope: [functions]}. The scope key is the name of a blueprint the functions are active for, or None for all requests.

To register a function, use the *after_request()* decorator.

This data structure is internal. It should not be modified directly and its format may change at any time.

app_context()

Create an AppContext. Use as with block to push the context, which will make current_app point at this application. An application context is automatically pushed by RequestContext.push() when handling a request, and when running a CLI command. Use this to manually create a context outside of these situations.

With app.app_context():

Init_db()

HTML:

HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

Fig 5.2.6.2 Basic construction of a HTML page

CSS

CSS stands for Cascading Style Sheets. It is the language for describing the presentation of Web pages, including colours, layout, and fonts, thus making our web pages presentable to the users. CSS is designed to make style sheets for the web. It is independent of HTML and can be used with any XML-based markup language. Now let’s try to break the acronym:

Cascading: Falling of Styles

Style: Adding designs/Styling our HTML tags

Sheets: Writing our style in different documents

CSS Syntax

```
Selector {
Property 1: value;
Property 2: value;
Property 3: value;
}
```

MODEL VALIDATIONS:

MEAN ABSOLUTE ERROR (MAE):

Mean Absolute Error (MAE) is the most popular criterion for evaluating the ML model's performance, particularly in regression cases. It is less sensitive to outliers, making it a better metric for evaluating the best model for a given problem. MAE computes the average absolute values difference between the model predicted values and the actual values where each set of differences has equal weight. The MAE evaluates the model performance with a single value ranging from 0 to infinity. This single value of the MAE portrays the overall performance of the algorithm. When the output shows fewer values, the model has better performance or goodness of learning, indicating that the model's forecast values are closer to the actual values. This behaviour can be measured mathematically, expressed below:

$$MAE=Pn\ j=1|yj-\ \hat{}\ Yj|\ N$$

where N is the number of coins, yj is the cryptocurrency Actual value and ˆ Yj is the Forecast value.

ROOT MEAN SQUARED ERROR (RMSE):

RMSE is a commonly used evaluation technique to measure forecasting quality, particularly in regression cases. The H. Shamshad et al.: Forecasting and trading of the Stable Crypto currencies formula calculates the average deviation between the forecast and actual values by taking the square root of the mean of the squared errors. RMSE outputs a single value that exhibits the total error of the ML model. Due to its scale dependent feature, it is mainly used to assess the quality of different ML algorithms or to check the quality of a single algorithm over different datasets using the derived equation, as expressed below:

RMSE=sPn j=1(yj− ˆ Yj)2 N

where N is the number of coins, yj is the crypto currency's Actual value and ˆ Yj is the Forecast value. A model with a lower RMSE value represents better performance, indicating that the predicted values of the crypto currency are closer to the actual values.

MEAN ABSOLUTE PERCENTAGE ERROR (MAPE):

MAPE is a valuable criterion to quantify the degree of error between the crypto currency coin's actual value and the forecast value, expressed as a percentage. It is always positive; lower MAPE means better predictive model performance. The average percentage difference is computed as expressed below:

MAPE= 100% N n X j=1 |yj− ˆ Yj| |yj|

where N is the number of coins, yj is the cryptocurrency Actual value and ˆ Yj is the Forecast value.

R-SQUARED (R2):

R-squared, or the coefficient of determination is a statistical technique to evaluate a regression algorithm's quality. R2 measures the goodness of fit that determines how the data fits the regression model. It has a scale between 0 and

1, representing none or all of the variances in the target variable is explained by the input variable(s). The coefficient of determination can be mathematically expressed as given below: $R2= \bar{}\sigma 2\ \sigma 2$ where ¯ σ2 represents the Explained variation, which is the sum of squared differences between the forecast values and the responsevariablevalues,whileσ2 denotes the Total variation and is the sum of squared differences between the crypto original values and its corresponding mean value.

MEAN SQUARED ERROR (MSE)

Mean Squared Error, or MSE for short, is a popular error metric for regression problems. It is also an important loss function for algorithms fit or optimized using the least squares framing of a regression problem. Here "*least squares*" refers to minimizing the mean squared error between predictions and expected values. The MSE is calculated as the mean or average of the squared differences between predicted and expected target values in a dataset.

MSE = 1 / N * sum for i to N (y_i – yhat_i)^2

Where *y_i* is the i'th expected value in the dataset and *yhat_i* is the i'th predicted value. The difference between these two values is squared, which has the effect of removing the sign, resulting in a positive error value. The squaring also has the effect of inflating or magnifying large errors. That is, the larger the difference between the predicted and expected values, the larger the resulting squared positive error.

