University of Mumbai

Cryptocurrency Investment Classification Using Machine Learning and Sentiment Analysis

Submitted in partial fulfillment of requirements
for the degree of

Bachelors in Technology

by

Jash Shah
Roll No: 1814054,

Neelay Jagani
Roll No: 1814024,

Rudresh Raval
Roll No: 1814104,

Harsh Sachala
Roll No: 1814105

Guide:
Prof. Ravindra Divekar

Department of Electronics Engineering
K.J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Batch 2018-2022

K.J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the dissertation report entitled **Cryptocurrency Investment Scorer** submitted by Jash Shah, Neelay Jagani, Rudresh Raval, and Harsh Sachala at the end of semester VIII of LY B. Tech is a bona fide record for partial fulfillment of requirements for the degree of Bachelors of Technology in Information Technology of University of Mumbai.

Guide

Head of Department

Principal

Date:

Place: Mumbai-77

K.J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this dissertation report entitled **Cryptocurrency Investment Scorer** is bona fide record of project work done by Jash Shah, Neelay Jagani, Rudresh Raval, Harsh Sachala. This project is approved for the award of Bachelors in Technology Degree in Information Technology of University of Mumbai.

Internal Examiner

External Examiner

Date:
Place: Mumbai-77

K.J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

DECLARATION

We declare that this written thesis submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Signature of the Student **1814054** **Roll No.**	**Signature of the Student** **1814024** **Roll No.**
Signature of the Student **1814104** **Roll No.**	**Signature of the Student** **1814105** **Roll No.**

Date:
Place: Mumbai-77

Abstract

The field of cryptocurrency is at peak and its highly volatile and price changes happens very frequently and drastically, so it becomes very difficult to analyse each and every aspect of the cryptocurrency into account so as to make an informed decision whether to invest in a particular cryptocurrency or not; there are already 2000 cryptocurrency in circulation and transactions for each of them happens every split of a second and tracking each cryptocurrency separately can get very tedious. Machine Learning algorithms have been found to make great predictions and historically have been used to predict diseases, house prices and even stock prices. However when we are dealing with markets an important parameter that gets missed and is paramount is the current market sentiment of the market. Hence clubbing machine learning with sentimental analysis in prediction the price of cryptocurrency can help us achieve a greater, a more reliable prediction on whether to buy, not buy a cryptocurrency. We plan to assign weight as to how much market sentiment should be given importance relative to our machine learning prediction so as to achieve maximum reliability.

Keywords : Cryptocurrency, machine learning, LSTM, tensorflow, sentiment analysis, prediction.

Contents

Abstract **iv**

List of Figures **viii**

List of Tables **ix**

Nomenclature **x**

1 Introduction **2**

1.1 Project Overview 2

1.2 Purpose 3

1.3 Scope 3

1.4 Functional Requirements 3

1.5 Non-Functional Requirements 4

2 Literature review **5**

2.1 Summarization 5

3 Software Project Management Plan **7**

3.1 Introduction 7

3.1.1 Project Overview 7

3.1.2 Project Deliverables 8

3.2 Project Organization 10

3.2.1 Software Process Models 10

3.2.2 Roles and Responsibilities 10

3.2.3 Tools and Techniques 15

3.3 Project Management Plan 17

3.3.1 Tasks 17
3.3.2 Assignments 21
3.3.3 Timetable 22

4 Software Requirement Specification 24
4.1 Product Overview 24
4.2 Specific Requirements 25
4.2.1 External Interface Requirements 25
4.2.2 Software Product Features 26
4.2.3 Software System Attributes 26
4.2.4 Data Set Requirements: 28

5 Software Design Description 29
5.1 Introduction 29
5.1.1 Project Overview 29
5.1.2 Purpose of Document 30
5.2 System Architecture Design 30
5.2.1 Chosen System Architecture 30
5.2.2 System Interface Description 31
5.3 Detailed Description of Components 31
5.3.1 Exploring and Preprocessing Datasets 31
5.3.2 Training Algorithm 31
5.3.3 Testing and Deployment 32
5.4 System User Interface 32
5.5 System Architecture 37
5.6 Data Flow Specifications 38

6 Implementation 40
6.1 Technologies Used 40
6.1.1 Python 40
6.1.2 HTML5 40
6.1.3 CSS3 41
6.1.4 Bootstrap v5.1 41

6.1.5 Heroku 41
6.1.6 Selenium 41
6.1.7 GTMetrix 42
6.2 Algorithm 42
6.2.1 Extraction and Processing 42
6.2.2 LSTM 43
6.2.3 Sentiment Analysis 44
6.2.4 Combined Approach 44

7 Software Test Document 47
7.1 Introduction 47
7.1.1 System Overview 47
7.1.2 Test Approach 48
7.2 Test Plan 48
7.2.1 Test Plan Objectives 48
7.2.2 Features to be Tested 49
7.2.3 Testing Tools and Environment 49
7.3 Test Cases 50
7.3.1 Automation Testing of the Result display using Selenium 50
7.3.2 Performance Testing of the Overall Website 50

8 Conclusions and scope for further work 52
8.1 Conclusion 52
8.2 Future scope 52

Acknowledgements 56

List of Figures

3.1 Gantt chart: Cryptocurrency Investment Scorer 1 22
3.2 Gantt chart: Cryptocurrency Investment Scorer 2 23

5.1 Home Page 32
5.2 About section 33
5.3 Methodology section 33
5.4 Features section 34
5.5 Prices and Predict section 34
5.6 Team section 35
5.7 Result Page with graphs 35
5.8 Result Page with Gauge and Prediction Score 36
5.9 Use Case 1 37
5.10 Use Case 2 38
5.11 DFD Level 0 39
5.12 DFD Level 1 39

6.1 Flowchart for combined approach 46

7.1 Performance Metrics 51
7.2 Browser Timings 51

List of Tables

3.1 Project Roles and Responsiblities 15

3.2 Tools Used 15

3.3 Libraries, APIs and Frameworks used 15

3.4 Tools and Techniques 16

3.5 Assignments 21

Nomenclature

AI	Artificial Intelligence
DFD	Data Flow Diagram
LSTM	Long Short Term Memory
ML	Machine Learning
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
RMMM	Risk Mitigation, Monitoring, and Management
RNN	Recurrent Neural Network
SDD	Software Design Document
SPMP	Software Project Management Plan
SRS	Software Requirement Specification
STD	Software Test Document
UML	Unified Modeling Language

Chapter 1

Introduction

This chapter provides an overview of the project, the motivation for creating it, the function it serves, the scope it has in today's data-driven environment, and a sneak peek at its Functional and Non-Functional Requirements.

1.1 Project Overview

Cryptocurrency is a type of digital currency that generally only exists electronically. There is no physical coin or bill unless you use a service that allows you to cash in cryptocurrency for a physical token. In the past, with prominent investments in cryptocurrency, has lead people to become millionaire and billionaires. The currency has two primary advantages, it can be used to buy goods and services online over the web and can be bought as investments to generate return. It will over time make physical currency obsolete. However with a lot of volatility and now there being more than 2000 cryptocurrencies, it becomes extremely difficult for commoners to invest. Hence we plan to create a Machine Learning Model that combined with natural language processing for sentimental analysis can accurately predict in which cryptocurrency to invest in, and whether if it is the right time to invest by analyzing the market sentiment.

1.2 Purpose

Although machine learning has been successful in predicting stock market prices through a host of different time series models, its application in predicting cryptocurrency prices has been quite restrictive. The reason behind this is obvious as prices of cryptocurrencies depend on a lot of factors like technological progress, internal competition, pressure on the markets to deliver, economic problems, security issues, political factor etc. Their high volatility leads to the great potential of high profit if intelligent inventing strategies are taken. Unfortunately, due to their lack of indexes, cryptocurrencies are relatively unpredictable compared to traditional financial predictions like stock market prediction. Hence, analyzing the sentiment of the market becomes essential for predicting the cryptocurrency market more efficiently. This is when, our algorithm comes into the picture as it provides a novel approach.

1.3 Scope

This product is made for a cryptocurrency trader or an amateur who wants to get into the crypto market but has minimal knowledge. This project aims to make the life of a trader simpler by recommending if a particular cryptocurrency must be bought at that point of time based on the predictions of machine learning algorithm and sentiment of the crypto market at that point in time. It uses APIs to fetch real time crypto prices and twitter sentiment of that cryptocurrency at that time.

1.4 Functional Requirements

1. High internet bandwidth

2. Secure API.

3. High RAM and CPU requirement.

4. Calculating score

1.5 Non-Functional Requirements

1. Availability - We will be having and optimizing it with respect to only top five most liquid cryptocurrencies as data for these highly traded cryptocurrencies is always available.

2. Reliability - We will be using top grade APIs so that data collected whether it be live or historical is always reliably.

3. Performance - We will be optimizing our once developed code to optimize time complexity, such that performance remains paramount

Chapter 2

Literature review

This chapter includes a detailed survey that was conducted for the objective of doing a literature review and researching other studies that were conducted along the same lines.

2.1 Summarization

In (1), illustrates the use of LSTM, bi-LSTM and GRU models in predicting cryptocurrency prices in BTC, ETH and LTC. Overall, there is a high degree of accuracy observed on the MAPE scale for GRU compared to other algorithms. Similarly this paper (3) tests investment portfolios' perfomances for gradient boosting decision trees and LSTM models and turns out that both methods generate profit even after considering transaction costs, this procedure was conducted for more than 1,600 currencies.

(6) According to this paper, combination of autoencoder and re-training methodology gives a great performance for the same task (F-measure of 78%). (7) tries to predict the short-term crypto prices using Google Trends and Twitter data. It also tries to find the relationship between the number of Tweets and the cryptocurrency price. It observes a high inverse correlation of the number of tweets with the price. Finally, (8) summarised previous research findings and gave us a retrospective viewpoint on using machine learning models for cryptocurrency predictions. This further enabled us to understand the concept and to further improve it.

This paper (9) acts as a verification for the conclusion of (7) on the same topic. (4) proposes a reinforcement machine learning-based active trading model. Like (1), (3) the paper's methodology is tested on 5 cryptocurrencies. This paper (10) presents that when it comes to high frequency bitcoin trading, the Levenberg Marquardt algorithm is the best predictor when compared with conjugate gradient and resilient algorithm. (11) compares the altered version of Binary Auto-Regressive Tree (BART) model with the ARIMAARFIMA models in terms of short-term predictions. Concluding that it faired better than the ARIMA-ARFIMA models in that matter.

(2) examines the bitcoin market's predictability over time horizons ranging from 1 to 60 minutes. RNNs and gradient boosting classifiers are apt for the discussed prediction tasks, according to the results of several machine learning models. The paper also finds that for larger prediction horizons, predictability improves.

(12) CIDR curves are stable, non-normal, and uncorrelated, yet they have conditional heteroscedasticity, according to the author. Second, they demonstrate the ability to forecast Bitcoin CIDR curves using projection scores and then evaluate forecasting success.

(13) Shows a methodology of retrieving tweets for a single month for two different years that is 2018 and 2019; comparing those and confirming the trends that can be predicted from the sentiment without the incorporation of complex business models.

