Cryptocurrency Forecasting Model

A Project Report
Presented to
The Faculty of the Computer Engineering Department

San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree
Bachelor of Science in Software Engineering

By

Berlun Devera
Deven Shah
Duc Tran
Brandon Truong

12/2022

APPROVED FOR THE COLLEGE OF ENGINEERING

Young Park

Dr. Younghee Park, Project Advisor

Dr. Wencen Wu, Instructor

Dr. Rod Fatoohi, Computer Engineering Department Chair

ABSTRACT

Cryptocurrency Forecasting Model

By Berlun Devera, Deven Shah, Duc Tran, Brandon Truong

The cryptocurrency exchange domain is a relatively volatile space. The most widely traded cryptocurrency coin Bitcoin has experienced a high of $44,533.00 and a low of $36,259.01 in the week of 1/31/22 - 2/7/22. The volatility of the cryptocurrency market stems from three accepted analyses. A technical analysis solely relies on metrics ranging from historical trends to net unrealized profit/loss to derive the effects of price movements. A fundamental analysis relies on factors that affect price movements, such as government policies. A sentimental analysis relies on the sentiment of a coin at a particular time, which can be identified using social media trends.

Given the abundance of variables that affect price movements, forecasting even near-future prices prove difficult for many traders. Each of the three analyses stated (technical, fundamental, and sentimental) have sub-analyses that would take an abundance of time even for the experienced trader. As the digital asset market increased exponentially over the past 2 years, many traders are not accustomed to these analyses, much less able to derive conclusions from them.

The cryptocurrency forecasting model aimed to traverse, analyze, and interpret data from the three types of analyses with a greater focus on technical and sentimental analysis. Using the data interpreted, the model has the ability to forecast price movements to the time scale of the customer's preference. This project reduced the time spent significantly analyzing technical data, assisted traders to make confident trading decisions, and detailed the price movement patterns that are difficult to infer with purely human capabilities

Acknowledgments

We would like to thank all of the professors that have helped guide us to the position that we are in today. We would not have been able to be able to complete this project without their knowledge and wisdom that they have shared with us. We would like to specifically thank Dr. Younghee Park, our project advisor for following our journey and leading us towards the right path when creating this project. A lot of helpful feedback was provided and she was always making sure that we were making our deadlines. Every meeting she would have all of us present something that we have accomplished since the previous meeting which ensured that everyone was doing their part and that we would be able to finish the project in time. She also provided a lot of helpful feedback on not only the idea of how to accomplish our project, but also on our assignments as well such as the drafts for this project report. We would also like to thank Dr. Wencen Wu for being the instructor of this class and all of the Engineering Faculty and Staff for providing us with this opportunity to show everything that we have learned at San Jose State University.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Project Goals and Objectives

One of the objectives of this project was to be able to forecast cryptocurrency prices using historical data and social media sentiment. In this case, the historical data of cryptocurrency prices was the collected data of a cryptocurrency's metrics such as return on investment (ROI), volatility, open-high-low-close (OHLC) prices, and market capitalization. Return on investment is the measurement of the amount of return on a cryptocurrency investment defined by the amount of profit divided by its cost. Volatility is the measurement of the price fluctuations of a cryptocurrency. A higher volatility on a cryptocurrency means that the range of its value can be spread out whereas a lower volatility on a cryptocurrency means that the price will be much more stable. Open-high-low-close prices is a bar graph that shows the opening and closing prices of a cryptocurrency as well as the highest and lowest prices during a particular period of time such as hourly or daily periods. The market capitalization of a cryptocurrency is the total market value of the circulating supply. Bitcoin always had the highest market capitalization, followed by Ethereum. In terms of cryptocurrency, social media sentiment is the attitude and feelings that people have about cryptocurrency. A positive sentiment towards a cryptocurrency means that people are expected to buy more of that cryptocurrency whereas a negative sentiment means that people are expected to avoid buying that cryptocurrency.

Another objective of this project was to determine the performance of prediction models. The prediction models that were used were neural networks and KNN (k-nearest neighbors). The KNN model was used to generate price directional predictions, which were then later used in the neural network model.

The last objective of this project was to provide the results of the forecasted cryptocurrency prices and the performance of the prediction models in a user-friendly interface. Users were able to view the forecasted cryptocurrency prices in different time intervals in order to make informed decisions on what to do with their trading portfolio.

1.2 Problem and Motivation

A problem that was addressed was how average users may be unable to obtain conclusions from technical and sentimental analysis. It was understandable that technical and sentimental analyses required the user to do extensive research and redundant work. Users who want to trade cryptocurrencies may not have the time or the energy to do that research. To mitigate this problem, the project allowed users to understand the technical and sentimental analyses of cryptocurrency prices without the busy work. Furthermore, the model accommodated a wider customer base for the cryptocurrency, in which average and experienced users alike participated in trading their most favorable cryptocurrency.

1.3 Project Application and Impact

This project met the specified needs of many different people whether it be people of different countries, people who are socially/environmentally disadvantaged, or just people who are looking for help in their economics. The website informed people about cryptocurrency dynamics and allowed every user to be able to have the opportunity to see future prices of a set of coins. Investing in cryptocurrency is a great way in securing a financially stable future and it is aimed to make it a safe investment opportunity for everyone. While the website generated an increased overhead because a lot of companies would have paid people to create a program like this, it was free because the more educated and informed people are about cryptocurrency the better it will be for every investor economically. The reason why it will be better for investors is because cryptocurrencies are decentralized, meaning that no bank or government can put a price cap on these coins. That is why one Bitcoin can be worth such a high amount of money (Filippi and Loveluck, 2016). If more people know how to invest in cryptocurrency safely, then there will be more people investing in the coins which will naturally increase the price. The good part about cryptocurrency being decentralized is that it could be used all around the world. So, people from different countries could invest in some cryptocurrencies and increase the prices even though they are living in another part of the world. The model did not limit the people who are at a disadvantage whether economically or environmentally because everyone deserves a fair chance on using the model. The main goal of the project was to create more traction and increase the number of users who want to safely invest in these coins.

According to the World Bank “Remittance Flows Register Robust 7.3 Percent Growth in 2021”, 589 billion dollars of remittance flow to low and middle-income countries in 2021. That amount of money was subject to paying an average fee of approximately 6.5% (“Remittance Flows”, 2021). That means 38 billion dollars was lost from paying the fees. Compared to cryptocurrency, the fee is expected to be less than 1%. This advantage of cryptocurrency will save a lot of money, and this money can be used for various projects to boost the economy from those low and middle-income countries. As the project attracts more investors in cryptocurrency, more transactions in cryptocurrency are made, so more money is saved for the economy.

Income inequality is a big problem in human society. A small group of people possess a large amount of wealth while a majority of people have to work hard but barely enough for their livings. However, things started to change. Nowadays, smartphones and the internet are more popular. Anyone with a phone and internet access can open a digital wallet to start investing and building a portfolio without having to go through the complex financial system. A study by Abdullah Othman et al. (2020) finds that “increasing the supply of cryptocurrencies in global circulation will lead to equal income and wealth distribution”. The young generation seems to get the most benefit from this opportunity. They pay attention and adopt the concept of the cryptocurrency better than other generations. Because of this reason, income inequality is believed to be fixed gradually, and the wealth will start transferring to this young generation.

1.4 Project Results and Deliverables

The project included a project report that documents the approach on the implementation of the cryptocurrency forecasting model. Furthermore, data from different social media platforms were collected to create a social media sentiment analysis. A data pre-processing model was used to filter out noise elements that may skew the model. Then, two models for technical and sentimental analyses were created to forecast the future cryptocurrency prices. For the technical analysis, two machine learning models were used. The KNN model was used to derive the directional predictions, which then was used in the neural network model to generate price predictions. From here, a performance model was implemented that measured the RMSE, actual vs. predicted values, prediction accuracy, absolute error (difference of actual and predicted), and relative error (absolute error divided by actual value) to return a weighted score measuring the effectiveness of the technical analysis models. After the performance model was created, the results will be presented on a clear GUI.