This has the effect of "*punishing*" models more for larger errors when MSE is used as a loss function. It also has the effect of "*punishing*" models by inflating the average error score when used as a metric. We can create a plot to get a feeling for how the change in prediction error impacts the squared error.
The units of the MSE are squared units.

5.2.6 EXPLAINED VARIANCE:

$$r^2 = R^2 = \eta$$

In ANOVA, explained variance is calculated with the "eta-squared (η^2)" ratio Sum of Squares$(SS)_{between}$ to SS_{total}; It's the proportion of variances for between group differences. R^2 in regression has a similar interpretation: what proportion of variance in Y can be explained by X (Warner, 2013).

Variance describes how much a random variable differs from its expected value. The variance is defined as the average of the squares of the differences between the individual (observed) and the expected value. This means that it is always positive.

The explained variance is used to measure the proportion of the variability of the predictions of a machine learning model. Simply put, it is the difference between the expected value and the predicted value. It is a very important concept to understand how much information we can lose by reconciling the dataset.

CHAPTER 6

SYSTEM IMPLEMENTATION

SERVER SIDE

Data Validation and Pre-Processing Technique:

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")
data=pd.read_csv('DOGE.csv')
data.head()
data.columns
del data['Date']
del data['Adj Close']
data.columns
data.info()
data.isnull()
data.isnull().sum()
df=data.dropna()
df.isnull().sum()
df.describe()
df.corr()
pd.crosstab(df["High"], df["Low"])
df.duplicated()
sum(df.duplicated())
df=df.drop_duplicates()
```

Exploration Data Analysis of Visualization and Training a Model By Given Attributes

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

data=pd.read_csv('DOGE.csv')
data.head()
data.columns
del data['Date']
del data['Adj Close']
data.columns
plt.figure(figsize=(12,7))
sns.countplot(x='High',data=data)
plt.figure(figsize=(20,10))
plt.subplot(1,2,1)
plt.hist(data['Open'],color='blue')
plt.subplot(1,2,2)
plt.hist(data['Close'],color='green')
data.hist(figsize=(15,60),layout=(15,4),color='green')
plt.show()
plt.boxplot(data['Open'])
sns.pairplot(data)
fig,ax=plt.subplots(figsize=(20,15))
sns.heatmap(data.corr(),annot=True,fmt='0.2%',cmap='autumn',ax=ax)
```

Performance Measurements of Huber Regressor

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

data=pd.read_csv('DOGE.csv')
data.head()
data.columns
data.isnull()
df=data.isnull().sum()
df
df=data.dropna()
df.isnull().sum()
df.head()

x = df.drop(labels='Close', axis=1)
y = df.loc[:,'Close']

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=4
2)
print("NUMBER OF TRAIN DATASET :",len(x_train))
print("NUMBER OF TEST DATASET :",len(x_test))
print("TOTAL NUMBER OF DATASET :",len(x_train)+len(x_test))

from sklearn.linear_model import HuberRegressor
from sklearn.linear_model import HuberRegressor
```

```python
HR=HuberRegressor()
HR.fit(x_train,y_train)
predicted=HR.predict(x_test)

from sklearn.metrics import r2_score
R2=r2_score(y_test,predicted)
print("THE R2 SCORE OF HuberRegressor :",R2)

from sklearn.metrics import rand_score
RAND=rand_score(y_test,predicted)
print("THE RAND_SCORE OF HuberRegressor:",RAND)

from sklearn.metrics import mean_squared_error
MSE=mean_squared_error(y_test,predicted)
print("THE MEAN_SQUARED_ERROR OF HuberRegressor:",MSE)

from sklearn.metrics import median_absolute_error
MAE=median_absolute_error(y_test,predicted)
print("THE MEDIAN_ABSOLUTE_ERROR :",MAE)

df2=pd.DataFrame()
df2["y_test"]=y_test
df2["predicted"]=predicted
df2.reset_index(inplace=True)
plt.figure(figsize=(15,7))
plt.plot(df2["predicted"][:100],marker='x',linestyle='dashed',color='red')
plt.plot(df2["y_test"][:100],marker='o',linestyle='dashed',color='green')
plt.show()
```

