Table of Contents

[Figures 4](#_Toc173100090)

[Acknowledgement 6](#_Toc173100091)

[Abstract 7](#_Toc173100092)

[Chapter 01 – Introduction 8](#_Toc173100093)

[1.1 Background 8](#_Toc173100094)

[1.1.1 Importance of Integrating Social Media Sentiment Analysis 8](#_Toc173100095)

[1.2 Aims of the Project 8](#_Toc173100096)

[1.3.1 Academic Questions 9](#_Toc173100097)

[1.3.2 Objectives 9](#_Toc173100098)

[1.4 Scope 10](#_Toc173100099)

[1.4.1 Data Sources 10](#_Toc173100100)

[1.4.2 Machine Learning Techniques 11](#_Toc173100101)

[1.4.3 Social Media Sentiment Analysis 11](#_Toc173100102)

[1.4.4 Big Data Analytics 11](#_Toc173100103)

[1.4.5 Innovative Forecasting Strategies 11](#_Toc173100104)

[1.4.6 Actionable Insights 12](#_Toc173100105)

[1.5 Functional And Non-Functional Requirements 12](#_Toc173100106)

[1.5.1 Functional Requirements 12](#_Toc173100107)

[1.5.2 Non-Functional Requirements 12](#_Toc173100108)

[1.6 Report Structure 12](#_Toc173100109)

[Chapter 01 – Introduction 13](#_Toc173100110)

[Chapter 02 – Literature Review 13](#_Toc173100111)

[Chapter 03 – Methodology 13](#_Toc173100112)

[Chapter 04 – Artifact 14](#_Toc173100113)

[Chapter 05 – Conclusion 14](#_Toc173100114)

[Chapter 02 – Literature Review 16](#_Toc173100115)

[2.1 On Forecasting Cryptocurrency Prices - A Comparison of Machine Learning, Deep Learning, and Ensembles. 16](#_Toc173100116)

[2.2 Predictability of cryptocurrency returns - evidence from robust tests. 16](#_Toc173100117)

[2.3 Cryptocurrency Price Prediction and Trading Strategies Using Support Vector Machines 17](#_Toc173100118)

[2.4 A Deep Learning-Based Cryptocurrency Price Prediction Model That Uses On-Chain Data 17](#_Toc173100119)

[2.5 Predicting Bitcoin Prices Using Machine Learning 18](#_Toc173100120)

[2.6 Forecasting of Cryptocurrency Prices Using Machine Learning 18](#_Toc173100121)

[2.7 Predicting Cryptocurrency Prices with Machine Learning Algorithms 18](#_Toc173100122)

[2.8 Cryptocurrency Price Prediction using Machine Learning 19](#_Toc173100123)

[2.9 Time Series Analysis of Blockchain-Based Cryptocurrency Price Changes 19](#_Toc173100124)

[2.10 Improved Bitcoin Price Prediction based on COVID-19 data. 20](#_Toc173100125)

[2.11 Causality between Sentiment and Cryptocurrency Prices 20](#_Toc173100126)

[2.12 Ensemble Deep Learning Models for Forecasting Cryptocurrency Time-Series 21](#_Toc173100127)

[Chapter 03 – Methodology 22](#_Toc173100128)

[3.1 Planning 22](#_Toc173100129)

[3.1.1 Identifying Project Aims 22](#_Toc173100130)

[Use Big Data for Analysis for Further Details 22](#_Toc173100131)

[Deliver facts to enable people to make the right decision. 22](#_Toc173100132)

[3.1.2 Identifying the Objectives 22](#_Toc173100133)

[3.1.3 Gantt Chart 23](#_Toc173100134)

[3.2 Analysis and Requirement Gathering 23](#_Toc173100135)

[3.2.3 Requirement Gathering 24](#_Toc173100136)

[3.3 Designing 24](#_Toc173100137)

[3.1.1 Social Media Sentiment Analysis System Design 24](#_Toc173100138)

[3.1.2 Bitcoin Prediction System Design 27](#_Toc173100139)

[3.4 Implementation 30](#_Toc173100140)

[3.4.1 Sentiment Analysis System 30](#_Toc173100141)

[3.4.2 Sentiment Analysis with Bitcoin Price Prediction System 42](#_Toc173100142)

[3.5 Testing 55](#_Toc173100143)

[Chapter 04 – Artifact 58](#_Toc173100144)

[The Line Paragraph 58](#_Toc173100145)

[Tabular Data 59](#_Toc173100146)

[Graphical Representation 60](#_Toc173100147)

[Observations 61](#_Toc173100148)

[Importance of Yearly Aggregation 61](#_Toc173100149)

[Chapter 05 – Conclusion 62](#_Toc173100150)

[05.01 Important Outcomes 62](#_Toc173100151)

[05.02 Limitations of the Project 62](#_Toc173100152)

[05.03 Critical Assessment of the Project 63](#_Toc173100153)

[05.04 Self-Reflection 63](#_Toc173100154)

[05.05 Future Works 64](#_Toc173100155)

[References 65](#_Toc173100156)

# Figures

[Figure 1 Gantt Chart 23](#_Toc173002706)

[Figure 2 Use Case Diagram - Sentiment Analysis 24](#_Toc173002707)

[Figure 3 DFD Level 0 Diagram - Sentiment Analysis 25](#_Toc173002708)

[Figure 4 DFD Level 01 Diagram - Sentiment Analysis 26](#_Toc173002709)

[Figure 5 Use Case Diagram - Bitcoin Prediction Analysis 27](#_Toc173002710)

[Figure 6 DFD Level 0 Diagram - Bitcoin Prediction Analysis 28](#_Toc173002711)

[Figure 7 DFD Level 1 Diagram - Bitcoin Prediction Analysis 29](#_Toc173002712)

[Figure 8 – importing the necessary libraries for sentiment analysis 30](#_Toc173002713)

[Figure 9 Last Wikipedia Comment Edit 30](#_Toc173002714)

[Figure 10 Get the first revision from Wikipedia edits 31](#_Toc173002715)

[Figure 11 First Wikipedia Comment Edit 32](#_Toc173002716)

[Figure 12 Natural language processing 32](#_Toc173002717)

[Figure 13 initializes an empty dictionary named edits and iterates 33](#_Toc173002718)

[Figure 14 calculates aggregate statistics for sentiment analysis 34](#_Toc173002719)

[Figure 15 converts the edits dictionary into a pandas DataFrame 35](#_Toc173002720)

[Figure 16 displays the contents of the edits\_df DataFrame 36](#_Toc173002721)

[Figure 17 Generates a range of dates starting from March 8, 2009, up to the current date 37](#_Toc173002722)

[Figure 18 Converting its index to datetime format 38](#_Toc173002723)

[Figure 19 Sentiment analysis data stored in the DataFrame 39](#_Toc173002724)

[Figure 20 Values of sentiment analysis data over a 30-day window 40](#_Toc173002725)

[Figure 21The rolling\_edits DataFrame export 41](#_Toc173002726)

[Figure 22 Retrieves historical price data for Bitcoin 42](#_Toc173002727)

[Figure 23 Showcases the historical price data for Bitcoin 43](#_Toc173002728)

[Figure 24 DataFrame btc is converted to the datetime 43](#_Toc173002729)

[Figure 25 Bitcoin’s closing prices 44](#_Toc173002730)

[Figure 26 Export Bitcoin’s closing prices data 44](#_Toc173002731)

[Figure 27 Sentiment Table 45](#_Toc173002732)

[Figure 28 the BTC DataFrame is merged with the Wiki DataFrame 46](#_Toc173002733)

[Figure 29 BTC Closing Price with Sentiment Prices 47](#_Toc173002734)

[Figure 30 imports the RandomForestClassifier 48](#_Toc173002735)

[Figure 31 Calculates the precision of the model's predictions 48](#_Toc173002736)

[Figure 32 Score of the model's predictions 49](#_Toc173002737)

[Figure 33 train predictors as features and train target 49](#_Toc173002738)

[Figure 34 Rolling backtest of the model using historical data 50](#_Toc173002739)

[Figure 35 importing the XGBClassifier module 50](#_Toc173002740)

[Figure 36 help of the next DataFrame predictions 51](#_Toc173002741)

[Figure 37 Computes precision of the model 51](#_Toc173002742)

[Figure 38 Precision\_score function output 52](#_Toc173002743)

[Figure 39 Calculate and add several rolling statistical features to the Bitcoin (**btc**) DataFrame for different time horizons 52](#_Toc173002744)

[Figure 40 Calculate and add several statistical features to the Bitcoin 53](#_Toc173002745)

[Figure 41 Precision scores in machine learning 53](#_Toc173002746)

[Figure 42 Final output of the sentiment 54](#_Toc173002747)

[Figure 43 export the DataFrame predictions 55](#_Toc173002748)

[Figure 44 Leverages historical data and sentiment analysis to forecast future BTC prices 58](#_Toc173002749)

[Figure 45 Visualization of the BTC price prediction system 59](#_Toc173002750)

# Acknowledgement

At first I would like to acknowledge the support and assistance of some colleagues in order to make this research work come into being. Firstly, I would like to thank my project supervisor Mr. Darshana Hettiarachchi, Senior Lecturer of the study. He encouraged me and gave me a helping hand to get this research done. He has been very useful for this entire project; His guidance was very relevant in the process of doing this. Furthermore, I would like to express my appreciation to Mrs. Vigini Mekala for her excellent supervisory support and for her major contribution to this research. I would also like to express my special thanks to my research coordinator, Mrs. Maduwanthi Uthpala, for her kind assistance to put our research to this stage. Finally, I would also express my deep appreciate to my parents with policies of encouragement they always make to me throughout my academics. The confidence reposed in me was one of the major stimuli for me Let me extend my heartfelt thanks to all of you for all the wonderful support you have given me.

# Abstract

The focus of this project is to enhance the prediction of cryptocurrency prices by utilizing sentiment analysis on social media and other advanced kinds of artificial intelligence. The project aims to develop a robust and comprehensive four-dimensional forecasting model by utilizing historical data, market coefficients, blockchain statistics, and sentiment analysis from platforms like Twitter. Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and XGBoost provide the potential to comprehensively capture the intricate patterns of the market owing to their intricate nature, hence ensuring a high level of accuracy and reliability. The research also examines novel methods of prediction and utilizes Big Data to evaluate substantial amounts of data and produce valuable insights and patterns. The end-users will derive advantages from the SEO reports, recommendations, and final analysis provided to the stakeholders in the dynamic and unpredictable bitcoin industry, enabling them to make informed decisions. The model demonstrates a high level of accuracy in reflecting the real-time fluctuations in the markets and provides insights into the efficacy of including sentiment analysis alongside or in combination with the quantitative measures for forecasting purposes.

# Chapter 01 – Introduction

## Background

Today, in large part to the ever-changing nature of cryptocurrency markets; they are being watched closely and have established themselves as subjects of financial analysis. These markets have become more complex and require new strategies for improving the predictions on prices. Having been a curious discoverer in the world of cryptocurrency and driven by an adventurous spirit that longs to break new ground, this project was motivated simply as part of the quest for exploring sophisticated techniques towards better prediction.

### 1.1.1 Importance of Integrating Social Media Sentiment Analysis

Altering market sentiment is essential in the world of crypto trading. Incorporating social media sentiment analysis gives another perspective into the current public view, an extra layer to traditional models. This project is about utilizing the power of sentiment analysis, to make hopefully better and more accurate cryptocurrency price prediction.

This effort is in good part due to the deep-rooted passion and desire for worthwhile participation in that domain, taking advantages of both innovations led by technology but also pure excitement as seen from other posts on various forums about it.

