**A Project Report on**

**Bitcoin Price Prediction Using Machine Learning**

submitted in partial fulfillment for the award of

**Bachelor of Technology**

in

**Computer Science & Engineering**

by

**U Tarun (Y20ACS580) P Mounika (Y20ACS532)**

**T Eswar prasad (Y20ACS574) R L Pujitha (Y20ACS542)**



Under the guidance of

**Mrs. M. Karuna, M. Tech.**

Department of Computer Science and Engineering

**Bapatla Engineering College**

(Autonomous)

(Affiliated to Acharya Nagarjuna University)

**BAPATLA – 522 102, Andhra Pradesh, INDIA**

**2023-2024**

**Department of**

**Computer Science & Engineering**



**CERTIFICATE**

This is to certify that the project report entitled **Bitcoin Price Prediction Using Machine Learning** that is being submitted by U. Tarun (Y20ACS580), P Mounika (Y20ACS532), T Eswar prasad (Y20ACS572), R.L. Pujitha in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

**Signature of the Guide Signature of the HOD**

**Mrs. M Karuna Dr. M. Rajesh Babu**

**Designation Prof. & Head**

**DECLARATION**

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

**U Tarun(Y20ACS580)**

**P Mounika (Y20ACS532)**

**T Eswar Prasad(Y20ACS574)**

**R L Pujitha(Y20ACS542)**

**Acknowledgement**

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **Mrs. M Karuna**, Asst.Prof., Department of CSE, for his/her valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr. M. Rajesh Babu**, Assoc. Prof. & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal **Dr. Nazeer Shaik** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator **Dr. N. Sudhakar,** Prof. Dept. of CSE for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

**U Tarun(Y20ACS580)**

**P Mounika (Y20ACS532)**

**T Eswar Prasad(Y20ACS574)**

**R L Pujitha(Y20ACS542)**

**Abstract**

In today's digital marketing landscape, Bitcoin reigns as the premier cryptocurrency, facilitating anonymous transactions over the internet. However, its virtual nature and extreme volatility pose significant challenges for accurate forecasting. This study aims to develop a precise predictive model for Bitcoin values, utilizing various machine learning techniques. Incorporating technical indicators like Exponential Moving Average (EMA), Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD), we analyze data sourced from Yahoo Finance to understand the factors influencing Bitcoin's value. By scrutinizing these parameters, we seek to forecast daily price changes with accuracy. Our research endeavors to provide valuable insights for investors navigating the complexities of the cryptocurrency market, ultimately enhancing understanding and prediction of Bitcoin price movements.

**Keywords**: Bitcoin, cryptocurrency, digital marketing, virtual money, anonymous transactions, internet, volatility, forecast, machine learning models, parameters, Yahoo Finance, stock market, technical indicators, EMA (Exponential Moving Average), RSI (Relative Strength Index), MACD (Moving Average Convergence Divergence).

**Table of Contents**

[**Abstract** v](#_Toc164199771)

[**Table of Contents** vii](#_Toc164199772)

[**List of Tables** ix](#_Toc164199773)

[**List of Figures** x](#_Toc164199774)

[**List of Abbreviations** xi](#_Toc164199775)

[1 Introduction 1](#_Toc164199776)

[1.1 Machine Learning 2](#_Toc164199777)

[1.1.1 Supervised Learning 2](#_Toc164199778)

[1.1.2 Unsupervised Learning 3](#_Toc164199779)

[1.2 Introduction to Cryptocurrency 4](#_Toc164199780)

[1.2.1 Cryptocurrency 4](#_Toc164199781)

[1.2.2 Bitcoin 5](#_Toc164199782)

[1.3 Objective 6](#_Toc164199783)

[2 Literature Survey 7](#_Toc164199784)

[3 Problem statement 9](#_Toc164199785)

[4 System Analysis 10](#_Toc164199786)

[4.1 Existing System 10](#_Toc164199787)

[4.1.1 Limitations 10](#_Toc164199788)

[4.2 Proposed System 10](#_Toc164199789)

[4.3 Architecture 11](#_Toc164199790)

[5 Methodology 13](#_Toc164199791)

[5.1 Dataset 13](#_Toc164199792)

[5.2 Preprocessing 14](#_Toc164199793)

[5.2.1 Data Cleaning 15](#_Toc164199794)

[5.2.2 Feature Selection 15](#_Toc164199795)

[5.2.3 Feature Engineering 15](#_Toc164199796)