```
import joblib
joblib.dump(HR, 'HR.pkl')
```

Performance Measurement of ElasticNet

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

data=pd.read_csv('XRP.csv')
data.head()
data.columns
del data['Date']
del data['Adj Close']
data.columns
data.isnull()
data.isnull().sum()
df=data.dropna()
df.head()
df.duplicated()
sum(df.duplicated())

x = df.drop(labels='Close',axis=1)
y = df.loc[:,'Close']

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=4
2)
```

```
print("NUMBER OF TRAIN DATASET:",len(x_train))
print("NUMBER OF TEST DATASET:",len(x_test))
print("TOTAL NUMBER OF DATASET :",len(x_train)+len(x_test))

from sklearn.linear_model import ElasticNet
GR=ElasticNet()
GR.fit(x_train,y_train)
predicted=GR.predict(x_test)

from sklearn.metrics import r2_score
R2 = r2_score(y_test,predicted)
print("THE R2 SCORE OF ElasticNet IS :",R2)

from sklearn.metrics import rand_score
RAND = rand_score(y_test,predicted)
print("THE ACCURACY SCORE OF ElasticNet IS :",RAND*100)

from sklearn.metrics import explained_variance_score
EVS = explained_variance_score(y_test,predicted)
print("THE EXPLAINED VARIENCE SCORE OF ElasticNet IS :",EVS)

from sklearn.metrics import mean_squared_error
MSE = mean_squared_error(y_test,predicted)
print("THE MEAN SQUARED ERROR SCORE OF ElasticNet IS :",MSE)

from sklearn.metrics import median_absolute_error
MAE = median_absolute_error(y_test,predicted)
print("THE MEAN ABSOLUTE ERROR SCORE OF ElasticNet IS :",MAE)
```

```
df2=pd.DataFrame()
df2["y_test"]=y_test
df2["predicted"]=predicted
df2.reset_index(inplace=True)
plt.figure(figsize=(20,5))
plt.plot(df2["predicted"][:100], marker='x', linestyle='dashed', color='red')
plt.plot(df2["y_test"][:100], marker='o', linestyle='dashed', color='green')
plt.show()

import joblib
joblib.dump(GR, 'GR.pkl')
```

Performance Measurement of BayesianRidge

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

data=pd.read_csv('ETHEREUM.csv')
data.head()
data.columns
del data['Date']
del data['Adj Close']
data.isnull()
data.isnull().sum()
df=data.dropna()
df.head()
```

```
x=df.drop(labels='Close',axis=1)
y=df.loc[:,'Close']

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=4
2)
print("NUMBER OF TRAIN DATASET    : ", len(x_train))
print("NUMBER OF TEST DATASET      : ", len(x_test))
print("TOTAL NUMBER OF DATASET    : ", len(x_train)+len(x_test))

from sklearn.linear_model import BayesianRidge
BR=BayesianRidge()
BR.fit(x_train,y_train)
predicted=BR.predict(x_test)

from sklearn.metrics import r2_score
R2 = r2_score(y_test,predicted)
print("THE R2 SCORE OF BAYESIANRIDGE IS :",R2)

from sklearn.metrics import rand_score
RAND = rand_score(y_test,predicted)
print("THE ACCURACY SCORE OF BAYESIANRIDGE IS :",RAND*100)

from sklearn.metrics import explained_variance_score
EVS = explained_variance_score(y_test,predicted)
print("THE EXPLAINED VARIENCE SCORE OF BAYESIANRIDGE IS
:",EVS)
```

```
from sklearn.metrics import mean_squared_error
MSE = mean_squared_error(y_test,predicted)
print("THE MEAN SQUARED ERROR SCORE OF BAYESIANRIDGE IS
:",MSE)

from sklearn.metrics import median_absolute_error
MAE = median_absolute_error(y_test,predicted)
print("THE MEAN ABSOLUTE ERROR SCORE OF BAYESIANRIDGE IS
:",MAE)

import matplotlib.pyplot as plt
df2 = pd.DataFrame()
df2["y_test"] = y_test
df2["predicted"] = predicted
df2.reset_index(inplace=True)
plt.figure(figsize=(20, 5))
plt.plot(df2["predicted"][:100], marker='x', linestyle='dashed', color='red')
plt.plot(df2["y_test"][:100], marker='o', linestyle='dashed', color='green')
plt.show()