There are a multitude of deliverables that will aid in implementing the forecasting model, which includes:

Table 1: List of Deliverables

Deliverable No.	Deliverable Name	Owner(s)
D1	Technical Data Preprocessing Model	Deven Shah, Duc Tran
D2	Sentimental Data Preprocessing Model	Berlun Devera, Duc Tran
D3	KNN Model	Deven Shah
D4	Neural Network Model	Deven Shah
D5	Technical Data Analysis	Deven Shah
D6	Sentimental Forecasting	Berlun Devera
D7	Sentimental Data Analysis	Berlun Devera
D8	GUI	Brandon Truong, Duc Tran

1.5 Project Report Structure

The following report documents the full life cycle of this project, from background research to implementation. Chapter 2 lists the high-level components used, details the information gained from research and outlines the state-of-the-art technologies implemented in the project. Chapter 3 establishes the business, system functional, non-functional, contextual/interface, and technological/resource requirements, utilizing UML diagrams for visualization. Chapter 4 describes the project design including the architectural, component, and logical design. Additionally, the design constraints and tradeoffs are outlined in this chapter. Chapter 5 details the implementation of the project, along with the problems and challenges faced at this stage. Chapter 6 lists the tools used to implement the project, along with the standards applied. Chapter 7 follows the testing strategy for the project, including a description of the strategy and the outcomes. Chapter 8 summarizes the project from the ideation to the implementation, as well as discusses potential future implementations that may benefit the models.

Chapter 2 Background and Related Work

2.1 Background and Used Technologies

Concepts that need background information for this project are the understanding of cryptocurrencies, blockchains and what makes the prices of these coins fluctuate. Cryptocurrencies are virtual coins that are encrypted using asymmetrical encryption so that they can stay secured. They are based on blockchain technology which means that these coins are stored digitally based on peer-to-peer networking (Frankenfield, 2022). Since cryptocurrencies are a blockchain, they are allowed to be decentralized, meaning that they are not being controlled by any main authority. Many factors can cause the prices of these coins to fluctuate, therefore it is hard to predict. These factors include government regulation, demand and supply, current economic, world news, social media such as Twitter, Reddit or Facebook, and so on. In the scope of this project, tweets which mention Bitcoin, Ethereum, and other top cryptocurrencies are collected. The tweets will then be pre-processed, analyzed and compared to the price of the same day. This step is to determine the correlation between Twitter sentimental and cryptocurrency's price. Besides analyzing Twitter sentiment, the prices will also be predicted by using machine learning models to perform technical analysis from historical data. The historical data that will be collected are candlestick components, market capitalization, and trading volume. These indicators are essential to predict possible price movement. In addition, Bitcoin dominance is also utilized for this project. Bitcoin is the first virtual asset and it still remains the largest market capitalization in this industry. Every movement of Bitcoin will affect other assets. By looking at the Bitcoin dominance, investors are able to learn how the money flows between Bitcoin and other assets, then make a prediction for the price correspondingly. Therefore, Bitcoin dominance and historical data will be used to train machine learning models.

Since the project will be created using machine learning with data pulled from various websites, there will need to be familiarity with the API frameworks from *Twitter, Yahoo Finance, and CoinMarketCap.* API calls from these sources will be used to gather the necessary data in order to create the prediction models. For Twitter, sentiment analysis will be used to gather keywords from Tweets that would be assumed to have a positive or negative effect on the prices. Prediction of prices using historical data will be created using a prediction model based on three different algorithms. The three algorithms are gradient boosted tree, Neural Network, and K-Nearest Neighbor. They will be implemented in Jupyter Notebook on Python3 to then create the prediction model. These models will be running asynchronously so that it would not take up the majority of the time and other stuff can be done while the prediction models are being trained.

For the website, Flask was the web framework used to run the website. According to hackr.io, the reason why Flask is a good framework because it is lightweight, so it is not that hard to use, it has its own built-in debugger, and it does well at handling requests ("Best", n.d.). For our cloud computing platform, Amazon EC2 was used. For frontend coding, HTML, Javascript, and CSS was used to create the interface and design of the website. For the backend, SQLite was used to store data that involves the Data Collection/Pre-Processing Model.

Waterfall methodology was implemented in that first a date will be set for the design, implementation, testing, and deployment. A linear process will be used which implies that before moving onto the next phase, the originally planned phase must be completed first. An example of the process for the backend people would be creating a workable prediction model first for one of the algorithms and then when finished with the testing stage, the frontend people will then test the prediction model and then figure out how to embed the model onto the website. After each individual prediction model is working, all of the algorithms will then be combined together using the boosted ensemble learning method so that it will create a more accurate prediction model. The boosted ensemble learning method uses 1 training sample and multiple different models. The method seeks to change the training sample to focus attention on samples that the model predicted incorrectly. It then iteratively adds models to correct predictions of previous models.

For this project, most of the time will be spent on training the prediction models. Creation of the website should not take that long, but there will be two separate teams each with two backend and two frontend developers. A lot of resources will be required to finish this project as there will be a lot of reliance on APIs since the project requires pulling historical data from a variety of sources and Tweets from Twitter. For now, this project will not cost anything to make, but that may change in the future as features will start to be implemented. The project will be using many API calls, in which the free tier of the sources chosen will provide sufficient information to create accurate prediction models. The risks of the project is that since there is a reliance on other websites to pull information from, if one of the sources goes down it would affect the models. Since machine learning is being used, it is necessary to at least know the mathematical fundamentals for statistics, linear algebra, probability, and calculus.

Table 2: List of Relevant Courses

CS-156	This course was applied to this project by integrating the machine learning algorithms learned in class such as KNN (k-nearest neighbors), Neural Networks, Gradient Boosted Trees, and the Ensemble Learning Method.
CS-157A	This course helped with data storage and using it to create databases for the project.

CMPE-165	The theory behind project management and having a project manager for this project was used. We had a project manager that would help the group keep on schedule and make sure that all of the work was being done on time. Also, creation of a website using html, css, and javascript in this class helped a lot with making a website for this project.
CMPE-187	This course was applied to this project by creating test plans that were used during the debugging stages of the application. Multiple variations were tested that could possibly affect the website or cause the application to have problems.

2.2 Literature Search

From our literature search, we have found several scholarly articles using the SJSU library and scholar.google.com. These articles include algorithms and data sets that have been previously used to predict the price of cryptocurrency based on historical data or Twitter feeds using Machine Learning. Chowdhury et al. (2020) gives a general overview of four methodologies that they used to predict the prices of cryptocurrencies. The names of their four methods are: gradient boosted trees, neural networks, ensemble learning method, and K-NN. From these four algorithms, they were able to use Machine Learning to train different models and display the accuracy of their results on a graph from RapidMiner.

A study by Akyildirim et al. (2021) suggests that the machine learning algorithm is "promising for short-term forecasting of trends in the cryptocurrency markets." In the study, the authors used 4 machine learning classification algorithms which are logistic regression, support vector machine, artificial neural networks, and random forest. In addition, the data used in the study were classified into 4 different time frames which are daily, 15 minutes, 30 minutes, and one hour. The result from this and the abovementioned study help our group to study and choose the suitable algorithms to fit into our project.

According to a journal written by Ahmadisharaf et al. (2022), neural network is a good choice of methodology that can be used for predicting cryptocurrency. It has an accurate prediction power and it is also very powerful. For their experiment, they used Yahoo! Finance as their chosen resource to scrape their data from and to get the historical prices of the cryptocurrencies. They said that while neural networking is strong on its own, it would be better to use a hybrid model that would help further fine tune the results and get more accurate predictions.