## 1.2 Aims of the Project

First and foremost, of the project is to build machine learning models capable of predicting prices associated with Over ledger. With the help of modern methods such as Long Short-Term Memory (LSTM), Gated Recurrent Unit, GRU and XGBoost we want to design models that are built upon an accuracy-first mindset. These models will be developed by forming historical data, in turn for the cryptocurrency price forecasting system to generate more accurate predictions. This objective points to the need for both technical discipline and ingenuity in developing robust predictive tools that can be trusted by such a turbulent market as cryptocurrency.

## 1.3.1 Academic Question

How can the integration of Social Media Sentiment Analysis, advanced machine learning techniques, and Big Data analytics enhance the accuracy and reliability of cryptocurrency price predictions, providing a comprehensive understanding of market dynamics and offering actionable insights for informed decision-making in the volatile digital asset landscape?

## 1.3.2 Objectives

**Evaluate the Impact of Social Media Sentiment Analysis**

This goal is to do an in-depth study with respect to implications of sentiment analysis from social media platforms into the cryptocurrency price prediction model. The review evaluates the extent to which augmenting predictions with real-time public sentiment enhances their validity too. The project aims to assess the worth and importance of social media sentiment in improving prediction results by examining its effect on a variety of cryptocurrencies, timeframes as well as market types.

**Implement Advanced Machine Learning Techniques**

The progress of this objective is concentrated around employing powerful machine learning techniques to better the Cryptocurrency Price prediction model. Through such approaches as deep learning, ensemble methods, and hyperparameter tuning, the project’s purpose is to better the model performance. The focus is on discovering minor relationships in data relating the cryptocurrencies, that is why the corresponding model is adjusted for the market’s high volatility.

**Integrate Big Data Analytics for Comprehensive Insights**

By utilizing the concept of Big Data, it is the purpose of this objective to undertake the analysis of enormous sets of data including parts of the market, block chain, and social media attitudes. Optimizing these many forms of data sources is the approach to gain a full and varied view of the bitcoin industry. The goal is to examine enormous collections of data to uncover links, trends, and outliers on the complete scale of big data to enrich the degree of decision-making on anticipation.

**Explore Novel Forecasting Strategies**

The achievement of this objective includes looking at and adopting approaches of predicting business outcomes which are not restricted to the financial domain. The suggested effort attempts to question several dimensions of data spanning from different indices to outside impacts. The fundamental idea of the more progressive approach of the model is to come up with tactical and consistent ways of the cryptocurrency market peculiarities to boost the resilience and flexibility of the prediction model.

**Provide Actionable Insights for Decision-Making**

This is a strategic way of arriving at a model that not only predicts the bitcoin prices but one that provides recommendations as well. This purpose is to provide investors, traders and stakeholders with information that will aid them beyond processes of predictions. The project seeks to provide management consulting and analysis based on the created mathematical model and its implementation in the unpredictable world of digital assets. Subsequently, it varies from other types of intelligence where the main focus is put on translating forecasts into meaningful and beneficial knowledge which can support decision-making processes.

## Scope

### 1.4.1 Data Sources

The scope is gathering the data from diverse sources, assembling the information list to deliver the result of the analysis. This involves historical price data to capture the prior trends in the market, market condition data to capture general conditions of market, block chain data to catch the transactional patterns, and sentiment data from social media. The objective is to design a parsimonious yet comprehensive and, consequently, multi-dimensional data collection platform specific to nearly every aspect of cryptocurrencies.

### 1.4.2 Machine Learning Techniques

This area includes the application of sophisticated machine learning algorithms to anticipate the relationships of variables, as well as of bitcoin values. Some methodologies to be applied in the project include deep learning approaches, ensemble learning, and optimization methods to give meaningful and trustworthy patterns and dependencies in the data to increase the accuracy and robustness of predictions.

### 1.4.3 Social Media Sentiment Analysis

As one of the project objectives, the sentiment analysis of the social media debates will have to be included. This has a correlative component that involves ascertaining the effects of perception and mood of the public on the cryptocurrency markets. The research involves sentiment analysis of the data gathered from sites like Twitter to reveal the relationship between public attitude on various themes expressed through social media and the swings in price, so providing to quantify the impact of public sentiment on the rates.

### 1.4.4 Big Data Analytics

This topic focuses on the application of massive Data analysis for handling massive data on components of the bitcoin ecosystem. It will be defined by the employment of extremely effective and flexible analytical devices for dealing with large data, which will guarantee the examination of all the essential tendencies, patterns and irregularities in the marketplace. This is because Big Data analytics will assist in making meaning out of the vast amount of information acquired from the Cryptocurrency marketplaces.

### 1.4.5 Innovative Forecasting Strategies

The scope comprises a search for and use of other forecasting concepts that are different from strictly financial ones. This comprises to become involved in such areas as different data dimensions, some additional indications and to specify the method to correspond to the maximal and highly changeable digital currency industry. It is for this reason that the existing methods of forecasting are to be upgraded in order to strengthen the adaptability of the prediction model taking into consideration the existing high volatility of the market.

### 1.4.6 Actionable Insights

The rules also mention that discussing recommendations and specific actions is one of the essential components of the project. In addition to projecting the price of cryptocurrencies, the project’s purpose is to explain how to adopt the advised tactics based on the constructed model. Investor and traders will get data which will not just provide plain crude projection that may be difficult and deceitful to read in the fluid and often unexpected digital asset market. The focus is on producing predictive analytics that have immediate applicability and directly benefit an organization’s mission.

## Functional And Non-Functional Requirements

### 1.5.1 Functional Requirements

The system shall be able to generate the effect of a cryptocurrency based on the sentiment data set employed and enable comprehension of the differentiated nature of sentiment in the market. It will also develop effects that evaluate cryptocurrency using the newest methods of machine learning and massive data processing. Further, the system should have the feature of unique prediction that is much beyond the current range of the existing prediction systems. Finally, it aims to provide prescriptions for relevant managers and decision-makers so that the users of the system may make the proper decisions out of it in the end.

### 1.5.2 Non-Functional Requirements

Many fundamental non-functional needs have to be achieved for an ideal system that is efficient and satisfying to the users. The accuracy of the forecasts must be very high while at the same time it has to be highly consistent in order to be trustworthy. Real-time analysis is particularly relevant in the fast-ever-changing field to deliver insights that can be valuable at the moment. Another component is simply convenience, switching between devices and platforms should be easy with the system. Last but not least, the factor of usability is recommended to be taken into attention as the system must be as easy to use as feasible for absolutely any user disregarding his or her level of IT literacy.

## Report Structure

### Chapter 01 – Introduction

In the first chapter, the authors discuss how markets associated to cryptocurrencies are continually evolving and describe the significance to develop ways for boosting the accuracy in price prediction of cryptocurrencies. It illustrates why social media sentiment research should be incorporated in the conventional forecasting models by emphasizing the advantages of employing real-time public sentiments from Wikipedia. The chapter also defines the primary objectives of the project among them are construction of feasible machine learning models and the usage of complicated algorithms like LSTM, GRU, and XGBoost. It also specifies the study topics and particular objectives, which include analyzing the utility of sentiment analysis, applying new machine learning methods, incorporating Big Data analysis, identifying new strategies in forecasting, and providing appropriate results. The project’s objectives are carefully outlined with regards to where data will be sourced, the types of machine learning that will be involved, where sentiment analysis will take place, the use of Big Data analytics for the project, as well as innovative approaches to forecasting for the corporate entity in question. Also, the chapter describes the functional and non-functional requirements that should be satisfied in the system to make it effective and meet users’ expectations.

### Chapter 02 – Literature Review

Literature review chapter offers broad synthesis of the most referred and significant works and papers about the cryptocurrency price prediction through the machine learning, deep learning, and sentiment analysis methodologies. It addresses numerous methodologies employed in the prior researches and their merits and cons. The chapter puts down the background of the research assessing the efficiency of sentiment analysis in improving the present approach for price prediction and highlights the research gaps which this project attempts to fill. Thus, describing the research subject, the chapter provides the groundwork for the study in terms of its connection to earlier research and techniques.

### Chapter 03 – Methodology

The methodology chapter provides the framework of the study along with how the project was performed. The part commences with the explanation of the scope of the project, including the aim and objective of the research; the graphic that displays the project chronology is added here. The chapter, then gives the technique for identifying the functional and non-functional needs with the help of Primary & Secondary data collecting. Recommended solutions are explained in detail together with use case diagrams and data flow diagrams of the social media sentiment analysis system and the Bitcoin prediction system. Thus, in the implementation part, the activities being performed to gather data, clean this information, train the models, and combine sentiment analysis with the price forecasts are discussed. Last of all, the testing methodologies together with the test cases that could implemented so as to ensure that the system under development fulfills all the needed specifications are given and so, the project development phase is well covered.

### Chapter 04 – Artifact

The artifact chapter discusses the system that was established in the project for estimating the prices of Bitcoin currency. It uses graphic graphics and particular data detailing the choice made by the model in order to present the system. Furthermore, both daily and annual aggregated data are supplied, where real closing prices are presented for comparison with the anticipated ones to demonstrate the model’s correctness. In this section, the performance & accuracy of the system is examined along with explained use of sentiment analysis along with other complicated forms of machine learning. They are delivering actual outcomes of the research to the stakeholders and therefore, tools to attain successful judgments in the context of cryptocurrencies.

### Chapter 05 – Conclusion

Conclusion chapter provides the primary findings of the research and underscores the attractive prospects of employing social media sentiment analysis, complemented with the state-of-the-art machine learning algorithms and Big Data analytics in the field of cryptographic currencies’ price predictions. This study outlines the potential usefulness of this model to market stakeholders, affirming the model’s preparation to take on more variables that can effect the dynamics of the cryptocurrency market, including cyclical patterns for the short-run and secular trends for the long-run. Some of the ideas for future research and possible enhancements to the proposed prediction model are expanded in the chapter as well as offering confirmation of the usefulness of ongoing advancements in data analysis and knowledge processing systems. In conclusion, this chapter elucidates the significance of the project in developing the area of cryptocurrency price prediction through providing useful instruments for investing businesses, traders, and other interested persons.