(14) Discusses how to forecast bitcoin's price using a multitude of Twitter features. They also feature weighted tweet sentiment features, which represent the overall mood regarding bitcoin on Twitter, as well as tweet volume features, which indicate general activity and engagement in bitcoin on Twitter.

In almost all of the papers discussed before, it is stated in their further work section that simply using machine learning models is not sufficient for highly accurate predictions regarding crypto currency prices and that there is a need to take into account the sentimental value of a cryptocurrency. Furthermore papers (9) and (7) use tweets and google trends to measure it via volume of tweets and searches. They mention a critical finding, i.e the number of tweets or searches doesn't determine whether the price of a cryptocurrency will go down or up, at best a sudden surge in tweets can only depict a sudden rise or fall in the price. This prompted for a better method to measure the sentiment of the market, which will be further elaborated upon in the methodology section of this paper.

Chapter 3

Software Project Management Plan

This chapter provides an overview of the project and its many deliverables, as well as a description of the project that highlights the process model employed, the roles and duties of the team members engaged, and the tools and techniques employed. This chapter concludes with an explanation of the many duties required as well as a timetable for the projaect's completion.

3.1 Introduction

3.1.1 Project Overview

Cryptocurrency is a type of digital currency that generally only exists electronically. There is no physical coin or bill unless you use a service that allows you to cash in cryptocurrency for a physical token. In the past, with prominent investments in cryptocurrency, has lead people to become millionaire and billionaires. The currency has two primary advantages, it can be used to buy goods and services online over the web and can be bought as investments to generate return. It will over time make physical currency obsolete.

However with a lot of volatility and now there being more than two thousand cryptocurrencies, it becomes extremely difficult for commoners to invest. Hence we plan to create a Machine Learning Model that combined with sentimental analysis can accurately predict the cryptocurrency to invest in, and whether if it is the right time to invest by analyzing the market sentiment.

3.1.2 Project Deliverables

A project management term for the quantifiable goods or services that will be provided upon the completion of a project. Deliverables can be tangible or intangible parts of the development process and are often specified functions or characteristics of the project. Deliverables serves as a general term that encompasses the requirements of a project. A deliverable may be an object used in the greater scheme of the project.

Project Duration: 8 Months
Start Date: 20/11/2021

1. Project Management Plan:

   A plan of actions to be carried out in the due process until project completion.

   Delivery Date- 24th November, 2021

2. Data Collection Module:

   Research various data sets on trusted platforms and selected the most appropriate data set.

   Delivery Date- 29th November, 2021

3. Data Pre-Processing Module:

   Pre-process and clean to transform the raw data in a useful and efficient format for training the model.

   Delivery Date- 10th December, 2021

4. Algorithm Selection Module:

   Researching, trying and evaluating the various prediction algorithms to select the most suitable and best performing algorithm in terms of accuracy and loss factors.

   Delivery Date- 15th January, 2022

5. Train and Validation Model Module:

   Performing train, test and validation split on data set. Training the model using the selected algorithm with input as the training data set. Validation the model and tuning the parameters based on the evaluating matrix.

   Delivery Date- 1st February, 2022

6. Testing and Prediction Module:

   Performing final test on the model and making effective predictions of the crytocurrency˙
   Delivery Date- 15th March, 2022

7. Creation of User Interface Module:

   Creating a user interface for the users to easily access the application without looking at the complex back-end.

   Delivery Date- 20th March, 2022

3.2 Project Organization

3.2.1 Software Process Models

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. The collaborative agile methodology breaks down large projects into small, manageable increments or “sprints,” typically of two weeks’ duration. It encourages experimentation and the use of small projects and quick iterations to facilitate fast-paced problem solving. It contains six phases as follows:

1. Concept
2. Inception
3. Iterations
4. Release
5. Maintenance
6. Retirement

3.2.2 Roles and Responsibilities

Position	**Team member**	**Responsibilities**

Project Leader	Jash Shah	1. The interface between the project owner and everyone else, channeling communication both ways. 2. Responsible for communication and formal agreements between stakeholders 3. Primary stakeholder within the production team and must ensure that the goals are being constantly communicated and met.
Project Manager	Neelay Jagani	1. Overall supervision of project and team. 2. Delegates requirements. 3. Maintains project plan. 4. Performs implementation. 5. Arbitrates issues that may arise.

Stakeholder Manager	Harsh Sachala, Rudresh Ravala	1. Performs the tasks of the Project Manager when the Project Manager is unavailable. 2. Analyzes software requirements. 3. Manages all project documentation .

UI/UX Designer	Harsh Sachala, Rudresh Raval	1. Gathering and evaluating user requirements, in collaboration with product managers and engineers. 2. Illustrating design ideas using storyboards, process flows and sitemaps. 3. Designing graphic user interface elements, like menus, tabs and widgets. 4. Develop UI mockups and prototypes that clearly illustrate how sites function and look like. 5. Create original graphic designs (e.g. images, sketches and tables). 6. Prepare and present rough drafts to internal teams and key stakeholders. 7. Identify and troubleshoot UX problems (e.g. responsiveness), and conduct layout adjustments based on user feedback.

Code Developer	Jash Shah, Neelay Jagani, Rudresh Raval, Harsh Sachala	1. Be involved and participate in the overall application lifecycle. 2. Optimization of the application for maximized metrics(accuracy, recall, precision, etc) speed and scalability and implementation of security and data protection. 3. Main focus on coding and debugging 4. Develop functional and sustainable web applications with clean codes.
Software Tester	Neelay Jagani	1. Analyzing users images inputs and/use cases/requirements for validity and feasibility. 2. Detect and track software defects and inconsistencies and provide timely solutions. 3. Apply quality engineering principals throughout the Agile product lifecycle and provide support and documentation.

Table 3.1: Project Roles and Responsiblities

3.2.3 Tools and Techniques

Tool Category	Tool Used
Wireframing Tool	Marvelapp
Testing tool	Selenium,GTMetrix
Gantt Chart	Lucidchart
UML diagram tool	Lucidchart
LaTeX editor	Overleaf

Table 3.2: Tools Used

Machine Learning	Keras, NLTK, Sci-Kit Learn, Tensorflow
UI/UX	Flask, HTML, CSS, Bootstrap, JavaScript.
Sentimental Analysis	Tweepy, NLTK, TextBlob
Preprocessing	Numpy, Pandas, Matplotlib
Data Extraction	YFinance

Table 3.3: Libraries, APIs and Frameworks used

Activity	Tools Required	Technique to be used
Documentation	MS Word, LaTeX	The procedure would be given in steps wherever necessary with labelled diagrams to be used.

Algorithm development	Keras, Sci-Kit Learn, Tensorflow, NLTK	After the documentation, the team will be working on the ML Algorithm, which forms the crux of the application.
UI/UX Designing	Adobe XD	The prototype will be first prepared and shared with the guide. After the design is approved implementation of the same will be performed.
Frontend Development	HTML, CSS, BOOTSTRAP, JS	After the approval of the prototype by the clients, the frontend developers start building the user interface of the application.
Project Storage	Github	The developers working on the application will add code repository to the github.
Meetings	Microsoft Teams, Zoom and Google Meet	All the meetings regrading the project with various groups and sub-groups will be carried out on these platforms.
Testing	Selenium, GTMetrix	To check the responsiveness of the application and to make it browser friendly.

Table 3.4: Tools and Techniques

3.3 Project Management Plan

3.3.1 Tasks

The tasks that are involved in this project are:

1. Problem Definition and Scope Designing
2. Getting Feedback from user, requirement analysis, develop an exploratory prototype
3. Developing the SRS document
4. Developing the Algorithm
5. System UI design
6. Developing the Version 1
7. Developing the Version 2
8. Software Testing

Problem Definition and Scope Design

Description Formulation of problem statement, done along the guidelines of stakeholder requirements to be done along with the scope that the project can achieve. Keeping in mind about how useful the product will be to its client and to develop a user-friendly product with minimum use of Tech Jargon. Major and minor functionalities also to be specified. Resources are required to carry out the project tasks. They can be people, equipment, facilities, funding, or anything else need for completion of a project activity.

Deliverables and Milestones Deliverable:
Problem statement and scope document.
Milestone:
Delivering the document before the scheduled deadline with including everything as per the requirement.

Dependencies and constraints Stakeholders requirements and needs must be known for the team to devise the problem statement. Modify the Problem Statements as per the need and requirement of the Product in the open Market.

Contingencies Stakeholders might add/update needs and requirements which might render the current version of the problem statement obsolete.

System Requirements with SRS

Description The SRS clearly and precisely describes each of the essential requirements (functions, performances, design constraints, and attributes) of the software and the external interfaces. Each requirement is defined such that its achievement is capable of being objectively verified and validated by a prescribed method, for example, inspection, analysis, demonstration, or test. Preparation of the Software Requirements Specification document that will outline all the requirements of the project. It must encompass functional requirements, non-functional requirements, software requirements, hardware requirements and database requirements to be assumed as a complete SRS document. The cost estimation and the schedule of the product is planned on the basis of the SRS document. The SRS document can be also used as a reference for end user testing A typical SRS includes: A purpose, an overall description, specific requirements, system features and non-functional requirements. This SRS will give a brief information about the Prototype to be developed (It will contain some assumptions).

Deliverables and Milestones Deliverable: Complete SRS document. Milestone: Delivering the document before the scheduled deadline.

Resources needed Problem statement and scope document, Technical limitations, LaTex and Microsoft Word.

Dependencies and constraints Stakeholder requirements and needs must be known for the team to prepare SRS. Also as per the technical aspect, some assumptions will be made in the SRS document.

Contingencies Failure to provide the allocated budget/resources once the project has been finalized and SRS hasn't been prepared can cause the project to be delayed/terminated.

Algorithm development

Description This part forms the main crux of the application. The algorithm's main function will be to predict the investment in a cryptocurrency to be advisable, not advisable or neutral at a point in time. The output must be displayed in an efficient and explanatory manner. Deliverable and Milestones Deliverable: Completing the algorithm and displaying the explainable output
Milestone: Completing the algorithm with high accuracy Resources needed Keras, Sci-Kit Learn, Tensorflow, NLTK Dependencies and constraints Good literature survey must be conducted and the algorithm must be viable for most liquid cryptocurrencies. Contingencies Poor performance of the algorithm may lead to huge consequences.

UI Design

Description The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse, and the appearance of a desktop. It is also the way through which a user interacts with an application or a website.

Deliverables and Milestones Deliverable:
The final version of User Interface.
Milestone:
Delivery of a full-fledged user interface for users before the deadline.

Resources needed Access to Adobe XD or Figma, Flask, HTML, CSS, JS, Bootstrap.

Dependencies and constraints UI design has to start with user research and task analysis.

Contingencies Poor/Complex design of user interface may lead to difficulties in accessing the product.

Development Version 1

Description Implementing the functionalities required in the application through coding. As this is the most important part it will be given utmost importance. There are members in the team who will be entirely responsible for this part. Also, through backup we can ensure that no code is lost and the system is always in recoverable state.