The research by Olivier and Johannes (2020) explains how Twitter Tweets can be used to predict future cryptocurrency prices based on their sentiment. The Tweets selected are related to the nine largest cryptocurrencies based on market capitalization in May 2018. The Tweets are then pre-processed to filter out noise elements from the tweets. After the Tweets are pre-processed, then the methodology for forecasting cryptocurrency prices is conducted. These Methods include testing for cryptocurrency-related Twitter bots, sentiment analysis, analysis using a lexicon-based approach, and Granger-causality testing. From these methods, they were able to determine if Twitter influenced the prices of different cryptocurrencies. Using this article, we can implement a sentimental analysis in our machine learning algorithm to predict the future price of cryptocurrencies. Similarly, a work by Pathak and Kakkar (2020) indicates that combining data from social media and the historical pricing data would return a highly accurate price prediction. The percentage of their prediction is approximately 80%. This result reinforces our choice of choosing data from Twitter and historical data for the project.

We will be studying these articles thoroughly because these are the exact ideas that we are trying to implement into our project. That means that as a reference these articles will be extremely useful to us and we as a group are glad that there was already a literature review on this information.

2.3 State-of-the-art Summary

To develop the models required to implement the cryptocurrency forecasting model, *Google Colab using Python3* is utilized.

To obtain data to use for forecasting the prices of cryptocurrencies, API frameworks were available from, *Yahoo Finance.* Additional datasets and metrics were also obtained from *IntoTheBlock.* These platforms provide trusted and accurate data from historical data to order books of variable time intervals. Another source that provides evaluated data is the *CCi30,* which is a relatively new concept of generalizing the volatility of the cryptocurrency market by evaluating the performance of the 30 largest currencies. The overall blockchain performance influences the performance of individual assets.

For forecasting prices via technical analysis, three machine learning techniques were selected as promising models: *Gradient Boosted Trees (GBT), Neural Network (NN), and K-Nearest Neighbor (KNN).* After experimenting, only two models were selected out of the three: NN, and KNN models. GBT's accuracy was lower than expected, therefore we decided to not to move with GBT. Each of these models will provide different predictions as a result of the different learning techniques. The *ensemble learning* technique is used subsequent to the parallel models to select the best model with the most accurate prediction. To evaluate the performance of the best model and the prediction via sentimental analysis, *RapidMiner* is used concurrently with manual algorithms in Google Colab to ensure a thorough evaluation.

For Twitter sentiment analysis, Twint was chosen to replace Twitter's API as it provided more benefits with zero cost required.

When beginning to forecast using sentiment analysis, standard NLP data pre-processing techniques were used to properly format the data. The *VADER algorithm* was then implemented to obtain polarity scores from social media text posts and to normalize the dataset. *Granger-causality testing* was utilized to determine a prediction pattern of the dataset.

HTML/CSS, Flask framework, and Scikit-Learn were used to clearly illustrate the prediction results, along with accuracy scores and prediction probabilities. The model would be deployed in an *AWS EC2* instance, due to the scalable sizing and frictionless configuration.

To ensure the program operates as intended, a particular software testing process would be used. *Unit testing* would initially be used to verify that individual elements perform successfully. *Integration testing* makes sure that components operate well when coalesced. *System testing* tested the system as a whole from a developer's perspective to verify flawless execution. *Acceptance testing* tested the system from an end-user's perspective.

Chapter 3 Project Requirements

3.1 Domain and Business Requirements

The diagram below illustrates the processes performed by the high-level entities of the project. The main entities involved in the project are the end user and the GUI. The objects utilized in the model are the Data Preprocessing Model, the Data Model, the Forecasting models, and the Performance Evaluation Model. The end user begins the process via a button on the user interface. After the initiation of the process, the Data Preprocessing Model prepares both the technical and sentimental data to be used for predictions. After the data is manipulated, the Data Model ensures that the data to be used is valid. Once verified, the Data Model sends the appropriate data to each model in the Forecasting package, which is used to generate price predictions. After the first iteration of predictions are made, the resulting predictions are sent to the Performance Evaluation Model to analyze the accuracy. If the accuracy is undesirable, the hyperparameters of the forecasting models are tuned to achieve greater accuracy. If the accuracy is optimal, the results are reported to the GUI. The GUI displays the price predictions generated to the end user.

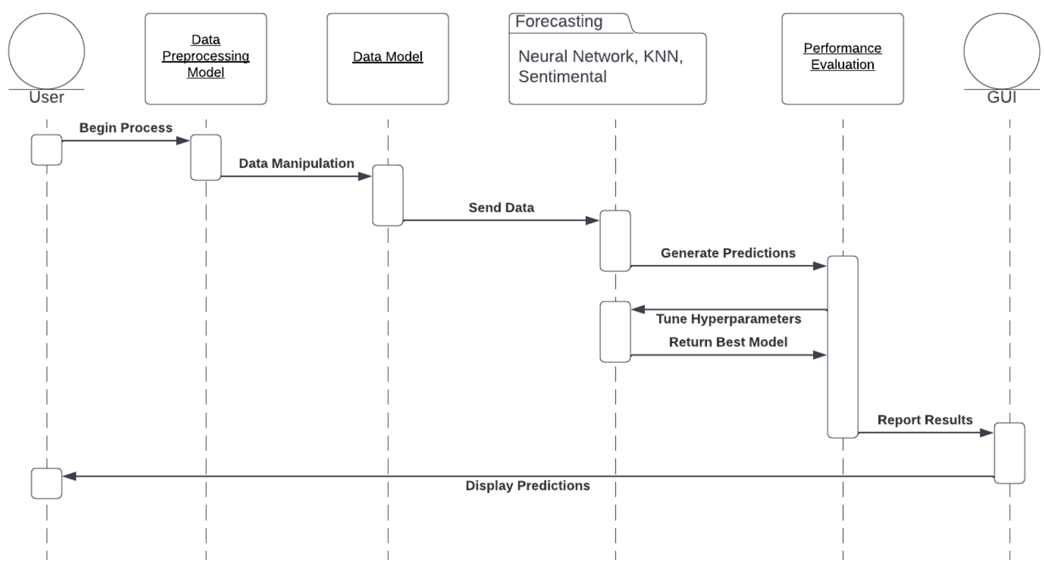

Figure 1: UML Process Diagram

The diagram below visualizes the relationships between the domain-level classes in the software. The four classes used in the technical prediction are DATA_PREPROC, KNN_DIR, NEURAL_NETWORK, and TECHNICAL_RUNNER. The implementation of KNN_DIR depended on DATA_PREPROC to prepare the data accordingly. NEURAL_NETWORK was associated with KNN_DIR to access the directional movement predictions. The TECHNICAL_RUNNER was associated with NEURAL_NETWORK to access the price predictions generated. The two classes used in the sentimental prediction are SENT_DATA_PREPROC and SENTIMENTAL_ANALYSIS. SENTIMENTAL_ANALYSIS was associated with SENT_DATA_PREPROC to access the prepared data for the predictions. Two classes utilized data from both the technical and sentimental predictions: DATA_ANALYSIS and the GUI. DATA_ANALYSIS was associated with KNN_DIR, NEURAL_NETWORK, TECHNICAL_RUNNER, and SENTIMENTAL_ANALYSIS. The reason this class was associated with all forecasting models individually was because it analyzed each prediction at a low-level to ensure accuracy. The GUI interface realizes the implementation of both the TECHNICAL_RUNNER and SENTIMENTAL_ANALYSIS for valid execution and visualization.