# Chapter 02 – Literature Review

## 2.1 On Forecasting Cryptocurrency Prices - A Comparison of Machine Learning, Deep Learning, and Ensembles.

In their landmark 2023 study, "On Forecasting Cryptocurrency Prices: To this interview, ‘A Comparison of Machine Learning, Deep Learning, and Ensembles,’ Andrea Rossi, Diego Carraro, and Kate Murray along with Andrea Visentin provide a novel view on forecasting Cryptocurrencies’ price. It contributes to the understanding of the fundamental challenges in the field: their analysis exposes the shortcomings of earlier studies that tend to focus on the most recognizable cryptocurrencies and do not have standard comparison methodologies. Credibly, they establish the supremacy of DL approaches with a lean towards LSTM model to existing statistical and virtually all the machine learning methods with the amazing average RMSE of 0. TO 0222 and Mean Absolute Error (MAE) of 0. 0173. Thus, while the researchers pay considerable attention to the replicability of the findings, they make the dataset and code open for further examination. As a further development beyond the study, a comparison analysis applies more sophisticated machine learning algorithms, as well as additional external data such as qualitative social media attitudes, which will grow into an all-encompassing market viewpoint for the users. This extended model does not only anticipate the prices but also delivers appropriate suggestions establishing awareness of the existing models for predicting bitcoin values, which makes it more useful[.(Murray et al., 2023)](#_References)

## 2.2 Predictability of cryptocurrency returns - evidence from robust tests.

The study by Siyun He and Rustam Ibragimov offers the use of proper and accurate statistical tools in the examination of cryptocurrency returns and prices in this quite challenging market. They also stress indicative methodologies whose probe of significant predictors involves the t-statistic tests. However, the project is different by employing machine learning algorithms, which concerns about the real-time sentiment analysis of the market. While He and Ibragimov focus on the quantitative analysis of historical data, this method includes these additional qualitative questions to help the schools’ personnel to make decisions. Nevertheless, both research works point to the fact that adequate methods are critical in evaluation of performance in the field of cryptocurrencies. Strengthening the basic knowledge founded by these people, the project incorporates complex procedures and real-time data to improve the cryptocurrency price prediction models contributing to development in the realm. [(He and Ibragimov, 2022)](#_References)

## 2.3 Cryptocurrency Price Prediction and Trading Strategies Using Support Vector Machines

The newsworthy research works of David Zhao, Alessandro Rinaldo, Christopher Brookins are mainly based on the short-term price forecast of major cryptocurrencies, including Bitcoin, Ethereum, and Litecoin. Their results are rather impressive in terms of 1-hour price prediction, including sharp classification performance and highly profitable trading performance, especially for the SVM builds that made $131,334 of profitable trades against the market from September 2011 to July 2013. While, to project it might enlarge targeted prediction intervals from daily to monthly and include the data from different sources including the sentiment analysis of the social networks and the data on the market news for a wider perspective. Also, while their vision is set on 1-hour forecasts, the project can investigate diversified trading techniques that might fit different periods of investment. [(Zhao, Rinaldo and Brookins, 2019)](#_References)

## 2.4 A Deep Learning-Based Cryptocurrency Price Prediction Model That Uses On-Chain Data

This path-breaking study by Kim, Shin, Choi, and Lim proposes a new framework for forecasting the Bitcoin (BTC) price while allowing for more complex change point detection and the incorporation of on-chain data in the model’s training process. Their proposed Self-Attention Based LSTM (SAM-LSTM) is effective in addressing the temporal relations and significant characteristics within the data; they achieve better performance in terms of several evaluation measures. However, while being devoted to Bitcoin only, the project is to have a more generalized scope and include several cryptocurrencies. Moreover, while they work with data concerning Bitcoin within blockchain, the work is focused on various data, such as the sentiment of social platforms and news texts for considering the market in its entirety. Also, they use LSTMs with self-attention, but the project could consider other deep learning models that are suitable for the characteristics of the dataset and the type of predictions. [(Kim et al., 2022)](#_References)

## 2.5 Predicting Bitcoin Prices Using Machine Learning

Athanasia Dimitriadou and Andros Gregoriou use the Machine Learning to predict the Bitcoin and refute the Efficient Market Hypothesis. They have collected 24 features, and amongst classification models, the best result of accuracy is 66% using Logistic Regression with 5-Fold Cross Validation which is slightly better than SVM and Random Forest. The project should be able to generate even better results than these; I will possibly use deeper learning or try to integrate other variables such as social media sentiments or other Bitcoin-related data that may exist on the blockchain. Furthermore, I’ll use their EMH results to examine their short-term prediction on various time horizons. [(Athanasia Dimitriadou and Andros Gregoriou, 2023)](#_References)

## 2.6 Forecasting of Cryptocurrency Prices Using Machine Learning

Andriy Viktorovich Matviychuk, Vasily Derbentsev, and Vladimir Soloviev has the research effort that seeks to anticipate the change in cryptocurrencies’ prices in the 90-day time horizon using ML. They examine BART, MLP Neural Networks, and RF ensembles, and reach satisfactory performance with MAPE below 3 on average. within 5% of the ideal for BART and MLP, and within 5% for RF. Their algorithms, on the other hand, focus simply on the price of prior cryptocurrencies as the predictors; the project may broaden the prediction window and include other data such as the sentiment analysis of social networks or news analysis. Also, despite their evidence that their models may be applied to trading, the project might look into such trading situations based on the choice of the prediction horizon and the output of the offered models. [(Vasily Derbentsev, Andriy Viktorovich Matviychuk and Vladimir Soloviev, 2020)](#_References)

## 2.7 Predicting Cryptocurrency Prices with Machine Learning Algorithms

Reading the work by Harsha Nanda and Khetan Venkata Ratnam, the researchers utilize Long Short-Term Memory (LSTM), stating that it is one of the most efficient Machine Learning (ML) models for long-term Bitcoin price forecasting, including Relative Strength Index (RSI), Exponential Moving Average (EMA), and Simple Moving Average (SMA). Thus, although their study studies crucial parameters and the effect of the moving averages, big-data effort expands upon their research. They are centered around LSTMs, but, work targets a range of other progressive deep learning approaches including bagging for higher accuracy. Furthermore, I present precise technical indications based on the chosen asset and use all sorts of feature engineering. Furthermore, in contrast to the historical pricing, project combines social media sentiment and blockchain data as some of the sources of information about Bitcoin price besides offering more comprehensive dynamics of Bitcoin price and boosting the predictiveness of the algorithm. [(Gudavalli, Harsha Nanda and Ratnam, 2023)](#_References)

## 2.8 Cryptocurrency Price Prediction using Machine Learning

Guru Pradeep G, Harishvaran M, and Amsavalli K's research acknowledges the challenges of Bitcoin price forecast, noting volatility and external influences. While exploring regression, time series analysis, and neural networks, they touch on historical price data and hint at social media sentiment analysis but lack clarity in data exploration. Areas for improvement include evaluation metrics and comparative analysis. In contrast, approach mixes multiple data sources like social media sentiment, traditional market determinants, and on-chain data, delivering a holistic framework. I examine a larger spectrum of machine learning approaches, including deep learning, for big data optimization. Despite volatility, technique tries for improved accuracy utilizing metrics like Mean Squared Error (MSE). While simpler methods exist, complexity offers exact predictions crucial for investors. Through data refinement and interpretable models, I strive to boost performance and acquire insights into prediction determinants. [(Guru Pradeep G, Harishvaran M and Amsavalli K, 2023)](#_References)

## 2.9 Time Series Analysis of Blockchain-Based Cryptocurrency Price Changes

Jacques Fleischer, Gregor von Laszewski, Carlos Theran, and Yohn Jairo Parra Bautista's research efficiently leverages Long Short-Term Memory (LSTMs) for assessing sequential bitcoin price data, applying Root Mean Squared Error (RMSE) for model evaluation. However, enhancements could include increasing data quality checks and comparing LSTM performance with alternative Recurrent Neural Network (RNN) designs. In contrast, technique stretches beyond past pricing to include social media, traditional market data, and on-chain variables for a holistic forecast landscape. While they focus on LSTMs, I investigate a larger range of approaches ideal for big data research. Despite wrestling with market volatility, methodology may require additional evaluation methods like feature importance analysis to boost prediction accuracy[. (Fleischer et al., 2022)](#_References)

## 2.10 Improved Bitcoin Price Prediction based on COVID-19 data.

Thus, Palina Niamkova and Rafael Moreira’s research addresses societal change, especially COVID-19, about its influence on the Bitcoin demand and price. They intersect the data on the Bitcoin price with the data on COVID-19 cases/deaths, to which they performed the feature significance analysis using Random Forest, and the price prognosis using LSTM. Their work explores LSTM performance based on \(w/c\) ratios with and without COVID features, while underlining the increased precision related to the latter. Nevertheless, there are several limitations including the effects of external variables on the price of Bitcoin and the applicability of the study outside COVID-19 that remain unaddressed. When it comes to comparative analysis, although their study covers COVID-19 influence, endeavor gives a more comprehensive picture by merging different databases. I investigate alternative machine learning algorithms different from LSTMs and verify a wider set of characteristics that affect the Bitcoin price swings, maybe boosting the model’s applicability. Both projects have issues in studying market volatility, whereas mine might need supplementary approaches such as feature importance analysis for the evaluation of the data source’s impact on the price prediction. [(Niamkova and Moreira, 2023)](#_References)

## 2.11 Causality between Sentiment and Cryptocurrency Prices

In his research, Mondal, Raj, S, Gowsik S B, P S, and Chandra try to examine the cryptocurrency discussion on the Twitter platform using Topic Modeling and Sentiment Analysis to uncover numerous subjects and effects on the cryptocurrency market values. However, the study does not disclose how much Twitter data was used, during what timeframe, hence the conclusions of the study may not be totally typical. At the same time, the addition of sentiment analysis of social media, basic market data, and on-chain data provides a broader viewpoint to big data techniques. Unlike most current models that utilize only two statistical data sets and try to uncover trends alone, technique is aimed to study interconnections between diverse forms of information, maybe discovering cause-and-effect links. Furthermore, some data sources contain a greater range of possibilities, therefore boosting the potential to generalize conclusions for multiple cryptocurrencies and market conditions. [(Chandra, 2023)](#_References)

## 2.12 Ensemble Deep Learning Models for Forecasting Cryptocurrency Time-Series

Solutions for stock exchange prediction is proposed by Ioannis E. Livieris, Emmanuel Pintelas, Stavros Stavroyiannis, and Panagiostis Pintelas: regression and classification for developing the deep learning models for forecasting the cryptocurrency prices using the ensemble learning. As they focus on model reliability, approach is more versatile since it involves the mixing of social media data, traditional markets, and blockchain data. I sum up more ways than LSTMs and convolutions while emphasizing on interpretability and utility in the actual world. [(Ioannis Livieris et al., 2020)](#_References)

# Chapter 03 – Methodology

## 3.1 Planning

### 3.1.1 Identifying Project Aims

Enhance Prediction Accuracy

* Utilize technical and strategical strategies in constructing a machine learning model to improve the bitcoin price predictions.

.

Incorporate Social Media Analysis

* Composite sentiment analysis of social media to map and harness the shared temper, and prevailing sentiment of the cryptocurrency fraternity.

Use Big Data for Analysis for Further Details

* Utilize Big Data processing techniques to analyze extensive data on market indicators ranging from block chain, social media data and other indicators that may affect the market.

Deliver facts to enable people to make the right decision.

* Build a model that helps in prediction of the price of the cryptocurrencies and also provides recommendations to investor, traders and stakeholders on what action they should take in the unstable market of the cryptocurrencies.

### 3.1.2 Identifying the Objectives

* Evaluate the impact of Social Media Sentiment Analysis.
* Implement Advance Machine Learning Technique.
* Explore Novel Forecasting Strategies.
* Integrate Big Data Analytics for Comprehensive Insights.
* Provide Actionable Insights for Decision Making.

### 3.1.3 Gantt Chart

A screenshot of a project

Description automatically generated

Figure 1 Gantt Chart

## 3.2 Analysis and Requirement Gathering

#### 3.2.1 Functional Requirements

* The system shall be able to output certain impact of a cryptocurrency based on sentiment data sets used.
* The system shall be able to generate a certain impact on cryptocurrency using machine learning techniques and big data analysis.
* The system shall be able to provide the ability to predict novel predictions.
* The system shall be able to provide actionable insights for stakeholders of the system.

#### 3.2.2 Non-Functional Requirements

* Accuracy
* Realtime
* Portability
* Usability

### 3.2.3 Requirement Gathering

1. Primary research

* Direct observation of Market trend in Trading View.