Deliverables and Milestones Deliverable:
First stage of Prototype Development.
Milestone:
Delivery of 75-80% prototype implementation before the deadline.

Resources needed A proper SRS and SDD along with the SPMP with explanation in brief.

Dependencies and constraints This part will totally be dependent on the technical team, how and when each individual does the task as from here, one part will be connected with another. Backups will be an important constraint, as one person's work should be saved in a backup copy and then another person should proceed further with his/her job and also the PC specifications for implementation.

Contingencies Proper understanding of the Problem and requirements and then start the implementation, as if there's some misunderstanding by any of the Technical Team member, then the prototype needs to be developed from scratch.

Development Version 2

Description After showing the first development version to the guide and receiving the feedback, he might suggest some changes or modifications. So taking his feedback into consideration, the Technical Team will have to modify the prototype and complete it.

Deliverables and Milestones Deliverable:
Final Prototype.

Milestone:
Delivery of entire perfect prototype implementation before the deadline.

Resources needed A proper SRS and SDD along with the SPMP with explanation in brief along with the modification/feedback from the end user(Stakeholder).

Dependencies and constraints It will be dependent about how big are the modifications suggested by the Stakeholder. Backups will be an important constraint, as one person's work should be saved in a backup copy and then another person should proceed further with his/her job and also the PC specifications for implementation.

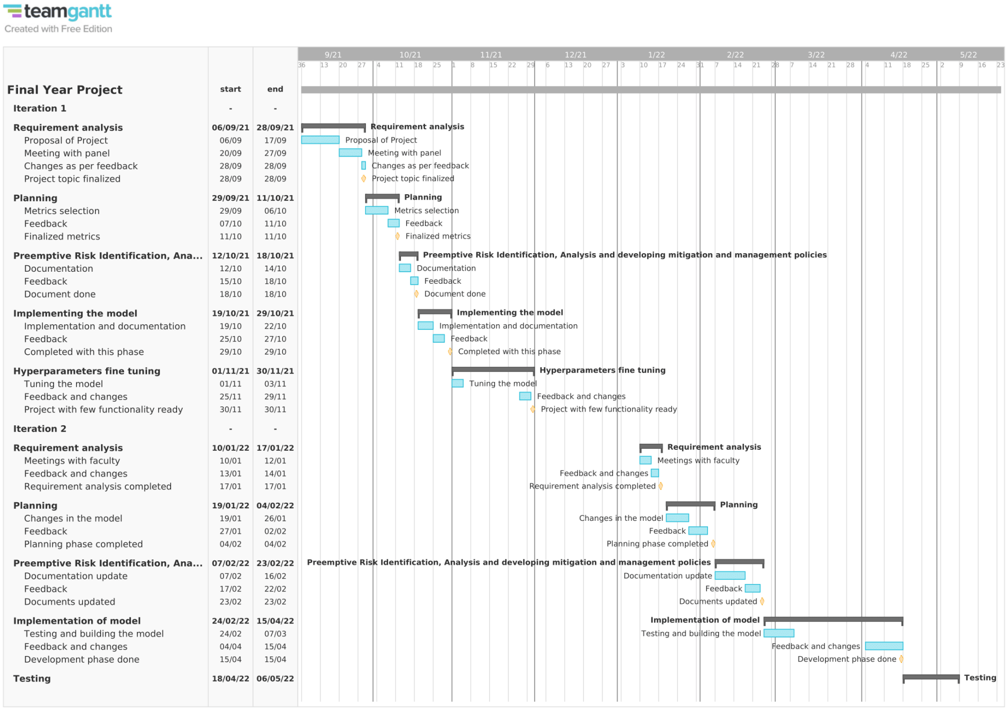
Contingencies Proper understanding of the changes/modifications by the Stakeholder and then start with the changes as if there's some misunderstanding by any of the Technical Team member, then the prototype needs to be developed from scratch.

3.3.2 Assignments

Sr.No.	**Tasks**	**Team Members**
1.	Requirements Gathering	Jash Shah, Neelay Jagani
2.	Planning and Project Flow discussion	Harsh Sachala, Rudresh Raval
3.	Work on Algorithm	Jash Shah, Neelay Jagani, Harsh Sachala, Rudresh Raval
4.	Testing	Neelay Jagani
5.	Completion and Maintenance	Rudresh Raval, Jash Shah

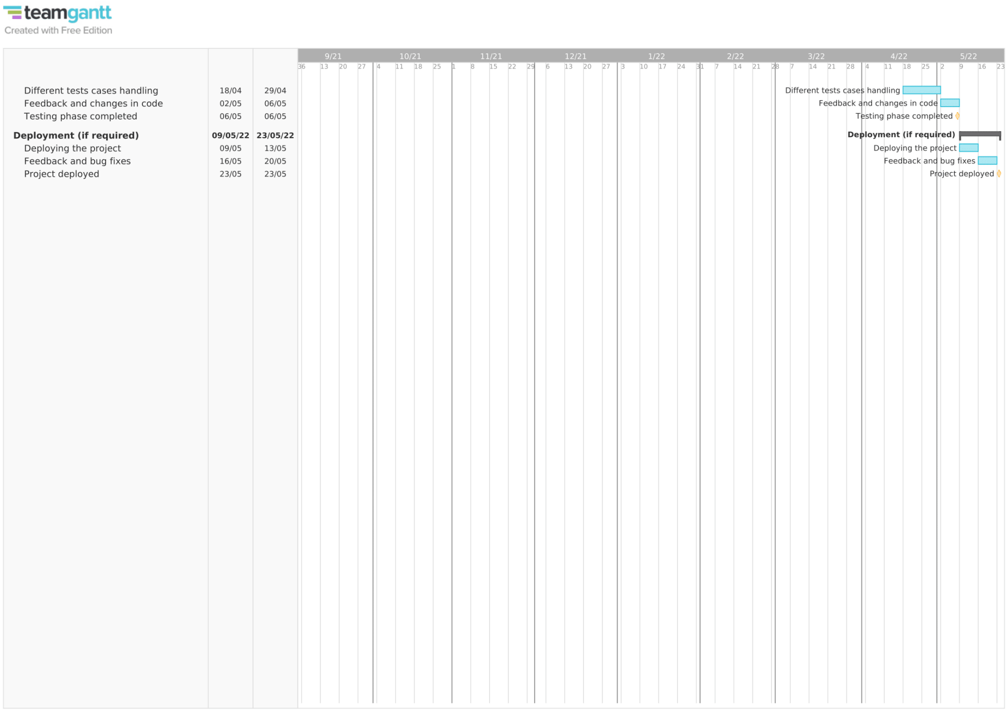
Table 3.5: Assignments

3.3.3 Timetable



Final Year Project	start	end
Iteration 1	-	-
Requirement analysis	**06/09/21**	**28/09/21**
Proposal of Project	06/09	17/09
Meeting with panel	20/09	27/09
Changes as per feedback	28/09	28/09
Project topic finalized	28/09	28/09
Planning	**29/09/21**	**11/10/21**
Metrics selection	29/09	06/10
Feedback	07/10	11/10
Finalized metrics	11/10	11/10
Preemptive Risk Identification, Ana...	**12/10/21**	**18/10/21**
Documentation	12/10	14/10
Feedback	15/10	18/10
Document done	18/10	18/10
Implementing the model	**19/10/21**	**29/10/21**
Implementation and documentation	19/10	22/10
Feedback	25/10	27/10
Completed with this phase	29/10	29/10
Hyperparameters fine tuning	**01/11/21**	**30/11/21**
Tuning the model	01/11	03/11
Feedback and changes	25/11	29/11
Project with few functionality ready	30/11	30/11
Iteration 2	-	-
Requirement analysis	**10/01/22**	**17/01/22**
Meetings with faculty	10/01	12/01
Feedback and changes	13/01	14/01
Requirement analysis completed	17/01	17/01
Planning	**19/01/22**	**04/02/22**
Changes in the model	19/01	26/01
Feedback	27/01	02/02
Planning phase completed	04/02	04/02
Preemptive Risk Identification, Ana...	**07/02/22**	**23/02/22**
Documentation update	07/02	16/02
Feedback	17/02	22/02
Documents updated	23/02	23/02
Implementation of model	**24/02/22**	**15/04/22**
Testing and building the model	24/02	07/03
Feedback and changes	04/04	15/04
Development phase done	15/04	15/04
Testing	**18/04/22**	**06/05/22**

Figure 3.1: Gantt chart: Cryptocurrency Investment Scorer 1



Different tests cases handling	18/04	29/04
Feedback and changes in code	02/05	06/05
Testing phase completed	06/05	06/05
Deployment (if required)	**09/05/22**	**23/05/22**
Deploying the project	09/05	13/05
Feedback and bug fixes	16/05	20/05
Project deployed	23/05	23/05

Figure 3.2: Gantt chart: Cryptocurrency Investment Scorer 2

Chapter 4

Software Requirement Specification

This chapter provides an overview of the project, including the motivation for creating it, the function it serves, and the scope of its application in today's data-driven environment. This chapter closes by describing the many Functional and Non-Functional Requirements that the project has and fulfils.

4.1 Product Overview

The cryptocurrency algorithm will help an individual to make an informed decision about buying a particular cryptocurrency or not. It will take into account the past records of a cryptocurrency and will club the sentiments got from the tweets, and other sources by combining, it will display a score on the basis of which it will indicate whether it is good investment option at the moment or not. The algorithm will have it's own metric with which it will generate a score on a scale of 0 to 100. The algorithm will be user-friendly to operate and it will not require any sort of education level, experience or technical expertise to.

4.2 Specific Requirements

4.2.1 External Interface Requirements

User Interfaces

The user will work on a python notebook (preferably Jupyter notebook), it will use the given buttons to run a particular code snippet and it will ask to select a particular cryptocurrency from the given list which will be menu-driven, upon successful input and selection it will process and display the score after done processing which will have either BUY, DO NOT BUY or NEUTRAL indicator on the basis of the score generated by the algorithm.

Hardware Interfaces

A device with at least 8GB of RAM and a CPU with 7th generation configuration is needed.

Software Interfaces

We are using python programming language for the implementation of the algorithm inside jupyter notebook, hence a python developing environment is required (python 3.7 or higher version). Various python library such as Sci-Kit learn, Tensorflow etc. is required which will be installed with a command that would be specified inside the notebook, to run those command a device should have a stable internet connection.

4.2.2 Software Product Features

Pre-processing:

The program should be able to take the input provided by the user and use it to preprocess for a chosen cryptocurrency, and should initiate the algorithm.

Efficient processing:

The algorithm should be able to work efficiently without any errors and display the required graphs, or any other visuals and generate the score in less amount of time.

Interpretable Results:

The results must be displayed in a simple and explainable format. It should provide clear result in a proper visual format which would be easy for any individual using the algorithm in notebook

Accuracy:

The model must have low values of false negatives. As for example, we do not want to give BUY result for DO NOT BUY.

4.2.3 Software System Attributes

Reliability:

For making the system more reliable, we aim to keep the prediction accuracy as high as possible. Hence making it an authentic algorithm on which users can trust upon.