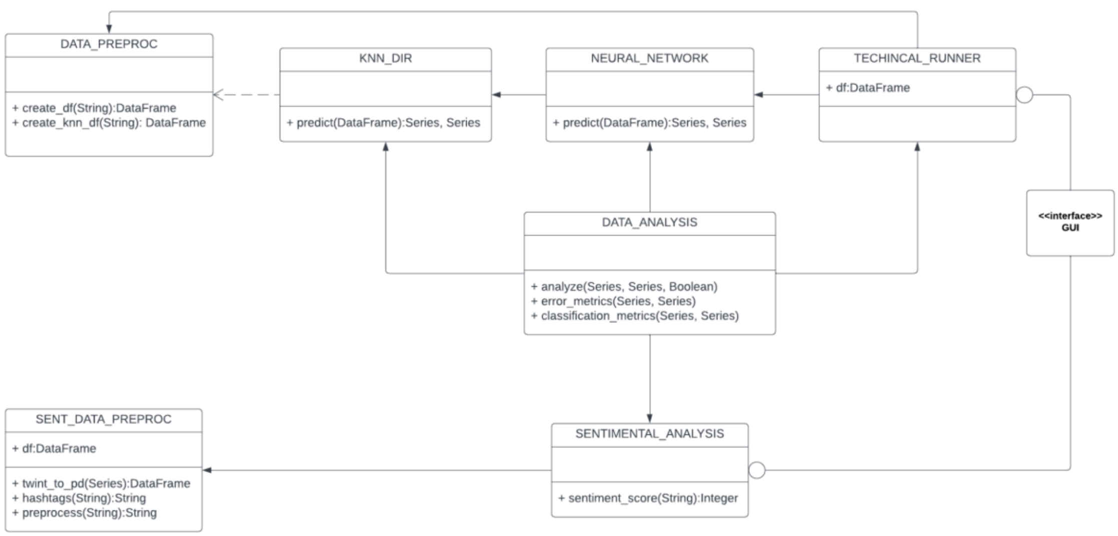

Figure 2: UML Class Diagram

The state diagram shown below describes the states/statuses of various systems throughout the execution process. Upon initiation, two processes executed in parallel: one for the technical predictions and one for the sentimental predictions. The technical prediction process was detailed as follows. A fixed dataset of technical indicators and derivatives was first imported onto the end-user's machine, into the Technical Data Preprocessing state. Upon successful execution, the preprocessed data was sent to the KNN Forecasting state. After all iterations finished, the directional movement predictions returned were imported into the Neural Network Forecasting state. From here, the price predictions were exported to the Technical Data Analysis state. The performance metrics determined from this state were then evaluated. If the accuracy is optimal, then the results were sent to the Technical Driver. If the accuracy is undesirable, the hyperparameters of the technical forecasting models were tuned for better accuracy. The Technical Driver sent the results to the GUI to visualize the predictions for the end user. The sentimental prediction process was outlined as follows. After initiation, the data was collected and preprocessed in the Sentimental Data Preprocessing state. The preprocessed data was used to generate predictions in the Sentimental Forecasting state. These predictions were then analyzed in the Sentimental Data Analysis state. From here, the final predictions were sent to the GUI to visualize to the end user. After both the technical and sentimental predictions were displayed to the end user, the process was terminated.

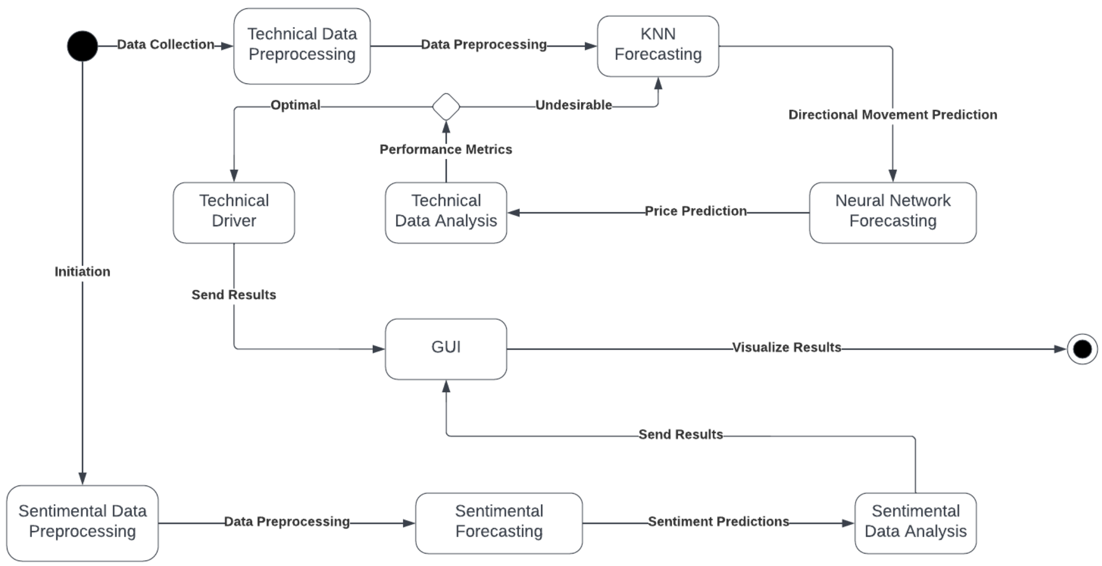

Figure 3: UML State Machine Diagram

3.2 System (or Component) Functional Requirements

Table 3: List of Functional Requirements.

Requirement No.	Functional Requirement Description
FR1	A list shall display all available coins
FR2	Result of prediction model should be displayed publicly on the website
FR3	Result of Twitter sentimental analysis should be display publicly on the website
FR4	Result shall be updated every 24 hours
FR5	System shall track the data and time of last API data pull

FR6	System shall provide read-only access to public
FR7	Displayed graphs should be interactive

3.3 Non-Functional Requirements

Table 4: List of Non-Functional Requirements

Requirement No.	**Non-Functional Requirement Description**
N-FR1	Each page must load within two seconds
N-FR2	The system must be able to run on macOS and Windows
N-FR3	The system must run on any integrated development environment (IDE)
N-FR4	The mean time to restore the system (MTTRS) following a system failure must not be greater than 10 minutes
N-FR5	The webpages must be available to users 99.98 percent of the time every month
N-FR6	The error rate of users navigating through pages shall not exceed 10 percent

3.4 Context and Interface Requirements

Environments that were used to develop, test, and deploy the project was Windows, MacOS, and Python. Windows and MacOS was the operating system environments that were used to develop and test the project. Python was the language that was used to create the actual project. To write the code for the project, IDE's such as PyCharm was used. One of the interfaces that was imported was matplotlib.pyplot which was used to show the actual results of the graph that was used to predict the cryptocurrency price. Flask was then used to create the website for the project and in order to run Flask, a localhost connection was needed with an open port. From there, a web application was needed to open the website of the application. AWS EC2 was then used for testing which allowed any person with an internet connection to be able to access the website and see the deployment of the project.

3.5 Technology and Resource Requirements

Table 5: List of Technology and Resource Requirements

Requirement No.	Type	Description
R1	Hardware	PC
R2	Operating System	Windows/MacOS
R3	Software Application	Web Browser
R4	Interpreter	Python3
R5	Text File	Requirements

There are six requirements needed to successfully execute the forecasting model. Requirement 1 calls for an operational PC to be used. This PC must be able to download files and have a sustainable internet connection. The operating system must preferably be Windows or MacOS, as detailed in Requirement 2. Arguably, a Linux operating system may be used if root privileges are used. Requirement 3 states some web browsers capable of visualizing HTML/CSS/JavaScript scripts must be installed. Google Chrome tends to work most flawlessly. Moreover, Python3 must be installed, as shown in Requirement 4. Note that an IDE is not needed, only the ability for the PC to interpret Python3. The final requirement is a valid text file listing all Python requirements, denoted “requirements.txt”. The Python3 interpreter discussed in Requirement 4 is needed to traverse through the requirements text file for sequential installation. The end user may opt to manually install the Python modules or allow the software to automate the installation. The file “requirements.txt” contains the following modules that are not included in the Python Standard Library.

Chapter 4 System Design

4.1 Architecture Design

The cryptocurrency forecasting model employed both supervised and unsupervised learning, parallel processing, and service-oriented architectures. The supervised learning approach and parallel processing architecture were used during the technical analysis. The unsupervised approach was used during the sentimental analysis. The overall model drew inspiration from a service-oriented architecture given the minimal dependencies. The architecture was divided into two categories that executed in parallel, one for technical forecasting and the other for sentimental forecasting. It is imperative to note that all the models described below are high-level and there are numerous lower-level models that exist within the said models.