import joblib
joblib.dump(BR, 'BR.pkl')
```

CHAPTER 7
PERFORMANCE AND ANALYSIS

PERFORMANCE REPORT

We considered different validation methods such as R2_score(R2), Random_Score(RAND), Mean Squared Error(MSE), Mean Absolute Error(MAE) , Explained Variance Score(Variance) and accuracy scores to evaluate the closing price forecasts of three different crypto currencies. The performance evaluation of Doge coin using the Huber Regressor algorithm is R2=0.283, RAND score=0.999, MSE=0.007, MAE=0.0029 while for XRP using Elastic net algorithm gives R2=0.072, Accuracy score = 100%, variance score=0.077, MSE=1.17e05, MAE=0.0072. The evaluation scores for Ethereum using Bayesian Ridge shows R2=0.999, variance score=0.998, MSE=1288.323, MAE=4.210

HUBER REGRESSOR	R2	RAND	MSE	MAE
DOGE	0.283	0.999	0.007	0.0029
XRP	0.0201	0.965	1.242	0.0010
ETHEREUM	0.2207	1.0	1170119.306	371.545

Table 7.1.1: Applying Huber Regressor on all three crypto currencies

ELASTIC NET	R2	RAND	MSE	MAE
DOGE	0.413	0.865	0.006	0.0417
XRP	0.0725	1.0	1.175	0.0017
ETHEREUM	0.998	1.0	1550.690	4.929

Table 7.1.2: Applying Elastic Net on all three crypto currencies

BAYESIAN RIDGE	R2	RAND	MSE	MAE
DOGE	0.998	0.710	1.977	0.0002
XRP	0.997	0.962	3.432	4.091
ETHEREUM	0.9991	1.0	1288.323	4.214

Table 7.1.3: Applying Bayesian Ridge on all three crypto currencies

RESULTS AND DISCUSSIONS

When evaluating the model's effectiveness over the closing price of Huber Regressor, Elastic Net, Bayesian Ridge each model outperforms each other on training them to different currencies. The Huber Regressor model works well with DOGE while the same yielded inaccurate predictions in the case of Ethereum with the mean squared error of 1170119.306.

The elastic net algorithm produces better accuracy in case of Ripple_XRP currency. Though the Bayesian ridge accurately predicted the closing price of Ethereum, it produces negative outcome when applied to other digital currencies.

CHAPTER 8

CONCLUSION

8.1 CONCLUSION AND FUTURE ENHANCEMENTS

This proposed predictive analytics method to improve reporting efficiency for communities. The method seeks to forecast the returns based on the investor's choice of coins and the amount of risk they are willing to take in this highly volatile market. Deeper understanding of our probe is provided by a standard data science framework, which incorporates three among the most effective data driven machine learning approaches: Huber Regressor, Elastic Net and Bayesian Ridge. In order to thoroughly examine every algorithm, we utilized three datasets related to digital currencies (DOGE, XRP and ETH) with