Availability:

The system should be available to the users for a maximum amount of time. So that the users may feel free to run the algorithm anytime they want. Also, the algorithm should require a less amount of maintenance work which will ensure more availability of the application to its users.

Security:

As we are dealing with large datasets of cryptocurrencies and tweets it is highly confidential data so we should have a secure connection to the source to pull the data without getting it compromised. The requests should be HTTPS instead of just HTTP.

Usability:

The algorithm must be easy to use such that they do not need to read an extensive number of manuals to use the notebook file. The system must be quickly accessible by the users. The system must be intuitive and simple in the way it displays all relevant score and results. The features of the system are easily navigable by the users with buttons that are easy to understand.

Portability:

The notebook file is portable to any device-mobile or desktop. The only things the device requires are stable internet connection,python environment, jupyter notebook, browser and sufficient memory requirement.

Performance:

Mainly the system must predict with an appropriate accuracy and reduce false positives and false negatives. The system must be efficient and fast, because the users using it don't have down-time to wait for it to complete an action. All the functions of the algorithm must be available to the user every time it is run. If for any reason some operation cannot be performed or has failed to execute, then the user must be alerted with appropriate messages.

4.2.4 Data Set Requirements:

The data set should be selected from a trustworthy source since it will influence the score generated by the algorithm which in turn help user make an informed decision. The dataset should consist of all the previous data of cryptocurrency and data of all the tweets with a proper tweet to work on it. If a data set does not fulfil the requirement, there will be an imbalance and there will be ambiguity in the score generated i.e. the output will be biased in a way leading to inaccurate predictions. Hence, it is really important for the data set to be accurate and balanced.

Chapter 5

Software Design Description

This chapter provides a summary of the project, as well as the motivation for creating this document and an explanation of the traceability matrix. It also justifies the software architecture chosen and goes into detail about each of its components. Following then, the project's numerous use case specifications were presented. This chapter wraps up with the project's user interface design and data flow specifications.

5.1 Introduction

5.1.1 Project Overview

Cryptocurrency is a type of digital currency that generally only exists electronically. There is no physical coin or bill unless you use a service that allows you to cash in cryptocurrency for a physical token. In the past, with prominent investments in cryptocurrency, has lead people to become millionaire and billionaires. The currency has two primary advantages, it can be used to buy goods and services online over the web and can be bought as investments to generate return. It will over time make physical currency obsolete. However with a lot of volatility and now there being more than 2000 cryptocurrencies, it becomes extremely difficult for commoners to invest. Hence we plan to create a Machine Learning Model that combined with natural language processing for sentimental analysis can accurately predict in which cryptocurrency to invest in, and whether if it is the right time to invest by analyzing the market sentiment.

5.1.2 Purpose of Document

Software design documents are an important way of looping everyone into the process who is involved in the product. Design documents are created to coordinate efforts of a large team, give them a stable reference point, and describe all parts of the software and how they will operate. Software design documents not only help others understand your system and provide documentation for future projects, but it also forces you to think through the entire system architecture. This ensures you go through every possible roadblock or challenge you might face, thereby exposing all gaps in your thinking.

5.2 System Architecture Design

5.2.1 Chosen System Architecture

The system is defined using a component diagram and a data flow diagram. The data flow diagram is attached at the end of the document. The different components of this system are as follows:
1. Data Exploration and Pre-processing - This allows the engineers to select the data and alter it according to the requirements to extract the highest amount of information possible.
2. Training Algorithm - This step allows the engineers to select the most efficient algorithm for machine learning and sentimental analysis to give the most accurate results.
3. Deploying using Flask - This allows the engineers to deploy the Machine Learning and Natural Language Processing models on the web.
4. Testing for ML- This step allows the engineers to validate on the designed algorithm and perform the tests to tune the hyperparameters to achieve the metrics required for output.
5. Testing for Flask - This allows the engineers to check if all the components of their website are functioning properly.

5.2.2 System Interface Description

The Product will be made on Google Collab and python. The system will be able to run on Windows and Linux OS Platforms. The system will run on a Flask Framework having an interface which makes it easy for the user to perform different actions.

5.3 Detailed Description of Components

5.3.1 Exploring and Preprocessing Datasets

Responsibility - Neelay Jagani
Constraints - Dataset should be neither too small nor be too small. Composition - Dataset should be balanced i.e. it should have enough ups and downs in the trends and enough positive and negative tweets
Interactions - Client server interaction using python GUI
Resources - TwitterAPI and Yahoo Finance API

5.3.2 Training Algorithm

Responsibility - Jash Shah
Constraints - Select the algorithm that is best suited for the dataset chosen as well as the one that gives the targeted metric combination.
Composition - The engineer performs trials on various algorithms and selects the most suitable.
Interactions - Client server interaction using python.
Resources - Google Collab and Stack Overflow .

5.3.3 Testing and Deployment

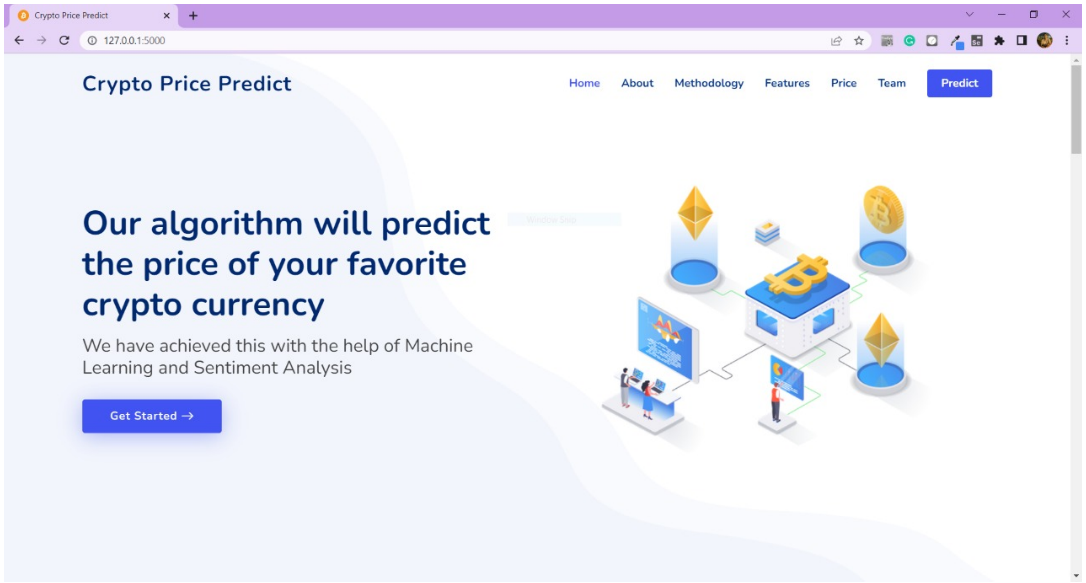
Responsibility - Harsh Sachala

Constraints - The algorithm should be validated with all the test cases to perform the hyperparameter tuning. And the final testing should be done using the test set. The website deployed using Flask also should be tested for appropriate functioning.

Interactions - Client server interaction using Flask

Resources - Test Set,Flask,Selenium,GT-Metrix and Google Collab.

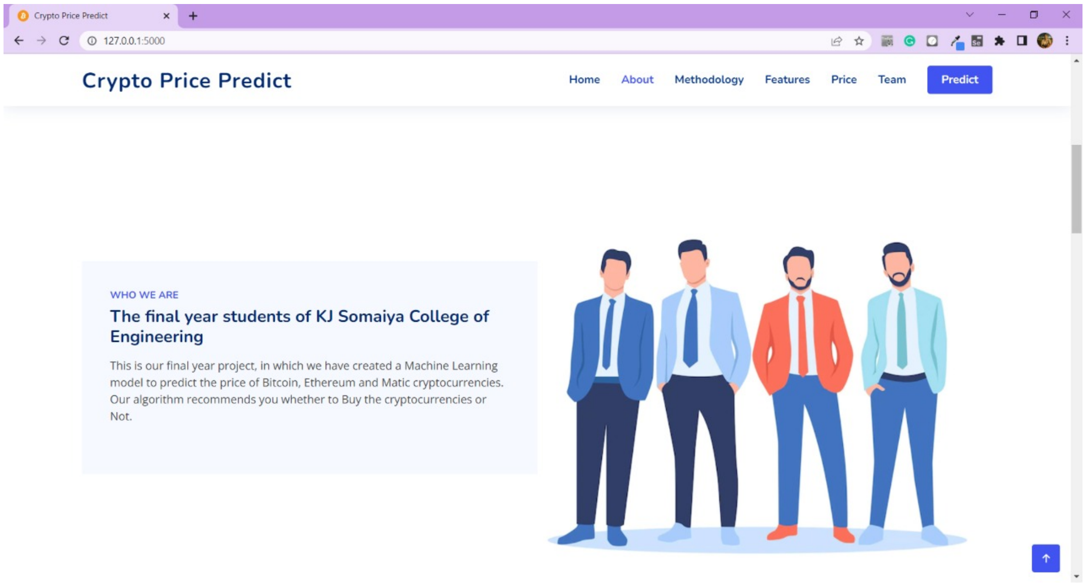
5.4 System User Interface



Our algorithm will predict the price of your favorite crypto currency

We have achieved this with the help of Machine Learning and Sentiment Analysis

Figure 5.1: Home Page

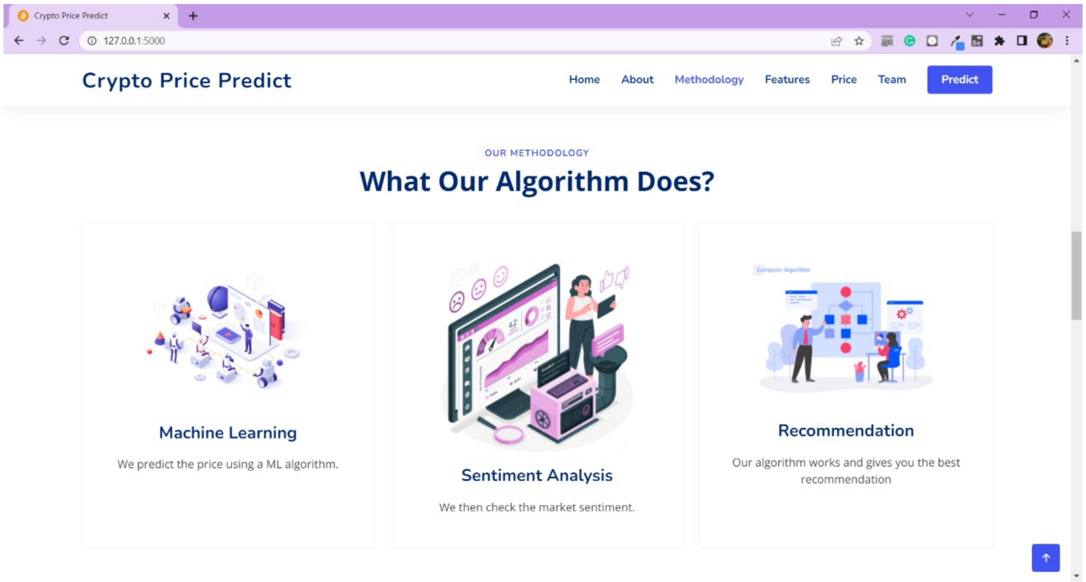


WHO WE ARE

The final year students of KJ Somaiya College of Engineering

This is our final year project, in which we have created a Machine Learning model to predict the price of Bitcoin, Ethereum and Matic cryptocurrencies. Our algorithm recommends you whether to Buy the cryptocurrencies or Not.