In the technical forecasting category, the data was retrieved from a pre-populated, fixed Comma-Separated Values (CSV) file, denoted “Technical Dataset”. All technical indicators and derivatives, with the exception of historical prices, reside in this dataset. The data from the Technical Dataset was then imported into the Data Preprocessing Model, where new values were determined using the values from the Technical Dataset and organized into a Pandas dataframe. Additionally, the historical prices were appended in this model as well. The dataframe returned from the Data Preprocessing Model was used in the KNN model to determine yet another feature: the direction. Using all the features derived from the previous models, a set of daily price predictions was calculated in the Neural Network Model.

In this sentimental forecasting category, the data was retrieved via the Twint tool in the Sentimental Data Collection Model. In this model, we extracted tweets from the social media application Twitter that correspond to the top ten cryptocurrencies. A total of 1000 tweets were extracted for each cryptocurrency. The tweets were preprocessed by removing hashtags and removing links. After preprocessing the tweets, a sentimental model was created using a pretrained tokenizer and model. The tokenizer was used to encode the data and the result was a model of the tokenizer that analyzes each word of the tweet and returns a score depending on the mood of the tweet. After these tweets are analyzed, the data was saved as a Pandas DataFrame and then saved as a CSV file. Each day, the model saves 10 CSV files, one file for each cryptocurrency which shows the amount of tweets limited to 1000 and the sentiment analysis score for those tweets. In a given day, there is a total of 10000 tweets extracted and scored using the sentimental analysis model.

At the final stages of each of the said categories, the predictions were exported to two different models concurrently. The first model was the Performance Evaluation Model. This model analyzed the predictions against the validation dataset to compute the performance of the forecasting models. The second model is the GUI that visualized the results to the end user.

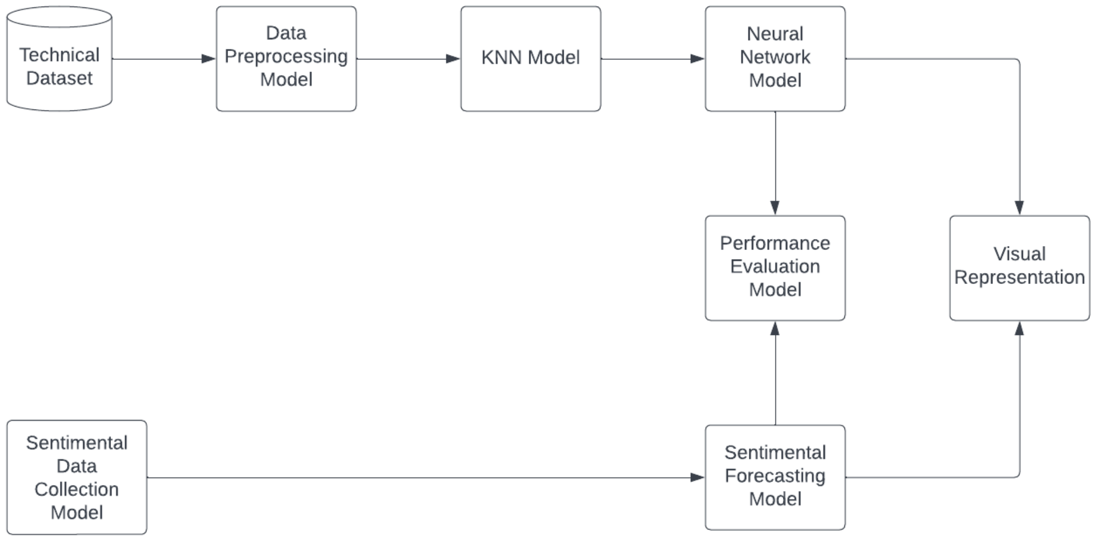

Figure 4: Architectural Design Diagram

4.2 Interface and Component Design

The component diagram below shows various components that form the application of the Cryptocurrency Forecasting model. Each component is responsible for a specific action or process.

There are 3 components in Technical Analysis package. These components are responsible for analyzing datasets that were retrieved from the Technical Dataset node. In the same time, the three components in Sentimental Analysis package are responsible for retrieving data from Twitter, pre-processing the data, and analyzing the data for the final result. The Performance Evaluation component would then evaluate results the results generated by Technical Analysis and Sentimental Analysis packages. The Graphical User Interface node is responsible for showing final results to users.

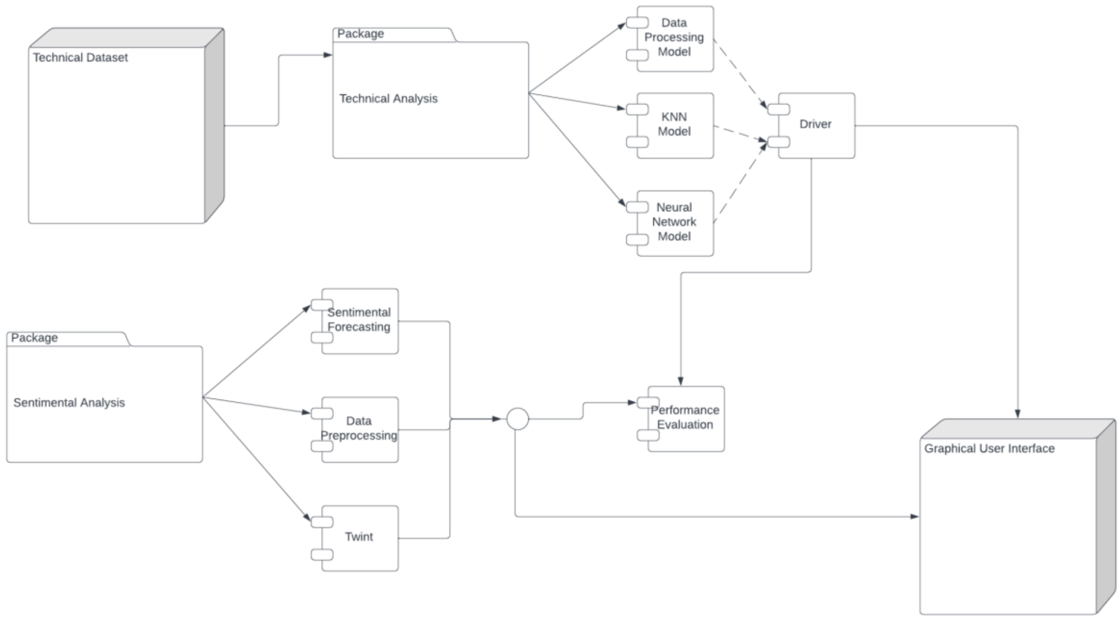

Figure 5: Component Design Diagram

4.3 Structure and Logic Design

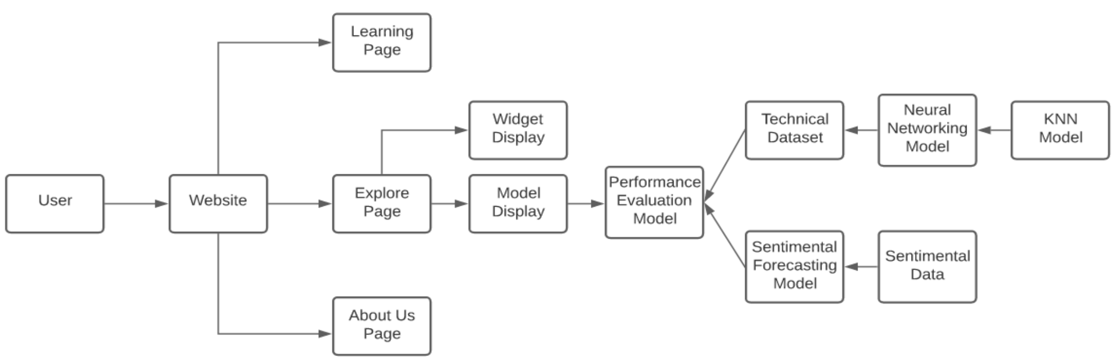

Figure 6: Structural and Logical Design Diagram

This is a structure and the logic design for what the project looks like. The structure is that the website will have 3 main pages excluding the startup page that the user sees immediately after visiting the website. From there they have the option to go on three different pages, either the learning page, the explore page, or the about us page. If they go onto the learning page, they will be able to see a variety of articles that can help them learn about cryptocurrency and why it is popular. If you go to the explore page, you will be able to see widgets that were provided by CoinMarketCap that displays the price, ranking, and market cap on the coins that the project is representing. There will also be the prediction model on that page which uses technical dataset and sentimental analysis to create an evaluation model of predicting whether the price of a cryptocurrency will go up or down and a rough estimate of what the price will be in the next few hours. For the performance evaluation model model, it uses KNN and Neural Networking and includes sentimental analysis from Twitter. Lastly, the user is able to go to the About Us page where it will just be information about the creators of the project.