various digits as performance measurements. For each dataset, a model was trained to foresee the closing price of that particular day. The performance of each coin's training and testing samples is evaluated using a variety of validation techniques, including MAE, RMSE, RSquare, and MSE, in order to further improve evaluation. Our research findings demonstrate that an algorithm that performs well for one currency does not function as well for other currencies.

In summary, this study provides insightful information about utilising cutting-edge ML algorithms to forecast stable digital coins such as Ethereum, XRP, and DOGE. Yet we must also recognize some inherent constraints that inform our research. Due to the close relationship between digital coin trading and events like regulatory pronouncements and security breaches, these factors include market volatility as well as external issues. These occurrences may lead to sudden and erratic patterns, which could jeopardise the dependability of our best models. Despite these drawbacks, our analysis demonstrates how well the suggested algorithms work in helping to give traders more straightforward reporting, providing a solid basis for further research. Our next goal is to construct a reactive crypto currency forecasting engine that can better predict prices for different coins on an hourly and minute-by-minute basis by utilizing federated learning techniques. Because of its ability to precisely forecast trends, this will enable us to look at usage patterns and possible future investments in the worldwide digital currency market.

APPENDICES

A 1. SAMPLE SCREENSHOTS

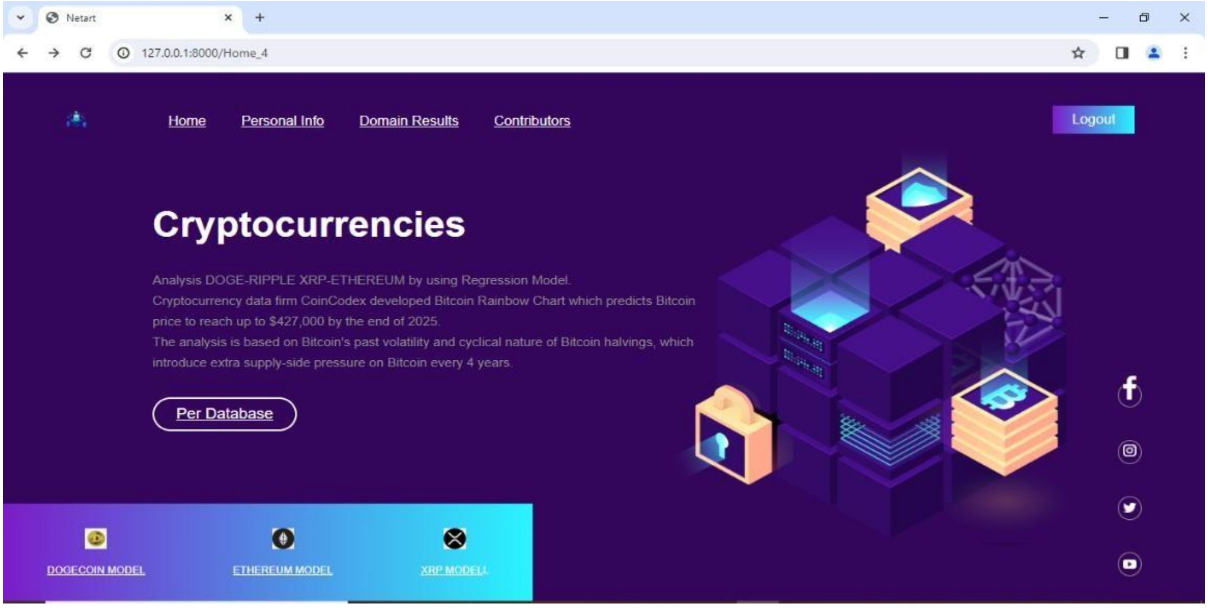


Fig A.1 Home Page

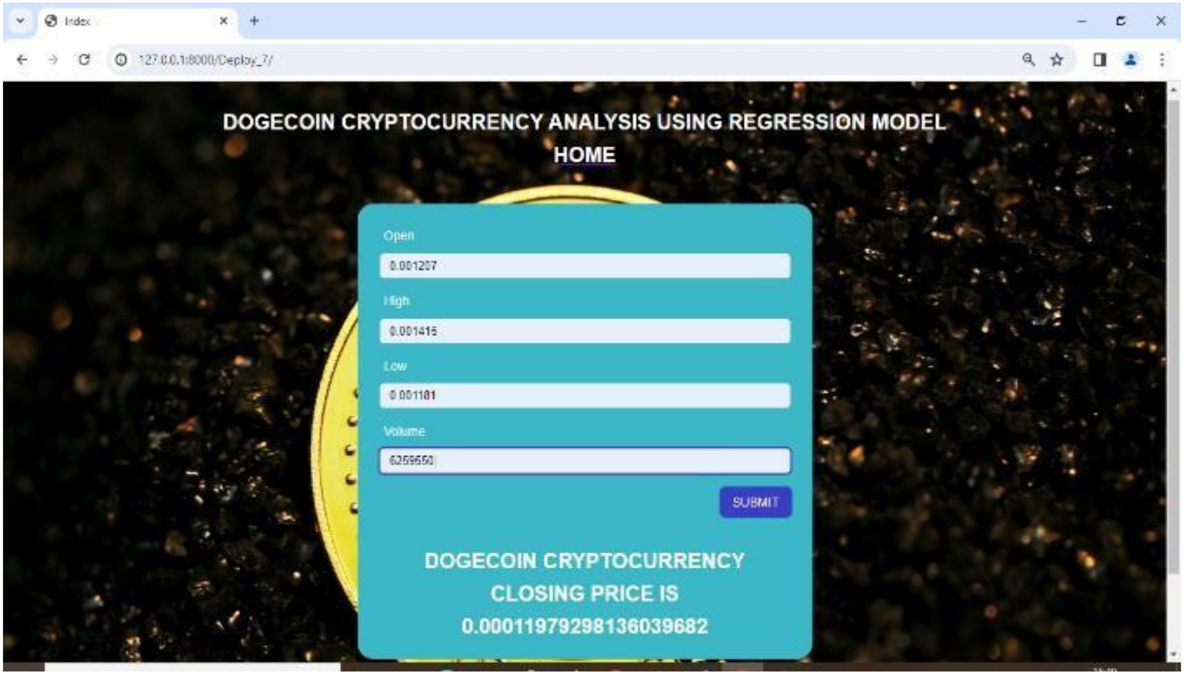


Fig A2 Doge coin prediction page

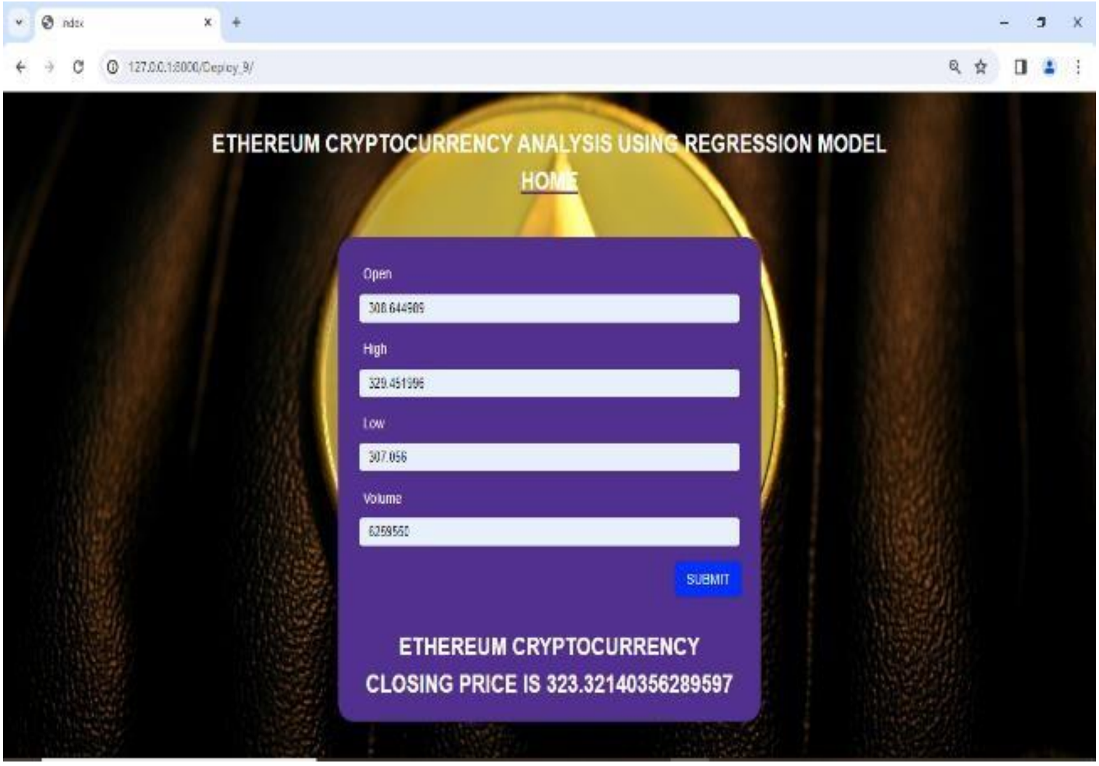


Fig A3 Ethereum coin prediction page

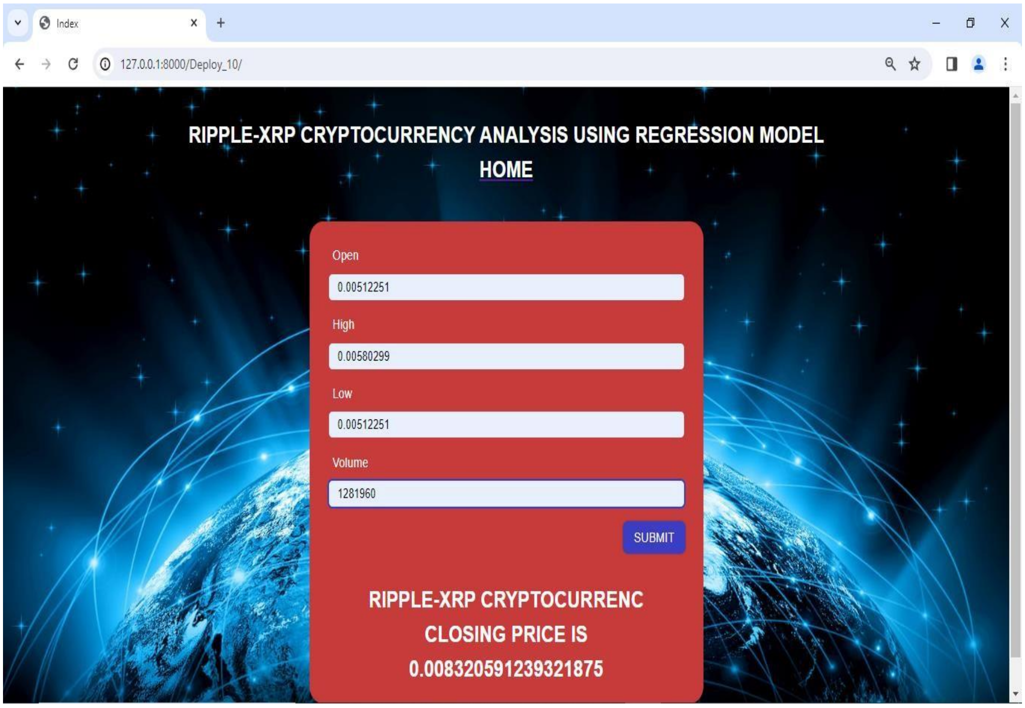


Fig A4 Ripple_XRP coin prediction page

A.2 PAPER PUBLICATION

Title: Trend analysis of Dogecoin, Ripple_xrp, Ethereum cryptocurrencies by Regression model

Conference : 7th International Conference on Intelligent Computing

(ICONIC 2k24)

Indexing : The conference proceedings will be indexed in Google scholar, ensuring wide visibility and accessibility to the academic community.

Authors: Dr.M.Sangeetha , S. Srimathi, S.Sri Varsshini , R.Yohamalar

Affiliation: Department of Computer Science and Engineering Panimalar Engineering College, Chennai, India.

Conference Track: ICONIC 2024

Conference Date: 22nd & 23rd March 2024

Abstract Submission Date: 14 February 2024

Paper Submission Date: 14 February 2024