Figure 5.2: About section



OUR METHODOLOGY

What Our Algorithm Does?

Machine Learning

We predict the price using a ML algorithm.

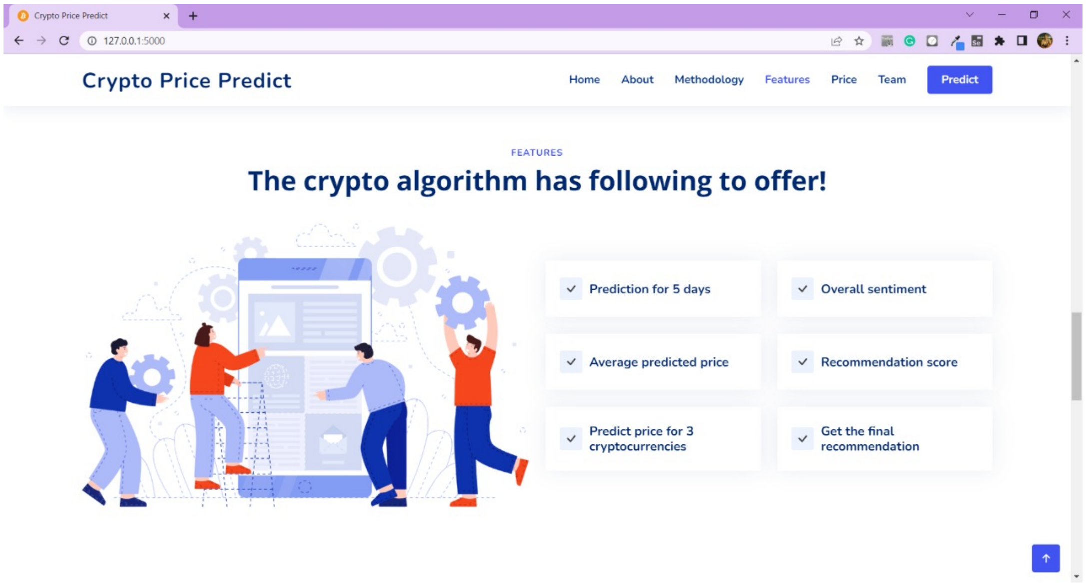
Sentiment Analysis

We then check the market sentiment.

Recommendation

Our algorithm works and gives you the best recommendation

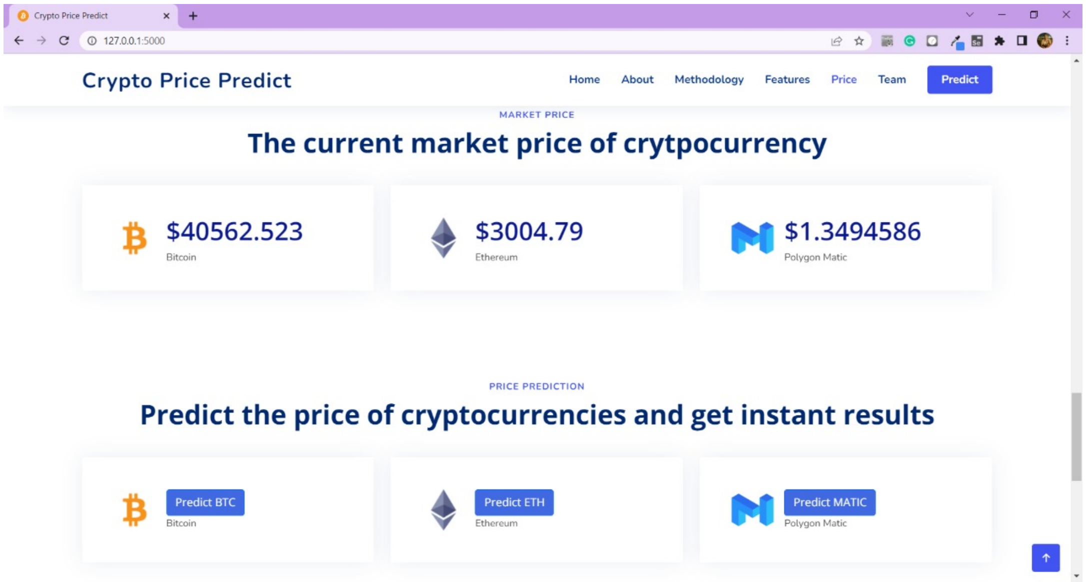
Figure 5.3: Methodology section



The crypto algorithm has following to offer!

- ✓ Prediction for 5 days
- ✓ Overall sentiment
- ✓ Average predicted price
- ✓ Recommendation score
- ✓ Predict price for 3 cryptocurrencies
- ✓ Get the final recommendation

Figure 5.4: Features section

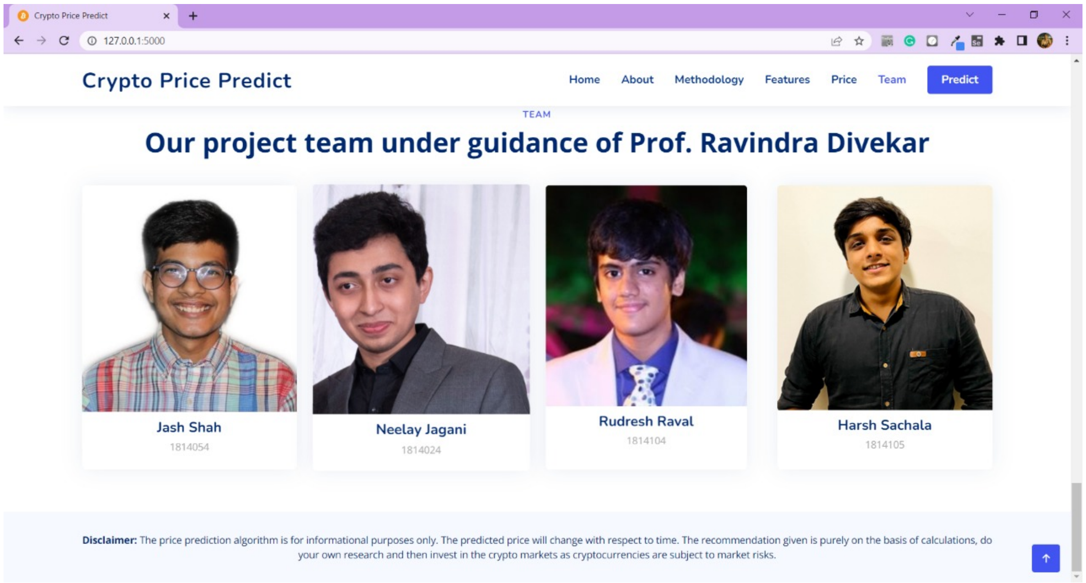


The current market price of crytpocurrency

$40562.523 Bitcoin	$3004.79 Ethereum	$1.3494586 Polygon Matic

Predict the price of cryptocurrencies and get instant results

Figure 5.5: Prices and Predict section



TEAM

Our project team under guidance of Prof. Ravindra Divekar

Jash Shah
1814054

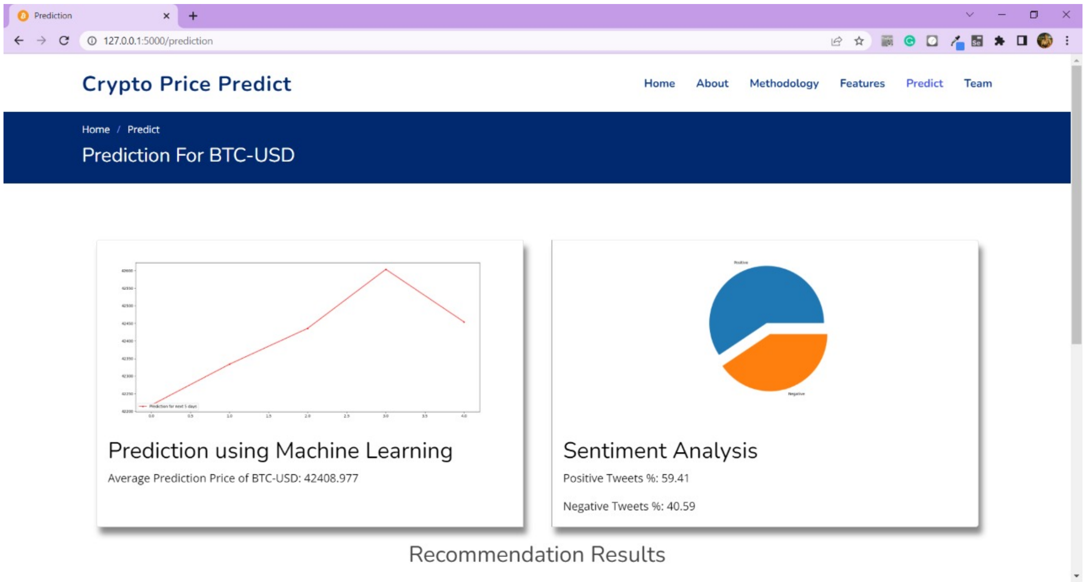
Neelay Jagani
1814024

Rudresh Raval
1814104

Harsh Sachala
1814105

Disclaimer: The price prediction algorithm is for informational purposes only. The predicted price will change with respect to time. The recommendation given is purely on the basis of calculations, do your own research and then invest in the crypto markets as cryptocurrencies are subject to market risks.