4.4 Design Constraints, Problems, Trade-offs, and Solutions

4.4.1 Design Constraints and Challenges

One constraint for this project is that it requires the usage of multiple APIs from different sources. This means that the reliability of the models for the project will depend on these sources. If one of them were to fail, then it could potentially affect the accuracy of the models. From some of those API sources, Twitter allows a maximum API request of 30 requests per minute which is about 3000 tweets per minute. Getting historical data of cryptocurrency is also a big challenge. Almost all of the other API calls that were found required some form of payment monthly to use the API that was necessary for the project, which was too much economically as the project is planned to be created with no money used.

For resources, the constraint would be Twitter in general. It has a reliance that the sentimental analysis will be trained correctly to where it will be able to sort out positive and the negative tweets for the Machine Learning process. However, with Elon Musk now buying Twitter, it is unsure of how that may affect how people tweet about cryptocurrency and the reliance of a majority of the tweets. Twitter bots, which are bots created to automatically tweet on behalf of a user, can be made to spam positive or negative tweets related to a cryptocurrency. This may skew the sentimental analysis since the model doesn’t know whether the tweet was made by a real person or a bot. Another constraint on resources would be the webserver that the website will be hosted on. The project will be hosted on Amazon EC2 which is free to try, which is why it was picked. However, to test the project, it will probably be tested locally and then when all of the steps are finally completed it will be moved onto Amazon’s cloud-computing platform.

A challenge that may be faced would be the mathematical theories needed to create the Machine Learning algorithm to train the models. The project will consist of using four different algorithms to train the models which means that a lot of knowledge in the mathematical aspect of the code would need to be understood in order to get the most accuracy for the training of the models. There may also be some safety issues with the website because cryptocurrency could be seen as a form of gambling, so it is required that we look into more information about how cryptocurrency fairs in other countries to make sure that no legal troubles could occur. The project will also have a disclaimer saying that the creators are not an expert in cryptocurrency trading and to use the website at their own cost.

4.4.2 Design Solutions and Trade-offs

A possible solution to replace Twitter APIs is to use Twitter scraping tools. Twint will be utilized for this project. The advantage of Twint is that it does not need Twitter APIs or Twitter sign in to work. It has no rate limitation compared to 3000 tweets per minute from Twitter APIs. The number of daily tweets posted regarding Bitcoin ranges from 100,000 to 300,000. The Twitter API is only able to pull a fraction of these tweets, which may result in an inaccurate model. Twint is the best tool in terms of collecting the Twitter data that is required for this project. However, scraping this many tweets requires a good amount of raw computing power, which is why the amount of tweets for each cryptocurrency was limited to 1000. That way, we can extract 10000 tweets in total for a given day which is a sufficient amount of data.

Next, Yahoo Finance was chosen as one of the APIs to be used for the project. Yahoo Finance provides APIs that allow developers to gather historical data from a longer time span than 30 days with a maximum request of 100 requests per day. And one important reason to choose Yahoo Finance API, it does not require any form of payment to use. Fortunately, only one HTTP request from the Yahoo Finance API provides the model with all the historical data needed.

Besides Yahoo Finance, data was also collected from Intotheblock.com. IntoTheBlock provides more on-chain datasets and metrics that can be ultilized for training machine learning model. One downside of ultilizing IntoTheBlock tools was that it was required a form of payment. However, the value it brings to the project is considerable.

A possible solution to maximize the accuracy for the model is the use of the ensemble learning method. The ensemble learning method is a common practice in Machine Learning where the hyperparameters of multiple models are changed, the models are weighed, and the models are combined to yield the best model. The ensemble learning method automates many of the mathematical theories needed to create the best model.

Chapter 5 System Implementation

5.1 Implementation Overview

This project was developed using laptops running on macOS and Windows. The IDE used to create the website was Microsoft Visual Studio Code and IntelliJ IDEA. The main language used was Python3 alongside Jupyter Notebook. The website framework used for the project was Flask, and users were able to run our project by installing the dependencies given inside the requirements.txt file. For the technical data, machine learning algorithms such as Neural Networks, K-Nearest Neighbors, and Gradient Boosted Trees was used. The machine learning algorithms depended on a fixed dataset of the technical data stored in a CSV file ranging from 1/1/2022 to 8/23/2022. For the sentimental analysis data, Twint was used to scrape the data and the Transformers module was used for the model that analyzed the sentiment of the tweets. The sentiment analysis model depended on the Pandas DataFrame of tweets that were scraped using Twint.

5.2 Implementation of Developed Solutions

The fixed dataset of the technical data resides in a static CSV file from 1/1/2022 to 8/23/2022. Using this data, new features were determined and were aggregated into a Pandas DataFrame. The historical prices of the respective cryptocurrency was acquired using the Yahoo Finance REST API and aggregated into the DataFrame. Initially, this data was sent to the K-Nearest Neighbors algorithm to perform a classification prediction to forecast the directional movement. 15 neighbors were used in this algorithm. These directional predictions were aggregated into the DataFrame to be used in the Neural Network algorithm. The Neural Network algorithm was a sequential model with the following ordered layers: LSTM (50 units), Dropout (0.2 dropout rate), LSTM (60 units), Dropout (0.3 dropout rate), LSTM (80 units), Dropout (0.4 dropout rate), LSTM (120 units), Dropout (0.5 dropout rate), and a Dense layer with 1 unit. All LSTM layers had the Rectified Linear Unit activation. These layers were compiled using the “adam” optimizer algorithm. The resulting predictions were sent into a data analysis model to evaluate the performance. Initially, the error metrics Mean Squared Error and Root Mean Squared Error (RMSE) were calculated. For the K-Nearest Neighbor evaluation, the accuracy score, confusion matrix, precision score, recall score, and the F-score were determined. The RMSE, accuracy score, and F-score were used to determine if the accuracy was optimal or undesirable. The dataset for the sentimental forecasting were simply tweets from Twitter extracted using the Twint module. 1000 tweets per day between the dates 1/1/2022 and 8/23/2022 were utilized. These tweets were then validated and aggregated into a Pandas DataFrame. The hashtags of the tweets were removed and the text was preprocessed for prediction preparation. The Transformers module tokenized the tweets to score them using an expansive list of keywords. The tweets along with the score were then imported into a CSV file. The daily sentimental score was determined by taking the average score of all 1000 tweets from a certain day. These average daily scores were compared against the actual direction of the cryptocurrency to determine causality. The Python Flask microweb framework was used to display the prediction results in a clear graphical user interface.