REFERENCES

1. R. Vijaya Saraswathi, Swathi Bindhu Bollina, Rachana Bongu, Ankitha Cherukuru, Nikitha Chintala, ‘‘A Novel LSTM based Approach for Crypto Currency Price Prediction’’, in 2023 International Conference on Sustainable Communication Networks and Applications (ICSCNA).
2. Sarveshwar Kumar Inani, Gaurav Kabra, Gaurav Nagpal, Peterson Owusu, ‘‘Unraveling the Landscape of Bitcoin Research Using Machine Learning: A Topic Modelling Approach’’, in 2023 IEEE International Conference On Electrical, Electronics, Communication and Computers (ELEXCOM).
3. Saptashwa Maity, Soujatya Khan, Sobhan Sarkar, ‘‘A Two-phase Approach to Determine User-Preference and Feature Importance in Pricing of Crypto currencies using Twitter Data’’, 2023 in IEEE International Conference on E-Business Engineering (ICEBE).
4. Adhi Dharma Wibawa, M Sadewa Wicaksana, Yuri Pamungkas, ‘‘Statistical Analysis to Reveal Factors Affecting Crypto currency Prices in Commodity Futures Trading and Stock Market’’, 2023 in International Conference on Electrical, Electronics And Information Engineering (ICEEIE).
5. Omar Al–Qudah, Hamzah Zureigat, Amer Al–Badarneh, Saleh Abu–Soud, ‘‘Prediction of Crypto currencies Prices Using Long Short Term Memory and Technical Indicators’’, 2023 in International Conference on Intelligent Circuits and Systems (ICICS).
6. Ashutosh Dhar Dwivedi, Subhayu Dutta, Subhrangshu Adhikary, Jens Myrup Pedersen, ‘‘Price Prediction of Digital Currencies using Machine Learning’’, 2023 in International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE).
7. Forhad Uddin Ahmed, Mamun Ahmed, Fahamida Hossain Mahi, Syed Hasnut Abdullah, Sayma Alam Suha, ‘‘A Comparative Performance Evaluation of Bitcoin Price Prediction Using Machine Learning Techniques’’, 2023 in International Conference on Inventive Computation Technologies (ICICT).
8. Lubna Noor AhmadChitkara, Pradeepta Kumar Sarangi, Merry Saxena, Ashok Kumar Sahoo, Amandeep Singh, ‘‘Next Day Bitcoin Price Prediction: Performance Comparison of Various Statistical and Machine Learning Algorithms’’, 2023 in IEEE India Council International Subsections Conference (INDISCON).

9. Mayank Puri, Aman Garg, Lekha Rani, ‘‘Crypto currency Trading Using Machine Learning’’, 2022 in International Conference on Trends in Electronics and Informatics (ICOEI).
10. Arun Kumar Marandi, Gordhan Jethava, ‘‘Impact of Machine Learning on Digital Pricing And Crypto currency Markets’’, 2022 in International Conference On Advance Computing And Innovative Technologies In Engineering (ICACITE).
11. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf.