1. Secondary Research

* Data Sources
  1. Yahoo Finance
  2. Wikipedia Data
* Bigdata Analysis Projects

1. Geeks For Geek
2. GitHub

## 3.3 Designing

### 3.1.1 Social Media Sentiment Analysis System Design

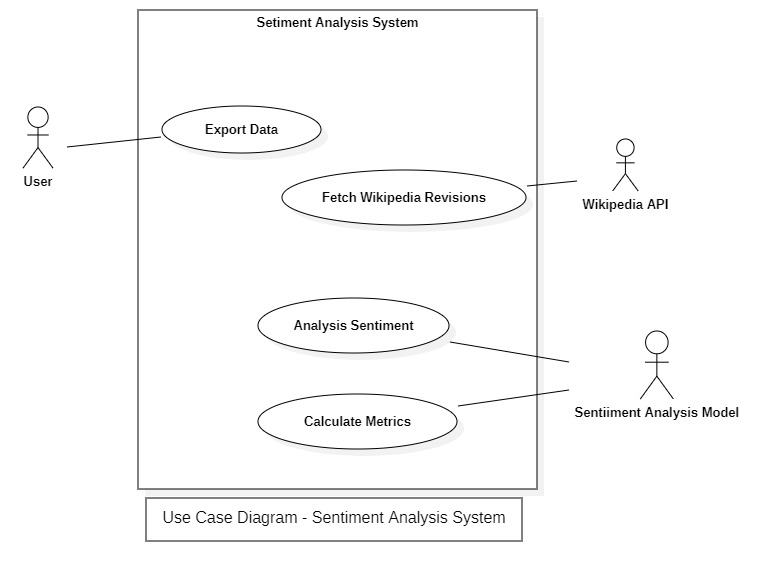


Figure 2 Use Case Diagram - Sentiment Analysis

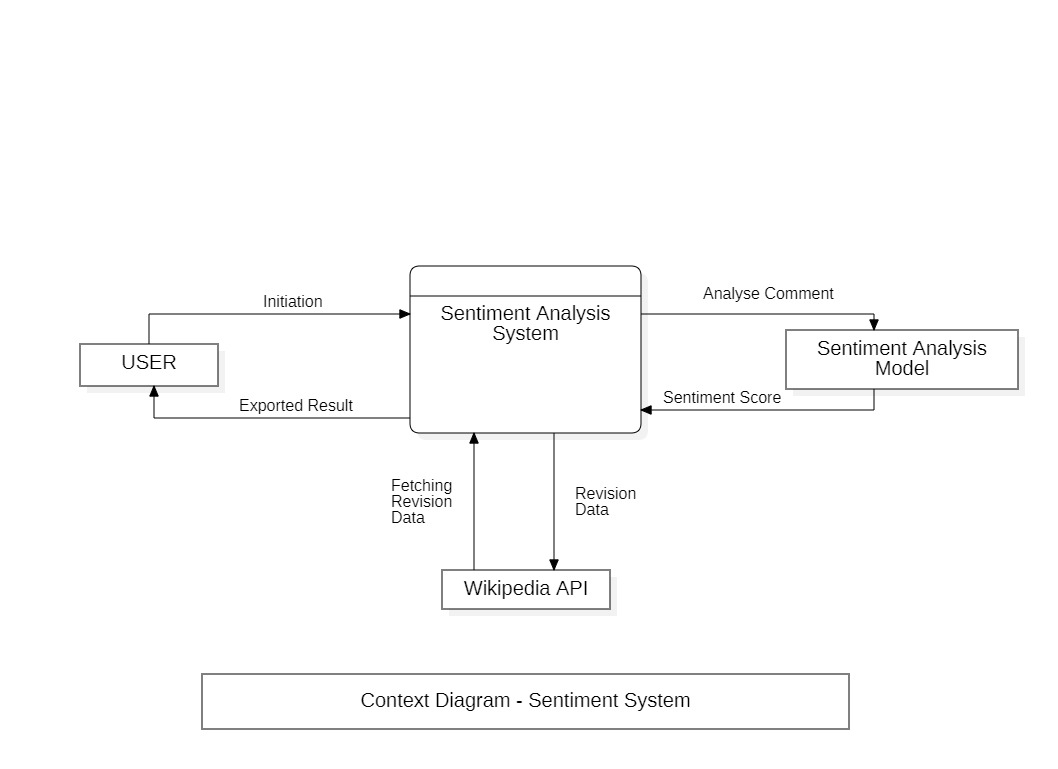


Figure 3 DFD Level 0 Diagram - Sentiment Analysis

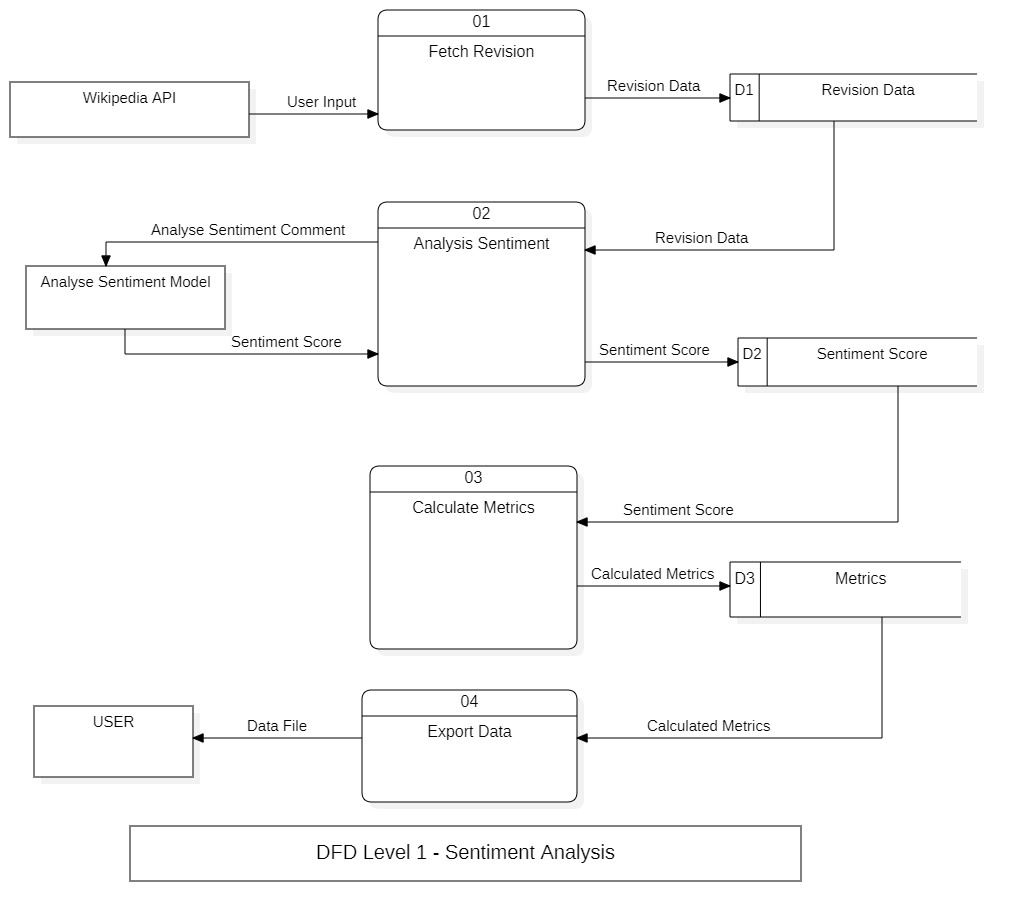


Figure 4 DFD Level 01 Diagram - Sentiment Analysis

### 3.1.2 Bitcoin Prediction System Design

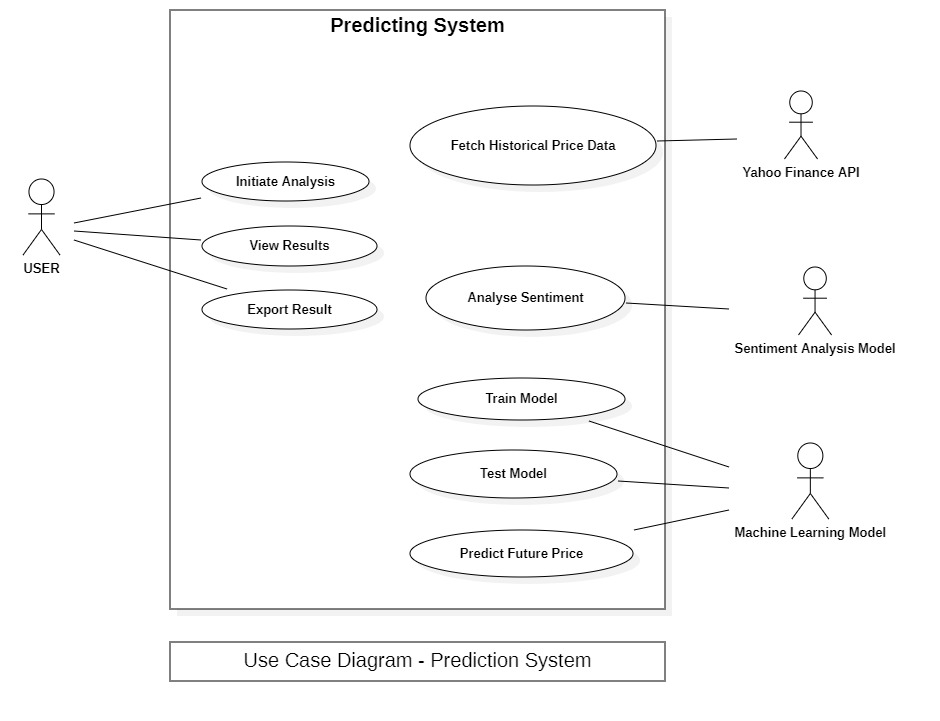


Figure 5 Use Case Diagram - Bitcoin Prediction Analysis

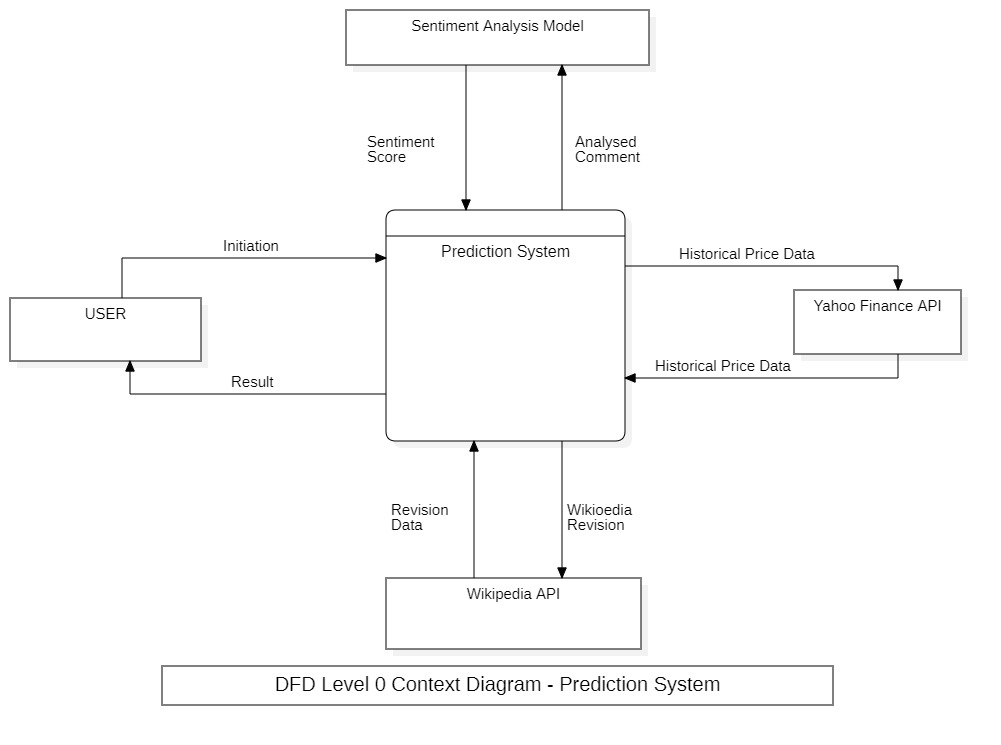


Figure 6 DFD Level 0 Diagram - Bitcoin Prediction Analysis

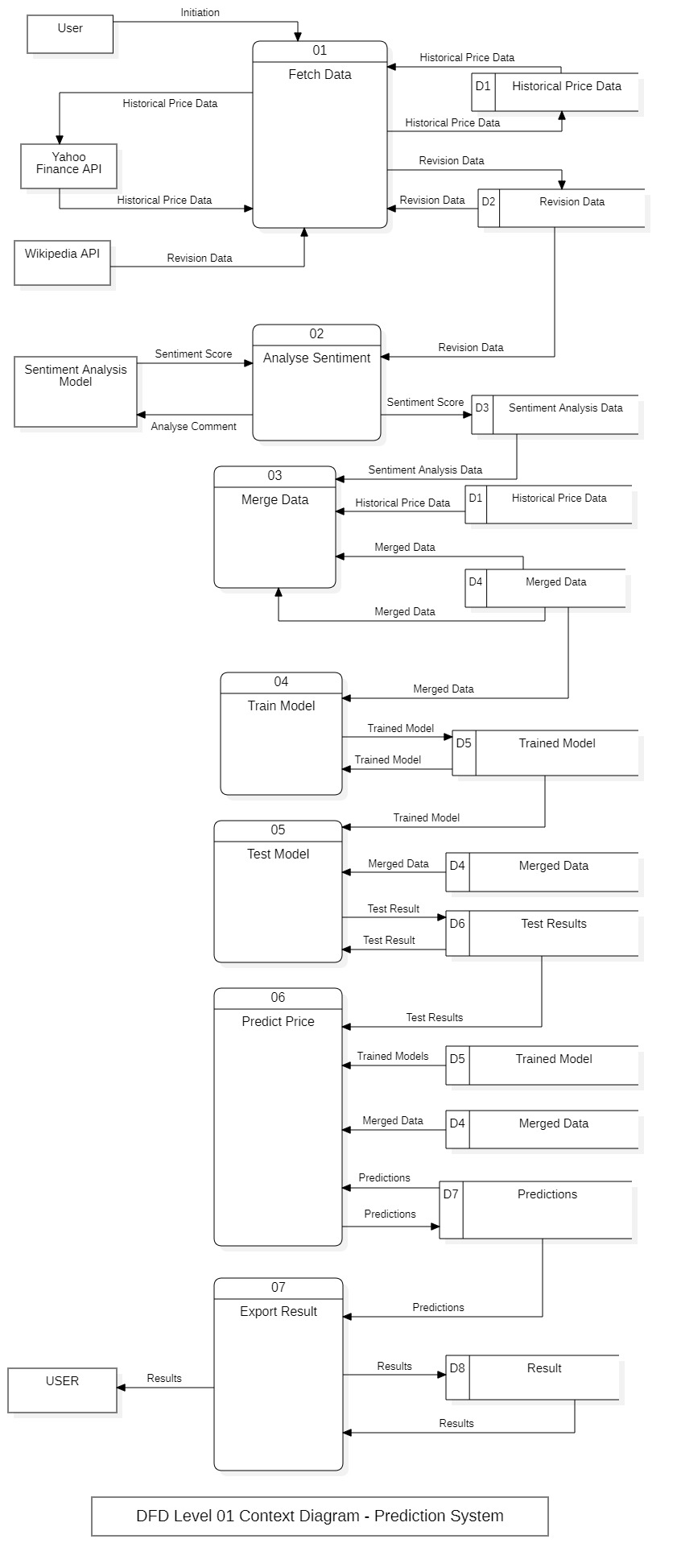


Figure 7 DFD Level 1 Diagram - Bitcoin Prediction Analysis

## 3.4 Implementation

### 3.4.1 Sentiment Analysis System

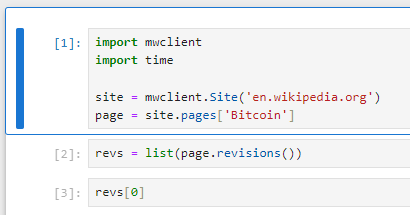


Figure 8 – importing the necessary libraries for sentiment analysis

This segment of code begins by importing the relevant libraries: mwclient for the operations on the Wikipedia API and time for time related activities. The site variable launches the session using mwclient with the connection to the English Wikipedia. Site. Then, it fetches the page object for “Bitcoin” at the Wikipedia site and stores the value in to page variable. The revs variable then fetches all the changes that has been done to the “Bitcoin” page by utilizing the revisions() function on the page object. To access the initial revision in this list, revs0 is used. It is important for gathering the prior #edit history of the selected Wikipedia page, which will be later used for assessing the sentiment of articles and forecasting the specified cryptocurrency prices..

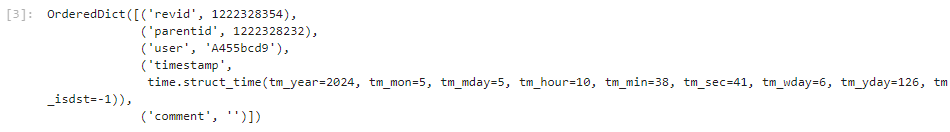


Figure 9 Last Wikipedia Comment Edit

The output that is delivered is in the form of OrderedDict of a standard dictionary in Python that keeps the order of keys while inserting them. This OrderedDict is an object and corresponds to a revision of the ‘Bitcoin’ article in the Wikipedia project. These parameters renewed in the form of key value pairs offer precise information about this specific revision. The ‘revid’ key provides the characteristics of the current revision identification number (1222328354) and the ‘parentid’ parameter has the prior identification number of the revision (1222328232). The ‘user’ key shows the user that conducted the change and, in this case, ‘A455bcd9’. To sum it up: the ‘timestamp’ key includes a time. The other attribute of the revision history is struct\_time object displaying the time when the revision was made, which in the case of the above code is May 5, 2024, at 10:38:41 AM. The ‘comment’ key on the other hand which is null here normally holds the edit summary or comment that was submitted by the user at the time of the revision. This is crucial in record keeping pertaining to modifications performed, examine the manner users are interacting, and conduct analysis of sentiment on the Wikipedia page edits.

A screenshot of a computer

Description automatically generated

Figure 10 Get the first revision from Wikipedia edits

The provided code sort the list of the revision entries (revs) by their time and then get the first revision. The line revs = sorted(revs, key=lambda rev: rev["timestamp"] arranges the list of revision dictionaries in ascending order according to their keys, which are ‘timestamp’. This is done through the . sorted method with a lambda function that takes the ‘timestamp’ from the dictionary and sorts the revisions list on it. After sorting, the variable revs of the list is adjusted to hold the persons in the list in this chronology. The expression revs[0] revives the sorted list of revisions to get the initial revision in terms of the timestamp. This procedure is significant concerning the evaluation of events and trends of the Wikipedia article ‘Bitcoin’ since it determines the first state that can be utilized for comparison. By retracting it to the first revision it gives a starting point for the sentiment analysis and comparison with other of the revisions.