5.3 Implementation Problems, Challenges, and Lessons Learned

Implementation problems and challenges that occurred during the process of making the project was that for the Twitter sentiment analysis, data could only be scraped from a limited number of dates. If the date for the tweet scraping went too far back, then there would be a message saying that there was “No more data!” and no tweets would be received from the program. Another problem that occurred for implementation, was fine tuning the prediction models. If too much or too little data was used to train the model, it would yield inaccurate predictions. Through intensive unit testing, the hyperparameters of the models were tuned and a solid dataset was used to achieve optimal accuracy. For the website, the most difficult thing was displaying the plot and making sure that the user is able to interact with it and that it updates over time. Some lessons learned from the project is that Cryptocurrency is highly volatile, so trying to get a really accurate prediction model might be difficult. Something else learned is that a lot of information can be found online and that there are a lot of companies that provide useful libraries and APIs that can be used by anyone.

Chapter 6 Tools and Standards

6.1. Tools Used

The machine learning model used to predict the price of a given cryptocurrency was developed using Python 3.10. The directional movement forecast utilized the K-Nearest Neighbor algorithm and the price prediction was evaluated using the Neural Network algorithm. The K-Nearest Neighbor classifier was made available using the sklearn module from the scikit library. The sequential Neural Network model was acquired from the Keras library. The website to display the models was also developed using Python. The website also displayed the average sentiment scores for the top ten cryptocurrencies using scraped tweets from Twint. Twint was used to scrape tweets instead of the Twitter API because Twint is more efficient in gathering tweets whereas the Twitter API has a rate limit. Furthermore, Twint utilized Twitter's search operators to scrape tweets with a certain criteria. The sentiment scores for the tweets scraped from Twint were calculated using a pretrained sentimental analysis model called RoBERTa (Robustly Optimized BERT Pre Training Approach). This model was used for the sentimental analysis because it is heavily optimized for language processing. The web framework that was used was Flask which utilizes HTML, CSS, and JavaScript to create the website. Other tools that were used to create the website were using a script created from coinmarketcap.com which is used to display the price and the market cap of the coins. In addition, Chart.js library was used to create interactive graphs to display the result from machine learning models. A state-of-the-art tool that was used to collaborate with code was Github where latest codes were pushed and then another person would take over and pull from the code that was pushed. Another thing that was used was Canvas to turn in all of the work and both Gmail and Discord as a form of communication.

6.2. Standards

A standard that was used in our project was to make sure that our website would be able to run on any system and that each section of the website is able to be seen. It should be able to run on any operating system and web browser. Requirements were that the interface should be user-friendly and that the website should be easy to navigate around. Testing was done to make sure that these standards were met, and that the website would be able to be run on any computer. Other standards that needed to be met were that our models would need to be accurate and that the sentimental analysis pie chart will be easy and useful to look at. The news page will display the most recent news about cryptocurrency and will be able to link to the article page so that the user can read about it. The standard for design is based on Figure 6, with the structure and logic design diagram. The user should be able to access the website and then from there, they will be greeted to the home page. After, they have three options to move to the news page, the explore page, or the about us page. From the explore page, there is a display of widgets that has information about the top 10 coins and the models will also be displayed on that page. There will be a graph chart for the technical analysis showing the prediction of the coin's price. Then, there will be another model that will be a pie chart that will display the sentimental analysis part which it would show whether the coin is doing well on Twitter or not.

Chapter 7 Testing and Experiment

7.1 Testing and Experiment Scope

The intent of the testing strategy is to ensure the successful execution of the features with a minimal number of bugs or errors. This includes an optimal performance of the technical predictions, the sentimental forecast, and the graphical user interface.

Unit testing and domain testing at the component level will be conducted by the respective developer. To ensure interoperability between each component, integration testing will be conducted by all developers.

Functional testing will confirm all functional requirements (Table 3) and non-functional requirements (Table 4) are met. All links in the webpage will be tested to make sure they are working correctly, and that no broken links exist. Links that need to be tested include:

- Outgoing links
- Internal links
- Anchor links

Usability testing tests on navigation and the content of the web pages. In navigation, the menu, buttons, and links are visible and consistent on all web pages. The contents are readable with no grammar. Images are always present, and graphs are responsive.

Interface testing will be conducted in two areas: Application and Web server. In the application, test requests will be sent correctly to the server. Output from the server will be displayed quickly and correctly in the client's web browsers.

The final testing process used is user testing, to maximize the user experience. The goal of utilizing the said testing processes is not limited to meeting the stated requirements, but also detecting and resolving existing bugs.

Table 6: Summary of testing processes used.

Testing Process	Conductor	Locale
Unit Testing	Developer of module	Individual units of module
Domain Testing	Developer of module	Individual units of module and entire module
Component Testing	Developer of module	Entire module
Integration Testing	All developers	Consolidated modules
Functional Testing	Developer of module	Consolidated modules
Usability Testing	Developer of module	Application
Interface Testing	All developers	Consolidated modules
User Testing	External participants	Application

7.2 Testing and Experiment Approach

The primary testing strategy is divided into four phases, with each phase focusing on a particular scope of the architecture. One or more testing processes are utilized within each testing phase. A detailed summary of the testing strategy, phases, testing processes, and test case distribution is illustrated in Table 7.

Test Phase 1

Test Phase 1 focuses on individual units of each module. Developers will begin by conducting unit testing for their respective modules to ensure each unit executes as intended. The test cases relevant to each unit are constructed depending on the inputs of each individual algorithm. Five unique test cases are generated for each unit, in which at least one test case is expected to fail. In addition to unit testing, domain testing on the unit level is utilized to verify the handling of different data types and inputs that may not be in the required domain. Three test cases are created for unit-level domain testing. Upon successful execution of the specified test cases, Test Phase 2 initiates.

Test Phase 2

Test Phase 2 focuses on individual modules of the system. As the process for predicting the price and sentiment is uniform, the focus of the respective testing oracles for this phase relate to the validity of imported data. As a result, input domain testing is used to ensure the imported data is acceptable or handled accordingly, inclusive of outliers and differing data types. In order to determine the test cases for this process, the output range of each unit and module is determined. Based on the output range, three particular test cases are generated, in which at least one test case is expected to fail. After successful unit testing, each developer will perform component testing to confirm that each module independently runs successfully. Successful component testing is determined by requirements FR1, FR6, and N-FR1. (Table 3 and Table 4). The test cases for this process are created such that each test case ensures module-level, interoperable execution. Ten test cases are generated where at least one test case is expected to fail. Subsequent to the unit, domain, and component testing processes, Test Phase 3 begins, and integration testing is used to ensure flawless execution of the system as a whole.

Test Phase 3

Test Phase 3 focuses on the complete system of the Cryptocurrency Forecasting Model. Both interface and integration testing will be conducted here. Interface testing focuses on ensuring proper communication between different subsystems. This testing process validates requirements FR4, FR5, and N-FR5. The two test cases for this process include testing on the application side and server side. Integration testing tests the interoperability of all of the modules/interfaces and is completed after requirements FR2, FR3, N-FR2, N-FR3, and N-FR4 (Table 3 and Table 4) are met. Eleven test cases for integration testing are constructed. Ten of these cases test each of the cryptocurrencies available. The eleventh test case is designed to intentionally fail the system to determine the MTTRS detailed in N-FR4. The final phase in the testing strategy is user testing.

Test Phase 4

Test Phase 4 focuses on the complete implementation of the minimum viable product. User testing ensures that the user experience operates as intended and requirement N-FR6 (Table 4) is met. A test procedure is created for two independent participants. The first user aims to view technical and sentimental forecasts of all cryptocurrencies available on the application. The second user aims to view three chosen news articles on the application. The time it takes for the user to complete the procedure and the number of errors made along the way will be recorded.

Table 7: Detailed view of testing strategy.