12. Raskin, M. and D. Yermack (2016). Digital currencies, decentralized ledgers, and the future of central banking. Working paper, National Bureau of Economic Research.
13. Athey, S., I. Parashkevov, V. Sarukkai, and J. Xia (2016). *Bitcoin pricing, adoption, and usage: Theory and evidence*. Working paper, Stanford University.
14. E. Bouri, E. Kamal, and H. Kinateder, ‘‘FTX collapse and systemic risk spillovers from FTX token to major cryptocurrencies,’’ Finance Res. Lett., vol. 56, Sep. 2023, Art. no. 104099.
15. S. Lee, J. Lee, and Y. Lee, ‘‘Dissecting the Terra-LUNA crash: Evidence from the spillover effect and information flow,’’ Finance Res. Lett., vol. 53, May 2023, Art. no. 103590.
16. W. et. al., “Multiscale characteristics of the emerging global cryptocurrency market,” Physics Reports, vol. 901, pp. 1–82, 2021.
17. A. Ho, R. Vatambeti and S. K. Ravichandran, "Bitcoin Price Prediction Using Machine Learning and Artificial Neural Network Model", Indian Journal of Science and Technology, vol. 14, no. 27, pp. 2300-2308, 2021.
18. K. Rathan, S. V. Sai and T. S. Manikanta, "Crypto-currency price prediction using decision tree and regression techniques", 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 190-194, 2019.
19. H. Luo and Y. Liu, "A prediction method based on improved ridge regression", 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 596-599, 2017.
20. I. Merediz-Sola and A. F. Bariviera, "A bibliometric analysis of bitcoin scientific production", Research in International Business and Finance, vol. 50, pp. 294-305, 2019.

21. A. Sunyaev, Distributed Ledger Technology, Cham:Springer International Publishing, pp. 265-299, 2020, [online] Available: https://doi.org/10.1007/978-3-030-34957-8.
22. H.M. Kim, G.W. Bock, and G. Lee, ‘‘Predicting Ethereum prices with machine learning based on blockchain information,’’ Exp. Syst. Appl., vol. 184, Dec. 2021, Art. no. 115480.
23. Y. Liu and A. Tsyvinski, "Risks and returns of crypto currency", The Review of Financial Studies, vol. 34, 2020.
24. Harvey, C. (2014). Bitcoin myths and facts. Working paper, Duke University. http://ssrn.com/abstract=2479670.
25. Pagnotta, E. and A. Buraschi (2018). An equilibrium valuation of bitcoin and decentralized network assets. Available at SSRN: https://ssrn.com/abstract=3142022.
26. Z. Li, Q. Lu, S. Chen, Y. Liu and X. Xu, "A Landscape of Cryptocurrencies", in 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 165-166, 2019.
27. D. Haryadi, A. R. Hakim, D. M. U. Atmaja and S. N. Yutia, "Implementation of support vector regression for polkadot cryptocurrency price prediction", JOIV: International Journal on Informatics Visualization, vol. 6, no. 1-2, pp. 201-207, 2022.