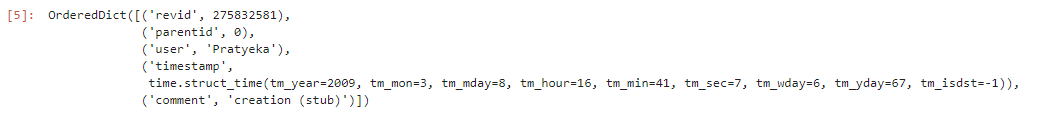


Figure 11 First Wikipedia Comment Edit

The given output is an OrderedDict that holds the data about a revision entry of the Wikipedia page ‘Bitcoin’. The key-value pairs in the OrderedDict detail more of this revision attributes. The ‘revid’ key contains reference number for the current revision (275832581) whereas ‘parentid’ key has the revision no: of the preceding revision which is zero in this case as it is the first revision. The key ‘user’ points to the user, namely ‘Pratyeka’, who displayed the information in a changed method for the program. The fact that the ’timestamp’ key probably includes a time can be determined from the content. struct\_time object generated and organized as it represents the time stamp of the changes made on March 8, 2009 4:41:07 PM. The ‘comment’ key provides supplementary information indicating the type of revision, revealing that it was the development of the Wikipedia page, ‘creation (stub)’. The first revision for the word ‘Bitcoin’ in the Wikipedia is this, which created ‘Bitcoin’ page and gave it a status of a stub article laying a ground for the future edits.

.

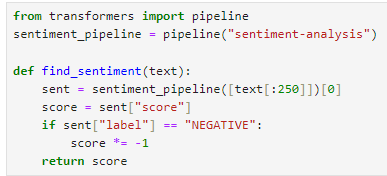


Figure 12 Natural language processing

Regarding the supplied code sample, it is associated with sentiment analysis that uses the Hugging Face Transformers library. Firstly, it imports this pipeline function from the transformers library which is utilized to bring out ready to use NLP models. Then, it submits a request to build a new pipeline for sentiment analysis by piping to pipeline with the argument “sentiment-analysis”. This pipeline employs a model which is pre-trained for sentiment analysis, hence it was picked on purpose. To discover the required sentiment for a specific textual content, the function find\_sentiment is constructed. An input text is accepted, stripped to the first 250 characters (due to the restrictions of the model’s input) and submitted to the sentiment analysis stream. The sentiment pipeline returns as an array of dictionaries with the results of the performing sentiment analysis of the input text. Because just one text input is available, [0] is utilized to acquire the first (and only) item of the list. The score of the sentiment is derived from the dictionary as sent[‘score’] which gives the intensity or polarity of the feeling. If the script is negative that is if sent[“label”] = “NEGATIVE”, then the score is increased by -1(score \*= -1). Finally the emotion score is returned out of the program. This piece of code explains how one may use pre-trained models for sentiment analysis to reduce on resource use when determining the sentiment of a given text..

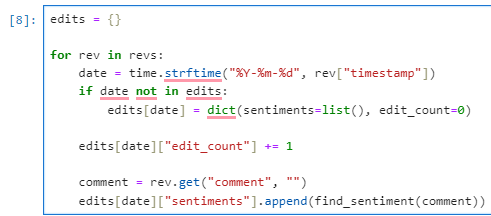


Figure 13 initializes an empty dictionary named edits and iterates

The present code segment constructs a new empty dictionary named edits and it processes a list of some revision entries (revs). For each revision item, it isolates the date from the timestamp, then convert the date to a string in the format of "%Y-%m-%d" through time. strftime(). Then, in the event of a first\_start\_date as the formatted date the script checks if the formatted date is found in the edits dictionary as a key. If there is none, new key-value pair is added to the edits dictionary where the first argument is the date. This entry includes a dictionary with two keys: The most significant parameter is the “sentiments” list which is created empty and “edit\_count” equals to 0.

After this, the sentiment of the revision comment is checked utilizing the find\_sentiment() function and the sentiment turn is added to the ‘sentiments’ list according to the specified date. Also, for that date value of “edit\_count” is added 1 to keep count of how many times it was edited on that specific date. This step is repeated for each revision element in the revs list of each current version of the file. On balance, this code segment classifies revision comments and their feelings by date and records the total number of revisions performed on every date.