Test Phase	Locale	Testing Process	Test Case Distribution	Requirements Accounted
1	Unit	Unit Testing	5 test cases per unit	N/A
	Unit	Domain Testing	3 test cases per unit	N/A
2	Module	Domain Testing	5 test cases per module	N/A
	Module	Component Testing	10 test cases per module	FR1, FR6, N-FR1
3	System	Interface Testing	2 test cases	FR4, FR5, N-FR5
	System	Integration Testing	11 test cases	FR2, FR3, N-FR2, N-FR3, N-FR4
4	Application	User Testing	2 test procedures	N-FR6

7.3 Testing and Experiment Results and Analysis

In the initial run of the testing strategy detailed in the previous section, there were a number of bugs encountered, as expected. Most bugs were found in Test Phase 4, during user testing. Table 8 illustrates the major bugs found during testing. After all known bugs were resolved, a second run of the testing strategy was conducted. After this run, no major bugs were found, and all requirements stated in Table 3 and Table 4 were met. The result of the testing procedure is a viable and usable product.

Table 8: Bugs logged during testing

Bug No.	Description	Phase	Assignee	Status
B1	Some price predictions rounded to only a single decimal place.	1	Deven	Resolved
B2	Upon first execution, JSON that holds the predictions was not created.	2	Deven	Resolved
B3	Tweets scraped from Twint would only show the most recent tweets instead of a date range.	3	Berlun	Resolved
B4	Less than optimal tweets were scraped for a certain topic from Twint, making the sentimental analysis score less accurate	2	Berlun	Resolved
B5	Models would not show up when switching to a different coin and then changing to a different page and changing back.	4	Brandon	Resolved
B6	News page not displaying properly	4	Brandon	Resolved
B7	Graphs would not display data unless resizing the web browser	4	Duc	Resolved
B8	Graphs responded slower than user movements and sometimes froze	4	Duc	Resolved

Chapter 8 Conclusion and Future Work

The digital asset industry is rapidly evolving and attracting brand new investors. However, two problems arise from this. The first being the volatility of digital assets resulting from uncertainties of investors. The second being the lack of education of novice investors, as most vital mistakes happen in the adolescent stages of one's investing experience. The Cryptocurrency Forecasting Model is a comprehensive system that predicts the price and evaluates the sentiment of ten selected cryptocurrencies. By analyzing a plethora of technical metrics and derivatives, as well as sentimental data from social media, the model is able to provide detailed and relatively accurate assessments on the state of the cryptocurrency market, specifically relating to the overall sentiment and the price dynamic of coins/tokens. This project met the specified needs of various people, whether it be people who are socially and/or environmentally disadvantaged, or simply people who are looking for help in their cryptocurrency portfolio. The website informed people about cryptocurrency dynamics and allowed every user to be able to have the opportunity to see future prices of a set of coins. To ensure meeting these needs, 7 functional and 6 non-functional requirements for the model were predetermined.

The technical analysis portion of the project utilized concepts of K-nearest neighbor, neural networks, and standardized performance evaluation techniques to effectively forecast the prices. A fixed dataset was aggregated using publicly available data from IntoTheBlock and Yahoo Finance. Subsequent to the preliminary exploratory data analysis and feature selection, the K-nearest neighbor algorithm was used to provide a brand new directional feature. Using the historical prices and the derived directional feature, predictions were made using a sequential neural network algorithm. On the other hand, the sentimental analysis portion of the project utilized the concept of tweet scraping, transformers models, and social media sentiment to predict the attitude of Twitter tweets made specifically towards different cryptocurrencies. These tweets were effectively scraped using Twint and then converted into a Pandas DataFrame in Python. A total of ten different DataFrames were then made which correspond to the top ten cryptocurrencies in the market. The tweets were then analyzed from a sentimental standpoint using a Robustly Optimized BERT Pretraining Approach (RoBERTa) with bidirectional encoders. Using this model, the tweets were then tokenized and given a sentimental score. This model was then used to determine the correlation between the Twitter sentiment of a cryptocurrency versus its historical price. To test the model, a four-phase testing approach was used. The initial phase focuses on specific components of each module and the final phase focuses on the general application as a whole. Upon 10 iterations of the technical analysis portion of the model, an average accuracy of about 0.57 was obtained.

Due to time constraints and resource availability, the true potential of the application has not been reached. Future work based on the current implementation is abundant. One possible future implementation is applying the model to all cryptocurrencies at a given time, dynamically accounting for new and terminated cryptocurrencies at the microsecond time scale. When idle, the model retrieves all data from a dynamic dataset utilizing various APIs or blockchain calculations. This widens the target market as there are hundreds of different cryptocurrencies that people wish to gain insight on. Another possible future implementation includes allowing experienced investors to optimize the provided machine learning models with respect to their own risk tolerance. The current machine learning model makes predictions based on a general view of major features. This future implementation allows flexibility for experienced investors to allocate feature weight to ones they believe in. These are only a select few of many possible future implementations, as the potential applications this project offers are limitless.

References

A. Ahmadisharaf, A. Lashgari, E. Rabiee, E. S. Pour, H. Jafari, “Cryptocurrency price prediction with neural networks of lstm and bayesian optimization,” *European Journal of Business and Management Research*, 07-Feb-2022. [Online]. Available: https://www.ejbmr.org/index.php/ejbmr/article/view/1307/701. [Accessed: 26-Sept-2022].

A. H. Othman, S. M. Alhabshi, S. Kassim, A. Abdullah, and R. Haron, “The impact of monetary systems on income inequity and wealth distribution: A case study of cryptocurrencies, Fiat Money and gold standard,” *International Journal of Emerging Markets*, 03-Apr-2020. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/IJOEM-06-2019-0473/full/html. [Accessed: 17-Apr-2022].

B. Devera, B. Truong, D. Shah, and D. Tran, “195A Project Workbook Template.docx.pdf”, 16-Mar-2022. [Online]. Available: https://docs.google.com/document/d/1CliQhGrSQRo6mfBUgI5mtsNeFcQFMJWQ/edit#. [Accessed: 20-Apr-2022].

Best Python Frameworks. (n.d.). [Online]. Available: https://hackr.io/blog/python-frameworks. [Accessed: 26-Sept-2022]

E. Akyildirim, A. Goncu, and A. Sensoy. "Prediction of cryptocurrency returns using machine learning," *Annals of Operations Research* , vol. 297, no. 1-2, February 2021. pp. 3+. Gale Academic OneFile, Available: link.gale.com/apps/doc/A650954725/AONE?u=csusj&sid=bookmark-AONE&xid=79d10607. [Accessed April, 14 2022].

J. Frankenfield, “What is cryptocurrency?,” *Investopedia*, 11-Jan-2022. [Online]. Available: https://www.investopedia.com/terms/c/cryptocurrency.asp. [Accessed: April 15, 2022].

O. Kraaijeveld, J. D. Smedt, “The predictive power of public Twitter sentiment for forecasting cryptocurrency prices,” *Journal of International Financial Markets, Institutions and Money*, Volume 65, 2020, 101188, ISSN 1042-4431, February 28, 2022. Available: https://doi.org/10.1016/j.intfin.2020.101188. https://www.sciencedirect.com/science/article/pii/S104244312030072X [Accessed April 17, 2022].

P. D. Filippi and B. Loveluck, “The Invisible Politics of Bitcoin:

Governance Crisis of a Decentralized Infrastructure," *Internet Policy Review*, Vol. 5, Issue 4, September 29, 2016. Internet Policy Review, Vol. 5, Issue 4., Available at SSRN: https://ssrn.com/abstract=2852691 [Accessed April 17, 2022].

"Remittance Flows Register Robust 7.3 Percent Growth in 2021," The World Bank, 17-Nov-2021. [Online]. Available: https://www.worldbank.org/en/news/press-release/2021/11/17/remittance-flows-re gister-robust-7-3-percent-growth-in-2021. [Accessed: 20-Apr-2022].

"Remittance prices worldwide - issue 39, September 2021," Sep-2021. [Online]. Available: https://remittanceprices.worldbank.org/sites/default/files/rpw_main_report_and_a nnex_q321.pdf. [Accessed: 21-Apr-2022].

S. Pathak and A. Kakkar, "Cryptocurrency price prediction based on historical data and social media sentiment analysis," Innovations in Computer Science and Engineering, pp. 47–55, 2020.