Figure 5.6: Team section



Prediction For BTC-USD

Prediction using Machine Learning

Average Prediction Price of BTC-USD: 42408.977

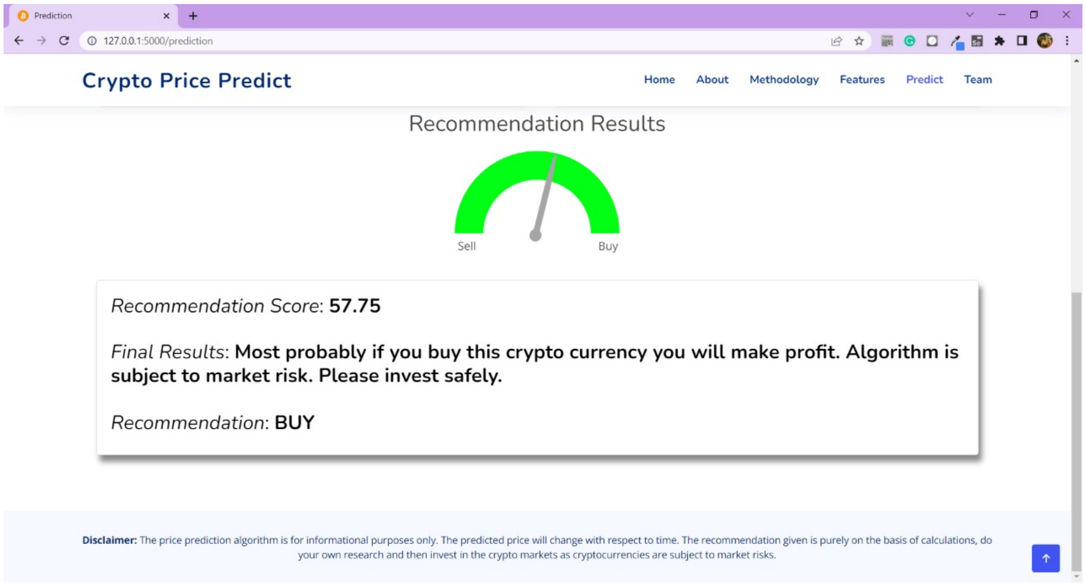
Sentiment Analysis

Positive Tweets %: 59.41

Negative Tweets %: 40.59

Recommendation Results

Figure 5.7: Result Page with graphs



Recommendation Results

Recommendation Score: **57.75**

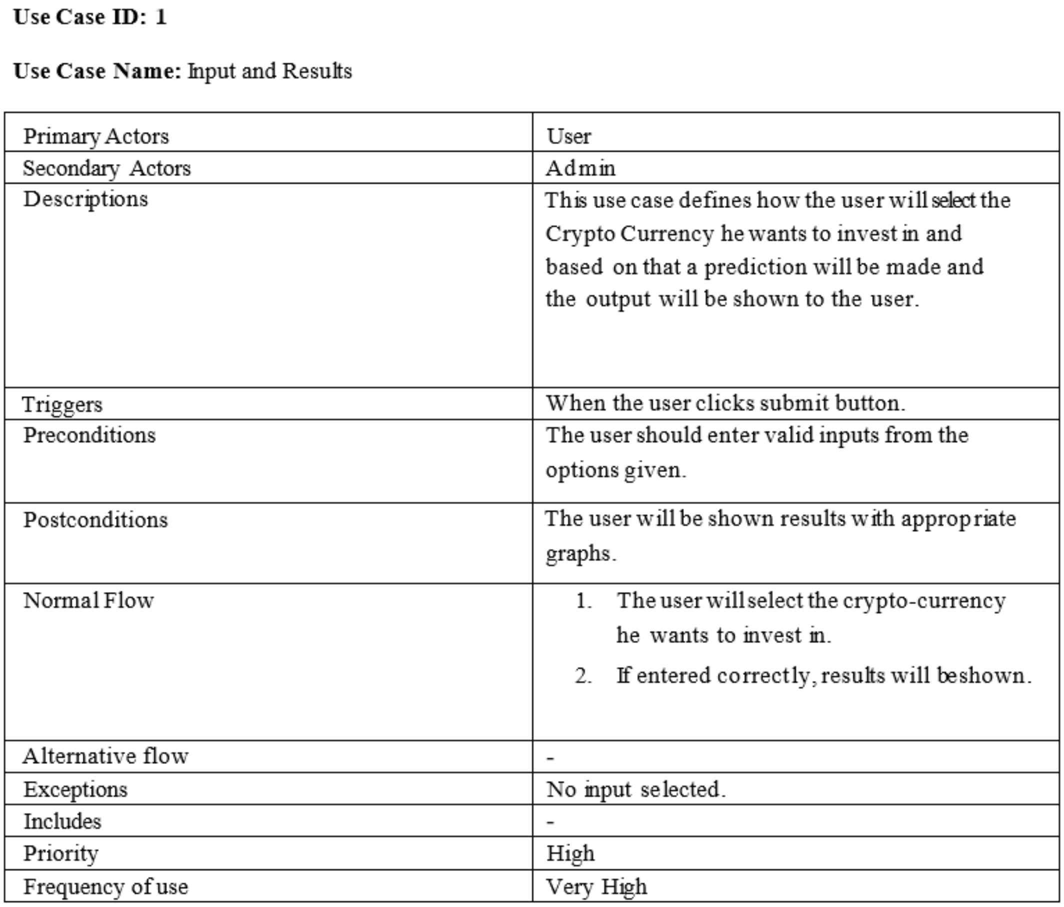
Final Results: **Most probably if you buy this crypto currency you will make profit. Algorithm is subject to market risk. Please invest safely.**

Recommendation: **BUY**

Disclaimer: The price prediction algorithm is for informational purposes only. The predicted price will change with respect to time. The recommendation given is purely on the basis of calculations, do your own research and then invest in the crypto markets as cryptocurrencies are subject to market risks.

Figure 5.8: Result Page with Gauge and Prediction Score

5.5 System Architecture

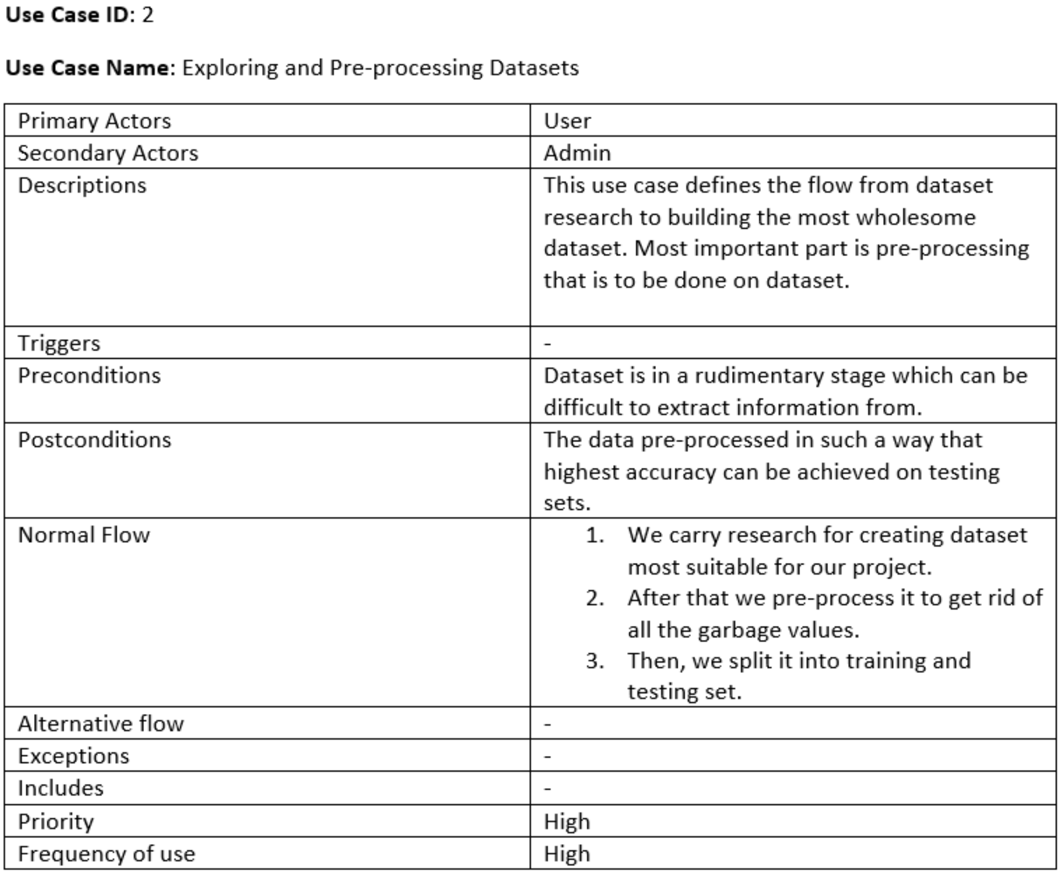


Use Case ID: 1

Use Case Name: Input and Results

Primary Actors	User
Secondary Actors	Admin
Descriptions	This use case defines how the user will select the Crypto Currency he wants to invest in and based on that a prediction will be made and the output will be shown to the user.
Triggers	When the user clicks submit button.
Preconditions	The user should enter valid inputs from the options given.
Postconditions	The user will be shown results with appropriate graphs.
Normal Flow	1. The user will select the crypto-currency he wants to invest in. 2. If entered correctly, results will beshown.
Alternative flow	-
Exceptions	No input selected.
Includes	-
Priority	High
Frequency of use	Very High

Figure 5.9: Use Case 1



Use Case ID: 2

Use Case Name: Exploring and Pre-processing Datasets

Primary Actors	User
Secondary Actors	Admin
Descriptions	This use case defines the flow from dataset research to building the most wholesome dataset. Most important part is pre-processing that is to be done on dataset.
Triggers	-
Preconditions	Dataset is in a rudimentary stage which can be difficult to extract information from.
Postconditions	The data pre-processed in such a way that highest accuracy can be achieved on testing sets.
Normal Flow	1. We carry research for creating dataset most suitable for our project. 2. After that we pre-process it to get rid of all the garbage values. 3. Then, we split it into training and testing set.
Alternative flow	-
Exceptions	-
Includes	-
Priority	High
Frequency of use	High

Figure 5.10: Use Case 2

5.6 Data Flow Specifications

DFD Level0

A context diagram is a top level (also known as ”Level 0”) data flow diagram. It only contains one process node (”Process 0”) that generalizes the function of the entire system in relationship to external entities. Draw data flow diagrams can be made in several nested layers. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.

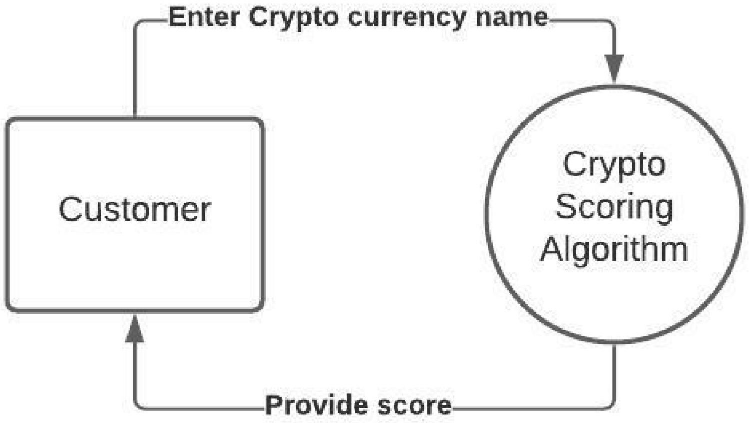


Figure 5.11: DFD Level 0

DFD Level1

In the level 1 DFD, the processes are further broken down and the flow can be understood in further detail. The database can also be seen in the level 1 DFD and the we a better idea about the processes regarding the Customer and Crypto Scoring algorithm entities.