28. D. Chowdhury, A. Dey, R. Garai, S. Adhikary, A. D. Dwivedi, U. Ghosh, et al., "Decrypt: a 3des inspired optimised cryptographic algorithm", Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 5, pp. 4745-4755, 2023.
29. T. Phaladisailoed and T. Numnonda, "Machine learning models’ comparison for bitcoin price prediction", 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 506-511, 2018.
30. M. Kuhn and K. Johnson, Applied Predictive Modeling, vol. 26. Berlin, Germany: Springer, 2013.
31. Jg Aravindan, Rama Krishnan V Sankara, “Parent Coin based Cryptocurrency Price Prediction using Regression Techniques”, in 2022 IEEE Region 10 Symposium (TENSYMP).
32. Mjimer Imane, Es-Saadia Aoula, El Hassan Achouyab, “Using Bayesian Ridge Regression to predict the Overall Equipment Effectiveness performance”, in 2022 2nd International Conference on Innovative Research

in Applied Science, Engineering and Technology (IRASET).

33. T. Phaladisailoed and T. Numnonda, “Machine learning models comparison for bitcoin price prediction,” in 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), 2018, pp. 506–511.
34. Nong Lili, “Superstore Sales Forecasting Based on Elastic net Regression and BP Neural Networks”, in 2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA).
35. Rashika Bangroo, Sanjoy Raj Verma, Shivangi, Shakuntala, “Comparative Study of Elastic Net Regression, Naive Bayes & Lasso Regression”, in 2023 International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM).
36. Naor Huri, Meir Feder “Selecting the LASSO regularization parameter via Bayesian principles” in 2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE).
37. H. Kavitha, U. K. Sinha, and S. S. Jain, “Performance evaluation of machine learning algorithms for bitcoin price prediction,” in 2020 Fourth International Conference on Inventive Systems and Control (ICISC), 2020, pp. 110–114.
38. S. Prakash Tiwari, S. Adhikary, and S. Banerjee, “Estimation of chlorophyll-a from oceanographic properties - an indirect approach,” in IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium, 2022, pp. 6872–6875.
39. T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, vol. 2. Berlin, Germany: Springer, 2009.
40. R. J. Hyndman and A. B. Koehler, ‘‘Another look at measures of forecast accuracy,’’ Int. J. Forecasting, vol. 22, no. 4, pp. 679–688, Oct. 2006.
41. J. Gareth, W. Daniela, H. Trevor, and T. Robert, An Introduction to Statistical Learning: With Applications in R. Berlin, Germany: Springer, 2013.