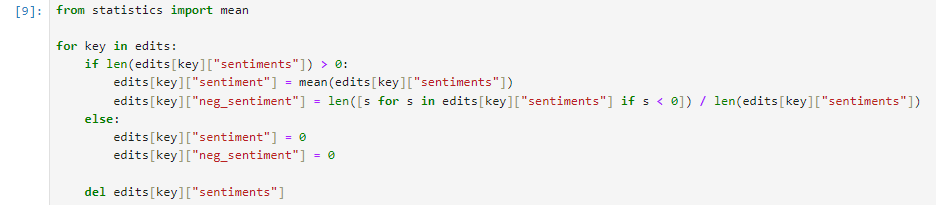


Figure 14 calculates aggregate statistics for sentiment analysis

This code segment sums variables important to aggregated statistics of the sentiment analysis kept in the edits dictionary. It can loop through each key in the dictionary called edits where the keys are dates. For each date, it compares the recent dates to program and verify if the sentiment score exists in the “sentiments” list. If there are sentiment scores present for that date (i. e. , if the length of the "sentiments" list is larger than 0), it computes two statistics: In this situation, the evaluated criteria comprised the mean sentiment score and the percentage of negative sentiments.

* The mean sentiment score is generated with the help of the mean() function from the statistics’ module. It is determined by the sum of all sentiment scores that have been put on that specific day and divide the result with the total number of sentiment scores. The mean sentiment score generated above is assigned to the ‘sentiment’ key in the date entry of the said edits dictionary.
* Number of addends having negative sentiments is computed as the total number of sentiment scores acquired and whereby the number of negative sentiment scores- sentiment scores being less than zero-and the overall total number of sentiment scores recorded on that specific date are split. This proportion is a measure of the negative feelings or the ratio of the negative sentiments to the number of all the sentiments that were ascertained in that date. The acquired value is kept with the help of the key “neg\_sentiment” and is assigned to the date item of the edits dictionary which corresponds to the date of the edit.

In the instance when there is no recorded sentiment for a certain date that is, when the “sentiments” list has no entries, the keys for ‘sentiment’ and ‘neg\_sentiment’ are set to zero for that particular date entry. Last of all, the “sentiments” list is popped out from each date entry to minimize the memory consumption and to align the edits dictionary into a plain shape. Collectively, this code section gathers sentiment analysis outputs depending on date and provides descriptive statistics to give a sense of the widespread sentiment expressed over the provided timespan.

A screenshot of a computer

Description automatically generated

Figure 15 converts the edits dictionary into a pandas DataFrame

This code piece transforms the edits dictionary into a pandas DataFrame with the name of ‘edits\_df’. To create the DataFrame the from\_dict() function from the pandas library is applied. The orient=’index’ attribute implies that the dictionary keys are dates from the ‘edits’ variable and they should be used as the index of the DataFrame.

The DataFrame’s index represents the date, whereas the columns are created from the keys of the dictionary. Particularly, the new columns are generated for calculation of “edit\_count”, “sentiment” and “neg\_sentiment”. The values under each column are the values of dictionary edits for each date.

The next cell creates this DataFrame and stores it in variable edits\_df, so edits\_df is now a structured representation of the results of sentiment analysis by date, with columns for the edit count, mean sentiment score, and proportion of negative sentiments. It can be processed or portrayed further in any way that fits the need of the data analyst or served in other analytic processes and needs.

A screenshot of a computer

Description automatically generated

Figure 16 displays the contents of the edits\_df DataFrame

The last output example shows the content of edits\_df DataFrame, in which different sentiments are summed up by the date they belong to. Each row represents a specific date, and the columns display the corresponding values for three attributes:Each row represents a specific date, and the columns display the corresponding values for three attributes:

1. edit\_count: Shows the number of edits that has been done on each date.

2. sentiment: Stands for the average of the sentiment Scores of comments made at that particular day. This score is expressed at -1 (negative sentiment) to 1 (positive sentiment).

3. neg\_sentiment: Shows the percentage of the negative emotions when all emotions are estimated by the corresponding date. This value lies in the interval 0 to 1; at 0, no negative sentiments were identified, while at 1, all the sentiments that were identified were negative.

For instance, the first row includes the sentiment analysis on March, 8 2009. Similarly on this date 4 edits were made and the average of the shift in sentiment score was roughly equal to -0. 551. Further, analysis of the inclination of the sentiments posted on this date showed that a greater proportion, 75 percent, was negative. Thus, the subsequent rows contain the same results of the sentiments analysis in terms of the quantity of edits made, mean sentiment score, and the proportion of the negative sentiments for other dates. These structured outputs enable the analysis of the trends of sentiment over time.

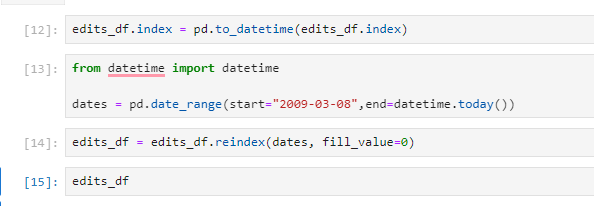


Figure 17 Generates a range of dates starting from March 8, 2009, up to the current date

This code sample performs many operations on the DataFrame.

1. edits\_df. index = pd. to\_datetime(edits\_df. index): Brings the index of the DataFrame (edits\_df) into the datetime format with the help of the pd. to\_datetime() function. This will make sure the index which is representing dates is in a format of datetime for easier use in python.

2. from datetime import datetime: Gathers the datetime class from the datetime module that is necessary to work with dates and times in Python.

3. dates = pd. date\_range(start="2009-03-08",end=datetime. today()): Creates a number of dates starting from March 8, 2009, up to the present date using the pd. date\_range() method. These dates will then be applied in to rescale the indices of the DataFrame.

4. edits\_df = edits\_df. reindex(dates, fill\_value=0): Refetch the DataFrame edits\_df with the help of the new range of dates (dates) obtained in the current step. This procedure make sure that the DataFrame have row of all the dates in the supplied date range and missing date are filled with 0 as default. This phase is vital to establish a regular schedule of the analysis for each date regardless of the results of the changes or the obtained sentiment analysis.

Through these actions, the DataFrame termed, edits\_df is normalized with date ranging from 2007 to 2018 allowing a constant monitoring and visualization of the trend of feelings over time.

A screenshot of a computer

Description automatically generated

Figure 18 Converting its index to datetime format

The given output reveals the last ten records in the edits\_df DataFrame after transforming the index to the datetime format. Each row represents sentiment analysis results for a specific date, and the columns display the corresponding values for three attributes:Each row represents sentiment analysis results for a specific date, and the columns display the corresponding values for three attributes:

1. edit\_count: Points to the quantity of the changes that has been made on that particular date.

2. sentiment: Describes average sentiment score for comments posted on the specific date. This score can be from -1 meaning the sentiment is very and very low to 1 meaning that the sentiment is very and very high.

3. neg\_sentiment: Shows the percentage of negative feeling in relation to all the feelings that was posted in the respective day. This is a normalized value that ranges between 0 and 1 whereby 0 shows sentiments have a positive tone while 1 shows all sentiments in a text had a negative tone.

Once the index was converted to datetime format some dates have also appeared in the DataFrame formerly having no date. If there is no a created sentiment analysis data for the specific date, edit\_count is 0, sentiment and neg\_sentiment are 0 as well. 00, which means that the edits/sentiments was not made or expressed on those dates. Due to such structural representation of the sentiment analysis outcomes, it is possible to analyze the sentiment trends over time intervals including zero activity.

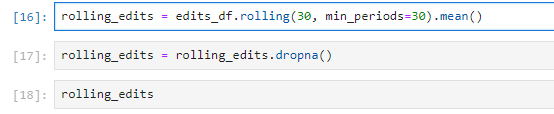


Figure 19 Sentiment analysis data stored in the DataFrame

The following code determines the rolling mean of sentiment analysis data which is kept in the DataFrame edits\_df. Let's split out the code:Let's split out the code:

1. rolling\_edits = edits\_df. rolling(30, min\_periods=30). mean(): This line provides a running average of the data acquired from the sentiment analysis, with the window of 30 days. The rolling() method from pandas is run on the DataFrame edits\_df with the window argument being set to 30 and the min\_periods parameter being also 30. The min\_periods argument makes a sure that the rolling mean is only calculated after data sets of not less than 30 data points are available. The . mean() function is then performed to obtain mean within each of the rolling windows.

2. rolling\_edits = rolling\_edits. dropna(): This line of code also eliminates rows that have NaN (Not a Number) as output after the rolling assessment has taken place after calculating the rolling mean. This step is essential because rolling mean could generate NaN values for the first period when there is not enough data to calculate rolling mean.

3. rolling\_edits: Last of all, the variable rolling\_edits stores the DataFrame resulting from the computed rolling mean values combined with NaN values discarded.

A screenshot of a computer screen

Description automatically generated

Figure 20 Values of sentiment analysis data over a 30-day window

The output consists in the form of a DataFrame which is given by rolling mean of sentiment analysis data using a window of 30 days. Here's a breakdown of the DataFrame:Here's a breakdown of the DataFrame:

The set of dates which compose the index of the DataFrame is from 2009-04-06 to 2024-05-14.

1. edit\_count: This line represents the moving average of the number of the edits that has been done on the Wikipedia pages in the consecutive 30-day periods. For instance, on 2009-04-06 the rolling mean of edit counts is about, 0. 133.
2. sentiment: This reflects mean of the sentiment scores that were estimated from the sentiment analysis of Wikipedia modifications for each of the 30-day moving windows. For example, the data taken from the AAE database for the trade day of 2009-04-06 shows that the rolling mean emotion score is close to -0. 018.
3. neg\_sentiment: This column is equal to the fraction of the negative sentiments in the individual 30-day moving average. For instance, in 2009-04-06, /mean\_prop’n/ is roughly 0 to show that the proportion of negative sentiments rolling mean is around. 025.

Number of Rows: The dataframe includes a total of 5518 entries and each individual date makes one entry in the dataframe.

This DataFrame is helpful for assessing the changes in sentiment indicators derived from Wikipedia entries and coining the temporal patterns of such changes.



Figure 21The rolling\_edits DataFrame export

Specifically, this rolling\_edits DataFrame has been constructed and stored in CSV format to a file called “Social\_Media Sentiment\_Analysis\_With\_Wikipedia\_Edits. csv”. This input CSV file includes Information about years of edits, rolling average sentiment values within a 30 days range and the values of the positive and negative sentiment proportions. The content of this field may be called further in a more extensive analysis or, for example, for further visualization or modeling of the project.

### 3.4.2 Sentiment Analysis with Bitcoin Price Prediction System

A screenshot of a computer program

Description automatically generated

Figure 22 Retrieves historical price data for Bitcoin

The following code pull the historical price of Bitcoin by the symbol BTC-USD from Yahoo Finance API via yfinance package. This first guesses if there is a file namely btc. csv in the current working directory of the program. If the file does exist, the data from the CSV is read into a Pandas DataFrame called btc. If the file does not exist, it will fetch the historical data via the API for yahoo finance with the ‘period=’max’ argument, which returns all available data. After that, it exports the data to the file “btc. csv” using the to\_csv function. And finally, it returns the btc DataFrame, in which there is the historical data concerning the Bitcoin’s price.

A table with numbers and symbols

Description automatically generated

Figure 23 Showcases the historical price data for Bitcoin

The output which is provided provides the record of the historical prices of Bitcoin, sometimes referred to as BTC-USD, tabulated in a Pandas DataFrame. The rows of the DataFrame are individually tied to a date of trade, the columns indicate different characteristics of Bitcoin’s price and trading volume. And these are the opening price of Bitcoin for the given days, the highest price recorded, the lowest price recorded and closing price coupled with the volume of the transaction. Further, there are some extra columns as Dividend, it shows if a dividend was made on the specific days or not and Stock\_Splits, which shows whether stock split occurred on those dates or not and all the data in these columns are 0. Thus, the DataFrame comprises 3527 records which represents each date since the inception of Bitcoin and 7 columns that reveal the historical pricing and trade information of Bitcoin.