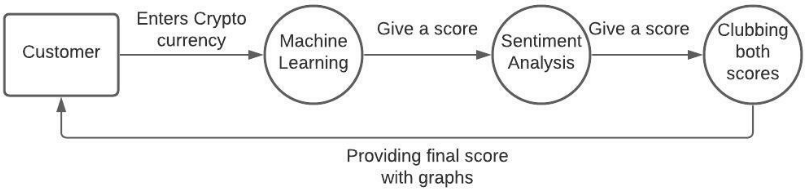


Figure 5.12: DFD Level 1

Chapter 6

Implementation

This chapter presents an overview about tech stack used for building the project which is followed by explanation of the methodology

6.1 Technologies Used

6.1.1 Python

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.

6.1.2 HTML5

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and final major HTML version that is a World Wide Web Consortium recommendation. The current specification is known as the HTML Living Standard.

6.1.3 CSS3

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, development of various parts of CSS specification was done synchronously, which allowed versioning of the latest recommendations.

6.1.4 Bootstrap v5.1

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

6.1.5 Heroku

Heroku is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. The platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

Heroku is fully managed, giving developers the freedom to focus on their core product without the distraction of maintaining servers, hardware, or infrastructure. The Heroku experience provides services, tools, workflows, and polyglot support—all designed to enhance developer productivity.

6.1.6 Selenium

Selenium is an open source umbrella project for a range of tools and libraries aimed at supporting browser automation. It provides a playback tool for authoring functional tests without the need to learn a test scripting language

6.1.7 GTMetrix

GTMetrix is a tool that analyzes the page speed and performance. It generates scores for our pages and offers Pagespeed and WiSlow. It is a free tool that we have found to be used to get detailed reports about site performance. It creates scores for the pages and also provides recommendations to fix them.

6.2 Algorithm

6.2.1 Extraction and Processing

Data Processing for LSTM

As there is much fluctuation in the cryptocurrency prices, the price for every hour was taken into account. Then, lag plots for various time frames like hourly, daily, weekly and monthly lag were plotted, and the daily lag plot turned out to be the best of them. Then, a dataset comprising the mean of all prices for one day was created; that is, every row had the mean price of the cryptocurrency for that day. All operations were performed only on the closing price of a cryptocurrency. Then, a MinMaxScaler was used to normalize all the values. Finally, a lookback period was defined to convert the dataset into a time series. A “lookback period” defines the window size of how many previous timesteps are used in order to predict the subsequent timestep. The timestep was set to five, which means that the value of every sixth day was predicted using the previous five days, and thus, a time series was created. After this, the dataset was divided into training and testing datasets, where the testing period was considered twenty-five days.

Data Processing for Sentimental Analysis

The most recent tweets containing the word bitcoin were fetched. Tweets retrieved that were already retweeted versions of other tweets were not fetched. The data retrieved was with much noise. Hence cleaning of data was required where operations such as dropping usernames (mentions), punctuation, link, and stop words were performed. An inbuilt list of stop words was used from the library nltk.corpus. The cleaned text was tokenized, and stemming was performed. Porter Stemmer algorithm was used for stemming the words to their root words. Duplicate clean tweets were also dropped. Thus, a dataset of all the clean tweets was prepared for sentimental analysis.

6.2.2 LSTM

For a variety of learning issues involving sequential input, recurrent neural networks (RNNs) with long short-term memory (LSTM) have emerged as an effective and scalable technique. They are crucial for catching long-term temporal dependencies since they are broad and pragmatic. The LSTM is an RNN-inspired model with information-flow control gates between cells. The input and forget gate structures can modify info as it passes through the cell state, with the ultimate output being a processed version of the cell state based on the context of the inputs.

The data was preprocessed, and lag plots were used to find which lag would be suitable for the model. The data were then grouped daily, and the closing price of each day was selected. The data was then normalized, split into train and test, and then the dataset for LSTM was generated and reshaped into a 3D tensor. The LSTM model used is a sequential one, having four layers and one dense layer. The number of units in the first layer is set to 128, and layer second and third contain 64 units each, and the final layer has 32 units. Each layer has a dropout regularisation set to 20% in each layer. The activation function used is 'relu.' The model was compiled using adam optimizer, and loss was set to mean squared error, and value loss was monitored for the maximum number of epochs.

6.2.3 Sentiment Analysis

As mentioned, the Twitter API named tweepy was used to retrieve the tweets.The dataset of retrieved tweets was then evaluated using TextBlob to provide the sentiment prevalent in the market regarding bitcoin. TextBlob uses the ”averaging” technique to calculate a polarity score for a single word, and thus a similar procedure applies to every single word, resulting in a combined polarity for more extensive texts. A pie chart was plotted, reflecting the percentage of positive tweets to negative tweets. A maximum of 100 tweets could be retrieved at a time because of security protocols set up by Twitter in its Tweepy version 4.8.0.

Required developer keys are hardcoded into the code, including consumer key, consumer secret, access token, and access token secret, which allow for safe use of the API. The tweets retrieved are pre-processed and passed into TextBlob for polarity calculation. TextBlob results in a polarity score, where a score higher than zero represents a positive tweet, a negative score represents a negative tweet, and a score equal to zero represents a neutral tweet. As for the research, a clear distinction of recommendation between ”buy” and ”sell” was required; the neutral tweet percentage was divided equally and added to the positive tweets percentage and negative tweets percentage. When the positive tweet percentage exceeded the negative tweet percentage, a ”buy” recommendation was proposed, indicating a positive outlook. Whereas whenever the negative tweet percentage exceeded the positive tweet percentage, a ”sell” recommendation was proposed, indicating a negative outlook

6.2.4 Combined Approach

The combination of Machine Learning and Sentimental Analysis is used to give a final recommendation on what an investor should do, whether to ”buy” or ”sell.” The prediction given by the LSTM model is then coupled with the sentiment of the cryptocurrency market at that time to conclude whether to buy or not buy. If the LSTM Model predicts the average bitcoin price over the next five days will be greater than the current market price (Buy recommendation from LSTM Model) and the sentimental analysis sees the percentage of positive tweets is greater than the percentage of negative tweets (Buy recommendation from Sentimental Analysis), a final recommendation of buy is given to the investor. Similarly, if the LSTM Model predicts the average bitcoin price over the next five days will be less than the current market price (Sell recommendation from LSTM Model) and the sentimental analysis sees the percentage of negative tweets is greater than the percentage of positive tweets (Sell recommendation from Sentimental Analysis), a final recommendation of Sell is given to the investor. A dilemma is created when there are conflicting opinions where the LSTM model recommends a buy and the Sentiment Model recommends a Sell or vice versa. A quantitative weighted average strategy is employed where both the models are assigned certain weights—the formula is given below. To find the weights, a simulation at the increments of 10 is done for days when there are conflicting recommendations. Starting with sentiment getting a weightage of 90 and LSTM getting a weightage of 10 to sentiment getting a weightage of 10 and LSTM getting a weightage of 90. A clear threshold is seen in three scenarios with 70:30, 80:20, and 90:10 between Machine Learning weightage to Sentimental Analysis weightage. A threshold of 0.078, 0.095, and 0.117 was seen. To pick a single scenario, a correlation test was done with the predicted recommendation from all three different weights and the actual scenario that transpired. It was seen that a weightage of 80:20 had the highest correlation of prediction classification with the actual scenario.

The formula used for the calculation is:

$$((Returns\ Predicted\,Using\ ML\ Algorithm{*}ML\ Weight)+(Positive\ Tweets\ Percentage{*}\ Sentiment\ Analysis\,Weight))/100 \tag{6.1}$$

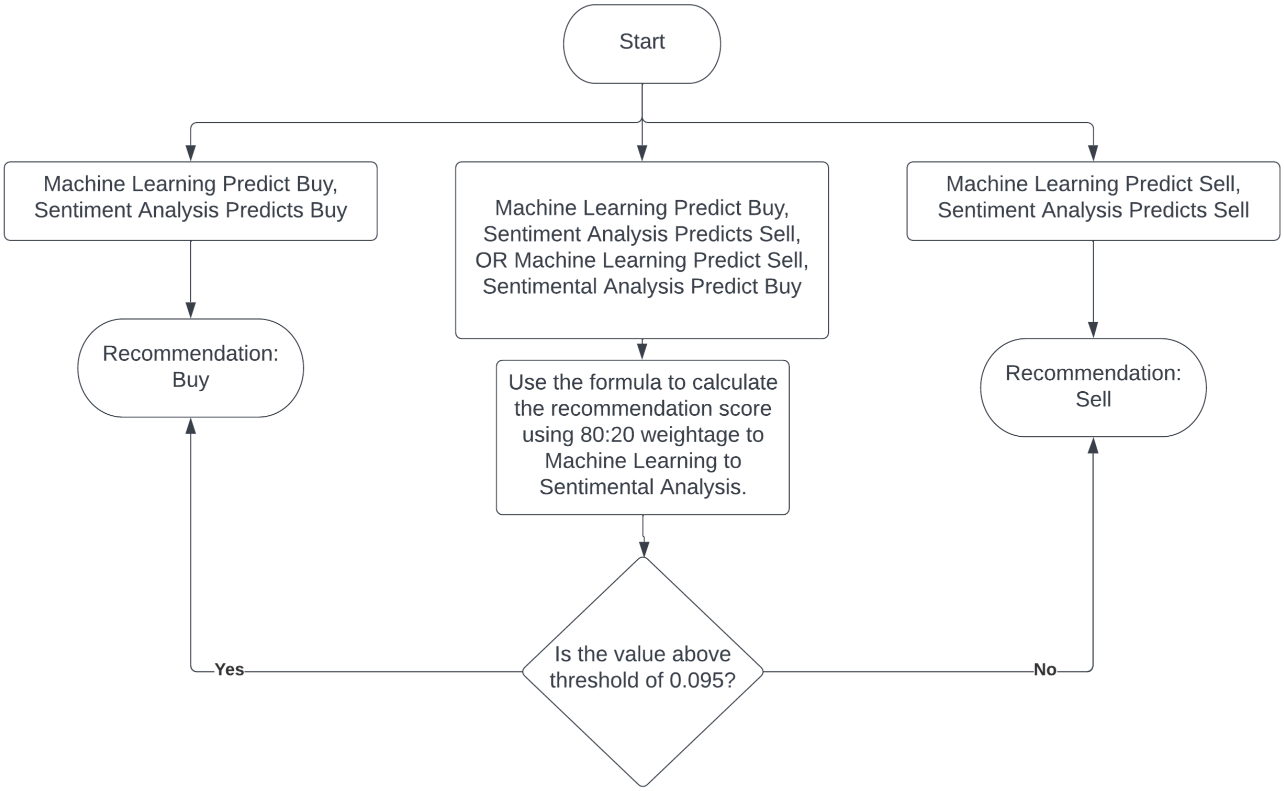

Start
Machine Learning Predict Buy, Sentiment Analysis Predicts Buy
Machine Learning Predict Buy, Sentiment Analysis Predicts Sell, OR Machine Learning Predict Sell, Sentimental Analysis Predict Buy
Machine Learning Predict Sell, Sentiment Analysis Predicts Sell
Recommendation: Buy
Use the formula to calculate the recommendation score using 80:20 weightage to Machine Learning to Sentimental Analysis.
Recommendation: Sell
Yes
Is the value above threshold of 0.095?
No

Figure 6.1: Flowchart for combined approach

Chapter 7

Software Test Document

This chapter provides an overview of the project as well as the testing approach that was used for the project. The testing process was then carried out according to a well-defined test plan. For the chapter's conclusion, a full description of the individual test cases has been written.

7.1 Introduction

7.1.1 System Overview

The field of cryptocurrency is at peak and its highly volatile and price changes happens very frequently and drastically, so it becomes very difficult to analyse each and every aspect of the cryptocurrency into account so as to make an informed decision whether to invest in a particular cryptocurrency or not.

We aim to club machine learning with sentimental analysis to predict the price of cryptocurrency which we believe can help us achieve a greater, a more reliable prediction on whether to buy, not buy a particular cryptocurrency. We plan to assign weight as to how much market sentiment should be given importance relative to our machine learning prediction so as to achieve maximum reliability.

This document is proposed to define the test plan and strategy to test the Cryptocurrency prediction Model. It describes the overall approach to testing for each major group of features or feature combinations , the major activities, techniques, and tools which will be used to test the designated groups of features.

7.1.2 Test Approach

A test approach is the test strategy implementation of a project, defines how testing would be carried out. This section defines the overall test approach to be taken for the project. We will focus on the below two types of testing :

- Automation testing: It is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it's a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development.
- Performance Testing : It is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size.

7.2 Test Plan

7.2.1 Test Plan Objectives

- Testing is the process of executing the program the program with the intention of finding an error.
- A good test is one that has high probability of finding an as yet undiscovered error/bug.
- A successful test is that which uncovers as-yet-undiscovered error.

7.2.2 Features to be Tested

1. Landing page
    - Testing on overall responsiveness (performance testing via GTMetrix)
2. Result display
    - Proper display of graphs and desired output on selection
    - Unexpected results displayed

7.2.3 Testing Tools and Environment

Testing Tools

1. Selenium
2. GTMetrix

Environment

Hardware Configuration:

1. 8 GB Ram
2. 120 GB Hard Disk
3. 2,6 GHZ Processor

Software Configuration:

1. Google Chrome
2. Jupyter Notebook
3. Microsoft Visual Studio Code
4. Microsoft Edge

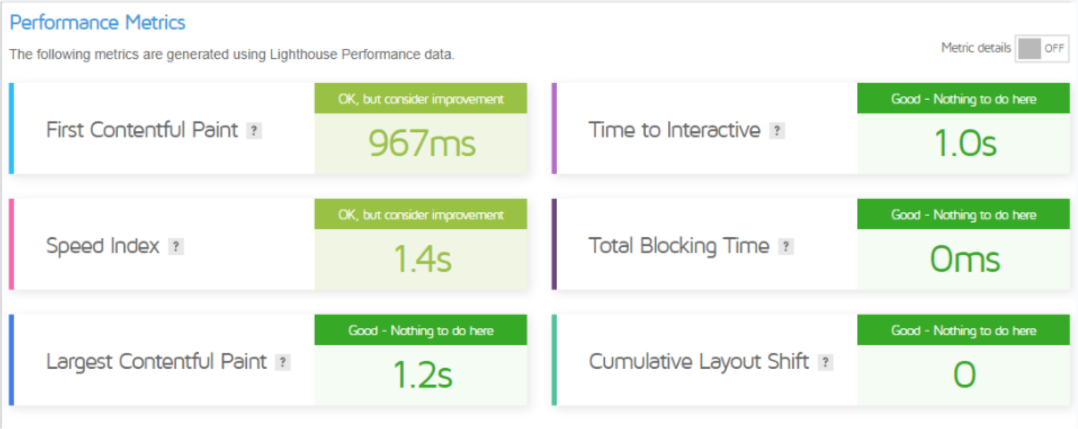
7.3 Test Cases

7.3.1 Automation Testing of the Result display using Selenium

Test case : GUI Display Page				
Test Case	Description	Input	Expected Output	Actual Output
Display 1	Accurate graphs and outputs	Click on desired cryptocurrency	Proper graphs and numerical outputs displayed	Same as expected output
Display 2	Unexpected output displayed	Click on desired cryptocurrency	”Refresh and try again, please try again”	Same as expected output

7.3.2 Performance Testing of the Overall Website

Many various factors like Interactive Time, TTLB, Speed Index, Total Time block etc were measured during the testing using GTMetrix in order to ensure the overall responsiveness of the website.



Performance Metrics

The following metrics are generated using Lighthouse Performance data.

Metric details OFF

First Contentful Paint ? — OK, but consider improvement — 967ms

Time to Interactive ? — Good - Nothing to do here — 1.0s

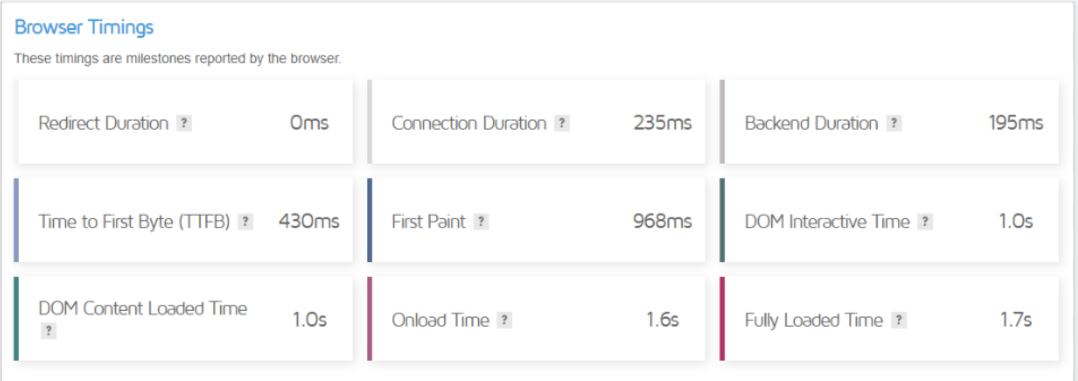
Speed Index ? — OK, but consider improvement — 1.4s

Total Blocking Time ? — Good - Nothing to do here — 0ms

Largest Contentful Paint ? — Good - Nothing to do here — 1.2s

Cumulative Layout Shift ? — Good - Nothing to do here — 0

Figure 7.1: Performance Metrics



Browser Timings

These timings are milestones reported by the browser.

Redirect Duration ? 0ms

Connection Duration ? 235ms

Backend Duration ? 195ms

Time to First Byte (TTFB) ? 430ms

First Paint ? 968ms

DOM Interactive Time ? 1.0s

DOM Content Loaded Time ? 1.0s

Onload Time ? 1.6s

Fully Loaded Time ? 1.7s

Figure 7.2: Browser Timings

Chapter 8

Conclusions and scope for further work

In a summary, this chapter summarizes the project's conclusion and offers various suggestions for project enhancement.

8.1 Conclusion

We have built an algorithm for investing in a particular cryptocurrency and have made the software documentation for the same. The application satisfies the requirements mentioned in the SRS document. The user interface is simple and can be used by any naïve user. No technical knowledge is required to use the application. The risks associated with the project have been mentioned in the RMMM plan. We were able to take care of these risks while completing the project. However, we understand that some of the risk mitigation techniques need cooperation from the side of the client and those risks have been mentioned appropriately in the RMMM plan. Agile process model was followed to make sure that the changes needed are incorporated in successive releases.

8.2 Future scope

A few features can be added to the existing application.

- The following algorithm can be extended to many other cryptocurrencies which are in talks and whose prices are affected as per user's sentiments.

- It can be extended in a way to ask a user how much they are willing to invest and possible loss/return.

Bibliography

[1] Jethin Abraham, Daniel Higdon, John Nelson, and Juan Ibarra. Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review*, 1(3):1, 2018.

[2] Laura Alessandretti, Abeer ElBahrawy, Luca Maria Aiello, and Andrea Baronchelli. Anticipating cryptocurrency prices using machine learning. *Complexity*, 2018, 2018.

[3] Aytug Altin. *An in-depth analysis on the impact of Twitter features on bitcoin's price prediction.* PhD thesis, 02 2022.

[4] Elie Bouri, Chi Keung Lau, Tareq Saeed, Shixuan Wang, and Yuqian Zhao. On the intraday return curves of bitcoin: Predictability and trading opportunities. *International Review of Financial Analysis*, 76:101784, 04 2021.

[5] Sudhansu Das and Dipak Das. *Sentiment Analysis for Prediction of Bitcoin Response*, pages 193–206. 01 2022.

[6] Vasily Derbentsev, Natalia Datsenko, Olga Stepanenko, and Vitaly Bezkorovainyi. Forecasting cryptocurrency prices time series using machine learning approach. In *SHS Web of Conferences*, volume 65, page 02001. EDP Sciences, 2019.

[7] Fan Fang, Waichung Chung, Carmine Ventre, Michail Basios, Leslie Kanthan, Lingbo Li, and Fan Wu. Ascertaining price formation in cryptocurrency markets with machine learning. *The European Journal of Finance*, pages 1–23, 2021.

[8] Samuel Gyamerah. Two-stage hybrid machine learning model for high-frequency intraday bitcoin price prediction based on technical indicators, variational mode decomposition, and support vector regression. *Complexity*, 2021:1–15, 12 2021.

[9] Mohammad J. Hamayel and Amani Yousef Owda. A novel cryptocurrency price prediction model using gru, lstm and bi-lstm machine learning algorithms. *AI*, 2(4):477–496, 2021.

[10] Patrick Jaquart, David Dann, and Christof Weinhardt. Short-term bitcoin market prediction via machine learning. *The journal of finance and data science*, 7:45–66, 2021.

[11] Ahmed M Khedr, Ifra Arif, Magdi El-Bannany, Saadat M Alhashmi, and Meenu Sreedharan. Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. *Intelligent Systems in Accounting, Finance and Management*, 28(1):3–34, 2021.

[12] Thomas E Koker and Dimitrios Koutmos. Cryptocurrency trading using machine learning. *Journal of Risk and Financial Management*, 13(8):178, 2020.

[13] Salim Lahmiri and Stelios Bekiros. Deep learning forecasting in cryptocurrency high-frequency trading. *Cognitive Computation*, 13(2):485–487, 2021.

[14] Keshav Lengare. Introduction to cryptocurrencies. 12:01–08, 03 2022.

[15] Samiksha Marne. Predicting price of cryptocurrency-a deep learning approach. 2021.

[16] Krzysztof Wołk. Advanced social media sentiment analysis for short-term cryptocurrency price prediction. *Expert Systems*, 37(2):e12493, 2020.

Acknowledgements

The completion of this project **Cryptocurrency Investment Scorer** would not have been possible without the help and cooperation of many. We would like to thank all those who helped us directly and indirectly in the completion of this project work. First and foremost, we would like to express our gratitude to our college **K.J. Somaiya College Of Engineering**, for providing us this opportunity to work on this project and upgrade our skills. We would like to express our gratitude to our guide **Prof. Ravindra Divekar**, Department of Information Technology, for providing excellent guidance, encouragement, inspiration, constant and timely support throughout this project.