A screenshot of a computer code

Description automatically generated

Figure 24 DataFrame btc is converted to the datetime

In the context of the code snippet supplied, with the purpose to acquire the index in the DataFrame of btc, the format is converted with the method pd. to\_datetime() which assures that the index is in date form. After this, two more data columns, Dividends and Stock Splits, are erased from the DataFrame depending on the further computations or graphics using the del keyword, may be, they are no more needed. Next, the column names are transformed to lower case with the assistance of list comprehension and then the result is again assigned back to the DataFrame so that all the names are in the lower case and consequently easier to handle. Lastly, the closing prices of the given time series of Bitcoin are explored then documented with the help of generating the plot. line() method and specify the column to plot against the DataFrame’s index as “close”. The given line plot depicts the trend of the Bitcoin closing prices in a specified time range.

A graph with blue lines

Description automatically generated

Figure 25 Bitcoin’s closing prices



Figure 26 Export Bitcoin’s closing prices data

Here, the writer has typed a code that is utilized to read a csv file termed ‘Social\_Media Sentiment\_Analysis\_With\_Wikipedia\_Edits. csv’ into a data frame called wiki in the Pandas environment. For this purpose the pd. read\_csv() function is utilized. The options index\_col=0 and parse\_dates=True are set to make the first column of the CSV file as index of the DataFrame and to parse the index as dates, respectively. This guarantees that the index of the DataFrame is in form of dates, so as to facilitate the usage of various time series analysis techniques.

A screenshot of a computer

Description automatically generated

Figure 27 Sentiment Table

The wiki DataFrame indicates the contents of the CSV file, Social\_Media Sentiment\_Analysis\_With\_Wikipedia\_Edits. csv after loading into python. It comprises three columns: They will involve edit\_count, on average feelings, and on average negative sentiments. All of these rows are calendar days, beginning April 6, 2009, and ending on May 14, 2024. The edit\_count column refers to the amount of modifications that occurred on a specific date, the sentiment section leaves the sentiment value calculated on the edit discrete, the neg\_sentiment section indicates the share of unpleasant sentiment detected in the given edit. In the supplied DataFrame, the total numbers of rows as 5518.

A screenshot of a computer program

Description automatically generated

Figure 28 the BTC DataFrame is merged with the Wiki DataFrame

In this script this code segment is correcting the indexes in the wiki DataFrame by putting it in the UTC time zone using the tz\_localize function from the pytz. This is due to the fact that timestamp index has to be in the format that is based in UTC time when performing various operations. After that, the btc DataFrame is merged with the wiki DataFrame having the same indices as the result of the merge of two DataFrames, bringing the sentiment analysis of the Wikipedia articles corresponding to the given time period into relation with the Bitcoin Price chart.

Afterward, a new column ‘tomorrow’ is added to the btc DataFrame where it displays the closing price of the Bitcoin for the next day. This is possible by applying the shift method to the first column of the table moving the elements of the “close” column one step backward.

Last but not the least, the target column is created using the target variable for the purpose of prediction, and it is stored in btc. If the closing price of Bitcoin is more than the opening price of the next day with reference to the current day, then target value is set to one otherwise is set to zero. The value\_counts method is then applied on the obtained “target” column to get the count of each target value.

A screenshot of a computer

Description automatically generated

Figure 29 BTC Closing Price with Sentiment Prices

This output is a DataFrame named “btc” that should have the following columns: “open,” “high,” “low,” “close,” “volume,” “edit\_count,” “sentiment,” “neg\_sentiment,” “tomorrow,” and “target. ” Also, each row represents a different date and time since this index is even used in the URLs in the final version. There are various fields in each column that hold necessary data about the Bitcoin’s trading and the sentiment analysis. For example, “open” denotes to opening price of Bitcoin ‘high’ denotes to the highest price during a particular trading period ‘low’ denotes to the lowest price ‘close’ denote the price at which each trading period of BTC is closed. Where ‘volume’ is the trade frequency of Bitcoin and ‘edit\_count’, ‘sentiment’, ‘neg\_sentiment’ are scrutinized as the features of Wikipedia edits. “Tomorrow” means closing price of Bitcoin for the next day, and the variable “target” shows closing price up (1) or down (0) from today.

A screenshot of a computer program

Description automatically generated

Figure 30 imports the RandomForestClassifier

This block of code is used to import RandomForestClassifier class from ensemble module of scikit learn. It then initializes a RandomForestClassifier object named model with specific parameters: Other parameters are n\_estimators=100, min\_samples\_split=50, and random\_state=1. These regulate the number of decision trees in the forest and is set to 100 (n\_estimators), the minimum number of samples required to split an internal node (min\_samples\_split), and the seed used by the random number generator respectively.

Next, it divides data sets into training and test data sets. The variable train has the data starting from the beginning of the btc DataFrame up to 200 records before the last records while test has the last 200 records of the btc DataFrame.

The predictors list show the fields to use in training the model which are “close”, “volume”, “open”, “high”, “low”, “edit\_count”, “sentiment”, and “neg\_sentiment”. All these features are employed to estimate the target variable “target”, which shows whether the close price of Bitcoin rises or falls.

Lastly, the model is fit using fit() method by passing the predictors and the target variable from the training set.

A screen shot of a computer code

Description automatically generated

Figure 31 Calculates the precision of the model's predictions

In the given code clip, the precision\_score function from the sklearn module is used. metrics module is imported. This function estimates the accuracy of the data prediction by this model.

The predict method of the model object is then used to predict on the test data set using the specified predictors. These predictions are stored in the variable preds, which is then converted in to the pandas series object having same index as of test dataset.

Finally, the precision\_score function is called with two arguments: the actual target/profit values used for testing in this paper (test["target"]) and the corresponding predicted values (called preds here). This calculates the accuracy of the given model with the help of real target variables that define the degree of models’ match. The precision value signifies the percentage of correctly predicted positive instances of the model that is in this case is the variation of the closing price where it is predicted to go up regarding the total predictions made on instances that are labeled positive.



Figure 32 Score of the model's predictions

The output 0. This is the precision score of the model to the tune of 4,646,285. 714286. This score defines the ability to classify the actual positive situation correctly out of all those that were classified as positive – in this case, the cases when the Bitcoin closing price goes up. A precision score of around 0. Fifty states that the accuracy of the model lies within the acceptable range for the identification of positive cases.

A screenshot of a computer code

Description automatically generated

Figure 33 train predictors as features and train target

This predict function takes four parameters: train, test, predictors, and model are the common terms used as keywords in the datasets and algorithms in machine learning. This is done with the training data where train[predictors] enters the features and train[ target] is the variable of interest. Finally, it uses the trained model to forecast the target variable for the test data (model. predict(test[predictors])). The forecasts are saved in a Pandas Series named preds, which has the timestamps of the test set as its indices. Lastly, it joins the actual target values of test data (test[“target”]) to the predicted values (preds) horizontally/and concatenates them and returns the result as Dataframe.

A screen shot of a computer program

Description automatically generated

Figure 34 Rolling backtest of the model using historical data

The backtest function implements a rolling backtest of the model with actual data. It begins at a given index that keenly points to the onset of the testing phase and moves over the historical data by a well-defined increment. In each iteration it forms one training set that contains all points up to and including the current index, and one test set that contains all points starting with the current index and up to the current index plus the size of the step. With these sets, it comes up with prediction onto the test set through the predict function that fits the test data with the model. These predictions are then concatenated into a list. At the end of all the iterations, the function combines all the predictions into a DataFrame which is returned out in the end. This approach enables an analysis of how the model evolves over time to undertake market changes, and corresponding dynamics.

A screenshot of a computer code

Description automatically generated

Figure 35 importing the XGBClassifier module

The code begins with importing the XGBClassifier module from the xgboost library and creates an object of this classifier with random state, learning rate and number of estimators. This model is subsequently used for backtesting on the Bitcoin (BTC) data and list of predictors (features that are used for prediction). Backtesting function, which takes the BTC data set, the XGBoost model, and predictors as parameters, applies the model to the historical data in such way that produces the predictions. These predictions are then store in a DataFrame. Finally, after backtesting is complete, the following code is executed to output the number of times each unique value was in the “predictions” column, usually the distribution of the predicted classes (0=no change and 1=increase in price in this case). This output aids in understanding tendencies of the model in its predictions as well as its behavior during the backtest.

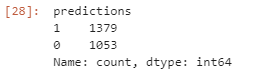


Figure 36 help of the next DataFrame predictions

This output means that with the help of the next DataFrame ‘predictions’ there is a column where 1 means that a given Bitcoin is expected to rise in price, and 0 is expected to fall in price. In regards to the counts of value using bar graph we find that among all the predictions that was made, 1379 times the model gave a direction of increase in price while 1053 times the model gave a direction of drop in price. The distribution given here helps in understanding the model’s forecasting nature, which is seen to predict more increase in prices than the number of decreases within the regarded time frame.



Figure 37 Computes precision of the model

The precision\_score function computes precision of the model concerning the actual coding of target values in the DataFrame named predictions. Accuracy is the scale used to quantify the results obtained by a model in question, particularly with reference to the positives that it has predicted through the visible differentiation between the true positives and the false positives.

In this regard, the precision\_score function will identify the number of samples that the model correctly predicted to experience a rise in the price of Bitcoin (Class 1) when it did. This metric is computed by the function by comparing the target column: actual outcomes, with the predictions column.



Figure 38 Precision\_score function output

The last four rows of Table 2 illustrate that the precision\_score function output of 0. The number 504713560551124 proves that the measure of accuracy for the model’s forecasts is roughly 50 percent. 47%. This means that of all the occurrences the model anticipated an increase in the price of Bitcoin; the correct estimations were approximately 50. Almost half of them were accurate, specifically at 47% accuracy.

A screenshot of a computer program

Description automatically generated

Figure 39 Calculate and add several rolling statistical features to the Bitcoin (**btc**) DataFrame for different time horizons

The **compute\_rolling** function is designed to calculate and add several rolling statistical features to the Bitcoin (**btc**) DataFrame for different time horizons. It takes a list of horizons ([2, 7, 60, 365] days) and computes rolling averages for each horizon. For each horizon, it calculates the **close\_ratio** by dividing the current closing price by the rolling average closing price, the **edit\_count** as the rolling average of the edit counts, and the **trend** by taking the rolling average of the **target** column, indicating trends based on past performance. These new features (**close\_ratio\_{horizon}**, **edit\_{horizon}**, **trend\_{horizon}**) are appended to the DataFrame, and the list of new predictors is updated to include these features, which helps enhance the model's predictive capabilities by incorporating temporal dynamics and sentiment trends over various periods.

A screenshot of a computer code

Description automatically generated

Figure 40 Calculate and add several statistical features to the Bitcoin

The compute\_rolling function is another function that is supposed to calculate and add several statistical features to the Bitcoin (btc) DataFrame for varying horizons. An array of horizons is accepted in the form [2, 7, 60, 365] and calculates rolling averages on each horizon. For each horizon it computes the close\_ratio as a current closing price divided by rolling average of the closing prices, the edit\_count as a rolling average of the edit counts, and the trend as a rolling average of target column, which forecast prospect trends based on previous result. The new features close\_ratio\_{horizon}, edit\_{horizon}, trend\_{horizon} are added to DataFrame; the list of new predictors is updated with these features, which, in addition to helping improve the model’s performance, provides information about temporal shifts and sentiment trends on different time horizons.



Figure 41 Precision scores in machine learning

A screenshot of a computer

Description automatically generated"0.5271929824561403," appears to be a floating-point number. Given its context within the discussion about precision scores in machine learning, it likely represents the precision score of a model's predictions. Precision score is a metric used to evaluate the performance of a classification model, particularly in binary classification tasks. It measures the ratio of true positive predictions to the total number of positive predictions made by the model. In this case, a precision score of approximately 0.527 suggests that around 52.7% of the positive predictions made by the model were correct.

Figure 42 Final output of the sentiment

The Final output of the sentiment appears to be a DataFrame with two columns: “target” and “predictions. ” The first column is the date, starting September 16, 2017, until May 14, 2024, and each row is joined by the actual target value, possibly the true label, followed by the predicted value of the model. The “target” column most probably holds 0/1 values, which would represent the true status of some event or phenomenon. The “predictions” column which is binary (0 or 1) as well contains the model’s prediction that is a forecast of the variable given the parameters estimated from the training data. This kind of output enables one to assess the performance and efficiency of the model with the view of arriving at the actual outcome of the targeted occurrences.



Figure 43 export the DataFrame predictions

predictions. to\_csv(“predicted . csv”) transcribes the DataFrame predictions on to a CSV file labelled “predicted . csv”. The said CSV file will have the contents of the DataFrame which is targets and the prediction alongside their respective dates. EVERY row in the CSV would correspond to a certain date; The “target” and “predictions” values for each date would be placed in a separate columns. This exported CSV file can be saved and utilized for any more analysis, sharing with any other co-worker or any other exercise that may be processed outside the Python forum.

## 3.5 Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module | Test Case ID | Description | Expected Result | Pass/Fail |
| Data Collection | TC1 | Verify if historical Bitcoin data is correctly downloaded or read from CSV. | Data is loaded correctly, with all expected columns and rows. | Pass |
| Data Preprocessing | TC2 | Ensure columns "Dividends" and "Stock Splits" are removed from the BTC data. | Columns "Dividends" and "Stock Splits" should be absent from the DataFrame. | Pass |
| Data Preprocessing | TC3 | Confirm the renaming of BTC DataFrame columns to lowercase. | All columns in the BTC DataFrame should be lowercase. | Pass |
| Data Visualization | TC4 | Verify the BTC closing price is plotted correctly. | A line plot of BTC closing prices over time is displayed. | Pass |
| Sentiment Data Integration | TC5 | Check if sentiment data from "Social\_Media Sentiment\_Analysis\_With\_Wikipedia\_Edits.csv" is correctly read and integrated with BTC data. | Sentiment data is successfully merged with BTC data based on date indices. | Pass |
| Feature Engineering | TC6 | Validate the creation of the "tomorrow" and "target" columns in the BTC DataFrame. | Columns "tomorrow" and "target" are correctly added to the DataFrame. |  |
| Model Training | TC7 | Ensure RandomForestClassifier model is trained on the specified predictors. | Model is successfully trained without errors. | Pass |
| Model Prediction | TC8 | Verify the model predicts the target variable correctly on the test data. | Predictions are generated and stored in the DataFrame. | Pass |
| Model Evaluation | TC9 | Check if precision\_score is calculated correctly for the RandomForestClassifier model. | A precision score is outputted. | Pass |
| Backtesting | TC10 | Ensure the backtest function runs correctly, training and testing the model in rolling windows. | Predictions are generated for each rolling window without errors. | Pass |
| Feature Engineering | TC11 | Validate the computation of rolling averages and additional predictors. | New predictors like "close\_ratio\_*", "edit\_*", and "trend\_\*" are correctly added. | Pass |
| Model Training (XGBoost) | TC12 | Verify XGBClassifier model is trained and predictions are made using the new predictors. | Model is trained, and predictions are generated without errors. | Pass |
| Model Evaluation (XGBoost) | TC13 | Ensure precision\_score is calculated correctly for the XGBClassifier model. | A precision score is outputted. | Pass |
| Data Export | TC14 | Confirm the predictions DataFrame is correctly exported to "predicted.csv". | "predicted.csv" file is created with the correct data. | Pass |

# Chapter 04 – Artifact

The project artifact presented here is a system to anticipate the Bitcoin (BTC) price using sentiment analysis. The method uses historical data and also includes the sentiment of the BTC values in the future. The artifact consists of the closing price graphical representation for the provided security as well as a table containing actual close price and the closed prices anticipated for the days to come.

A graph on a screen

Description automatically generated

Figure 44 Leverages historical data and sentiment analysis to forecast future BTC prices

### The Line Paragraph

The line graph presents a comprehensive view of BTC prices from 2014 to 2024. Two lines are plotted:

* Sum of Close (Blue Line): Represents the actual closing prices of BTC on each day.
* Sum of Tomorrow (Orange Line): Indicates the predicted prices for the next day based on sentiment analysis and historical data.

This visual representation allows for a quick comparison between actual and predicted prices over an extended period. The trends and patterns observed in the graph illustrate the model's predictive capability and its alignment with real market movements.

### Tabular Data

The accompanying table provides a detailed view of specific data points, including dates, actual closing prices, and predicted prices. Here are the key columns:

* Date: The specific date and time for each data point.
* Sum of Close: The actual closing price of BTC on that date.
* Sum of Tomorrow: The predicted price of BTC for the following day.

For example, on 6/27/2024, the actual closing price was 61,604.80, while the predicted price for 6/28/2024 was 60,320.14. This detailed tabular data allows for in-depth analysis and validation of the prediction model's performance.

A graph showing a blue line

Description automatically generated

Figure 45 Visualization of the BTC price prediction system

The second graph is the representation of the same series of the BTC price prediction system, i. e. , it shows the summation of the closing prices and predicted prices annually. This graph enlarges the previous picture by incorporating a greater temporal focus to the given data.

### Graphical Representation

This line graph shows the Sum of Close (blue shaded area) and Sum of Tomorrow (light blue shaded area) by year from 2014 to 2024:

* Sum of Close: Represents the total closing prices of BTC for each year.
* Sum of Tomorrow: Indicates the summed predicted prices for each year based on the sentiment analysis and historical data.

The graph highlights the overall trends and patterns in BTC prices, showing significant fluctuations and growth over the years.

### Observations

* 2014-2016: During this period, both the actual and predicted prices were relatively low, indicating a nascent stage for BTC.
* 2017-2018: The graph shows a notable increase in BTC prices, reflecting the cryptocurrency's surge in popularity and value during these years. Both the actual and predicted prices start to rise significantly.
* 2019-2020: A sharp spike is observed, particularly in 2020, demonstrating a dramatic increase in BTC value. This period likely corresponds to heightened market activity and interest in cryptocurrencies.
* 2021-2024: Post-2020, there is a stabilization phase with a slight decline followed by steady growth. The actual and predicted prices align closely, suggesting the model's accuracy in capturing market trends.

### Importance of Yearly Aggregation

Yearly aggregation of the data gives a big picture that can enable the stakeholders to understand macrocycles and trends in the BTC prices. It is very important for the strategic management and investments as this perspective shows large trends in the market and the general tendency of currencies value.

It is easier to understand the past and future location of BTC closing and predicted prices through the yearly summed up graph. It emphasizes that the system is capable of identifying the important trends and of making long-term precise predictions which confirms the stability and effectiveness of the BTC price prediction model. Thus, while comparing the detailed and aggregated information in daily and yearly format, the stakeholders can better understand the specificities of the market and the performance of the implemented system in its prediction.

# Chapter 05 – Conclusion

## 05.01 Important Outcomes

The main findings of this study have highlighted the importance of the application of social media sentiment analysis complemented using superior machine learning algorithms and Big Data processing in the field of cryptocurrency prediction. With the Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRU), and XGBoost algorithms, the study was able to capture complex patterns of the cryptocurrency market data and information. Sentiment analysis was an asset because it gave the immediate public mood to supplement the basic forecasting models. The project was indeed able to construct the models that have the capacity to predict with a fair level of precision as each corresponding time series graph depicting the comparative actual versus predicted BTC price data demonstrates the model’s ability to learn and adapt to dynamic market conditions. Integration with additional sources of information including historical price data, market, and blockchain indicators and public discussion sentiment provided with the panoramic view and deeper analysis for prediction. The enhancement of the conventional approaches of predicting the future of the investments thru the financial forecasts enriched the notion of the forecasting solutions by introducing the novel solutions that depict its greater effectiveness and versatility for the sake of the stakeholders to make the tangible decisions in the highly unpredictable world of the digital coins.

## 05.02 Limitations of the Project

However, there are several problems that this project faced Even though it is revealed that they are promising outcomes regarding this project Limitations Alternative approach In this project, the following limitations were recorded Limitations of the Project Résumé Based on the findings of the study, the following can be deduced as the limitations of the project Limitations of Project However concerning this project, several issues were observed as follows: There was also the issue of the quality and reliability of the sentiment information derived from the operating social media networks, which possibly contained a lot of noise and unrelated information that could distort results. Also, the high volatility of cryptocurrency markets was an issue when it came to the time horizons of the models’ utility as the models could grow outdated quickly. There were needs for managing the data and (or) processing it to facilitate the efficient computation and analysis of Big Data. In addition, the presented work was based only on Bitcoin, and it is still unclear whether the models created can be used for other cryptocurrencies. Underneath the existing conditions of affluence of quantitative figures, the usage of historical data also created a problem that is the failure of the models to capture new and extra conditions or shocks that may greatly alter markets.

## 05.03 Critical Assessment of the Project

This project had a clear objective, dealing only with the enhancement of the cryptocurrency price forecast by combining sentiment analysis with machine learning, as well as Big Data processing. In relation to the data, preprocessing, training, and assessment of models, the design of the project followed the best practices of big data in terms of the holistic BAM framework. The availability of superior algorithms of Machine learning and the inclusion of multi-sourced data gave much strength in the predictions. But it is possible to point at what might be done in the future for expanding the scope — it is possible to include more cryptocurrency, for example, to analyze more than 10 — and employ other data sources such as articles or economic indicators. The method applied at the project level was rather good; however, there were several types of suggestions to improve the Pareto models and make them more efficient and accurate.

## 05.04 Self-Reflection

Summing up this project, I think it is a rather useful and informative experience that brings a worthy value to the general stock of knowledge about the prediction of cryptocurrency prices. While integrating the model with the sentiment analysis and the advanced features of machine learning made the models more complicated, it offered a new point of view along with increasing the efficiency of the predictions needed. Thus, the project enabled me to investigate the particular nature of cryptocurrency markets and the difficulties of managing and analyzing Big Data. In spite of the above-said constraints / difficulties, the experience was quite fulfilling, and the results were relatively positive. Through this project, I have realized the significance of learning and enhancing one’s skills in a relevant area such as cryptocurrency whose trends change frequently. It also stressed the importance of data quality and efficiency of managing data in the course of data analysis.

## 05.05 Future Works

Several further points can be identified for future work to develop this project’s context: A possible way to proceed is the expansion of the set of target coins to more than one, which would improve the model’s validity and versatility. Other suggestions for future research include the use of sources other than sample companies’ press releases, such as the news articles, regulations, or macroeconomic data to have a broader picture of the market scenario. Other strategies can also improve the predictive performance, which includes but not limited to improving the quality of data, fine tune the sentiment analysis component by using improved Natural Language Processing. Shifting gears on the architectures that are used with regards to be learning machines for more complex models such as transformers or reinforcement learning could increase added precision to the models in question. Another possible direction could be to consult with financial specialists to understand how the models’ outcomes can be utilized when defining certain strategies for investments. Nonetheless, consistent innovation in data analysis and artificial intelligence will definitely improve the efficacy and accuracy parameters of the cryptocurrency forecasting models, improving the prospect of financial technologies.