[5.2.4 Normalization 16](#_Toc164199797)

[5.2.5 Data Splitting 16](#_Toc164199798)

[5.3 Algorithms 17](#_Toc164199799)

[5.3.1 Logistic Regression 17](#_Toc164199800)

[5.3.2 Random Forest 19](#_Toc164199801)

[5.3.3 XGBoost 20](#_Toc164199802)

[5.4 Evaluation of Algorithms 21](#_Toc164199803)

[5.4.1 Classification Model Evaluation 21](#_Toc164199804)

[5.4.2 Regression Models Evaluation 22](#_Toc164199805)

[6 System Requirements and Specifications 24](#_Toc164199806)

[6.1 Hardware Requirements 24](#_Toc164199807)

[6.2 Software Requirements 24](#_Toc164199808)

[6.2.1 Software Libraries 25](#_Toc164199809)

[7 System Design 26](#_Toc164199810)

[7.1 UML Diagrams 26](#_Toc164199811)

[7.1.1 Use case diagram 26](#_Toc164199812)

[7.1.2 Class diagram 26](#_Toc164199813)

[7.1.3 Object diagram 26](#_Toc164199814)

[7.1.4 Sequence diagram 26](#_Toc164199815)

[8 Testing 27](#_Toc164199816)

[8.1 Unit Testing 27](#_Toc164199817)

[8.2 Integration Testing 27](#_Toc164199818)

[8.3 System Testing 27](#_Toc164199819)

[8.4 Acceptance Testing 27](#_Toc164199820)

[8.5 Performance Testing 28](#_Toc164199821)

[9 Implementation 29](#_Toc164199822)

[9.1 Loading the Dataset 29](#_Toc164199823)

[9.2 Dropping Unnecessary Columns 29](#_Toc164199824)

[9.3 Converting Date to Datetime 29](#_Toc164199825)

[9.4 Extracting Date Features 29](#_Toc164199826)

[9.5 Feature Engineering 29](#_Toc164199827)

[9.5.1 Technical Indicators 29](#_Toc164199828)

[9.6 Splitting Data and Model Training 31](#_Toc164199829)

[9.7 Predicting future prices 32](#_Toc164199830)

[9.8 Evaluating Model Performance 32](#_Toc164199831)

[9.9 Generating Trading Signals (Signal Generation) 32](#_Toc164199832)

[9.10 Implementation GitHub link 32](#_Toc164199833)

[10 Result 33](#_Toc164199834)

[11 Conclusion 34](#_Toc164199835)

[12 Future Enhancement 35](#_Toc164199836)

[13 References 37](#_Toc164199837)

**List of Tables**

No table of figures entries found.

**List of Figures**

[[Figure 4.1 Combined Architecture of both Classification and Regression Technique 12](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx" \l "_Toc164199838)](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx#_Toc164243394)

[[Figure 5.1 Dataset 14](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx" \l "_Toc164199838)](#_Toc164243395)

[[Figure 5.2 Dataset 14](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx" \l "_Toc164199838)](#_Toc164243396)

[[Figure 5.3 Sigmoid function 18](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx" \l "_Toc164199838)](#_Toc164243397)

[[Figure 5.4 Random Forest 19](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx" \l "_Toc164199838)](#_Toc164243398)

[[Figure 5.5 XGBoost 20](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx" \l "_Toc164199838)](#_Toc164243399)

[[Figure 7.1 Use case Diagram 26](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx" \l "_Toc164199838)](#_Toc164243400)

[[Figure 7.2 Class Diagram 27](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx" \l "_Toc164199838)](#_Toc164243401)

[[Figure 7.3 Activity Diagram 28](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx" \l "_Toc164199838)](#_Toc164243402)

[[Figure 7.4 Sequence Diagram 29](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx" \l "_Toc164199838)](#_Toc164243403)

[[Figure 10.1 Classification report 36](https://d.docs.live.net/5b6eef12da3f2906/Desktop/bitcoin%20final%20report.docx" \l "_Toc164199838)](#_Toc164243404)

**List of Equations**

[Eq. 1 Accuracy 21](#_Toc164200343)

[Eq. 2 Precision 21](#_Toc164200344)

[Eq. 3 Recall 22](#_Toc164200345)

[Eq. 4 F1 Score 22](#_Toc164200346)

[Eq. 5 RMSE 22](#_Toc164200347)

[Eq. 6 MAE 23](#_Toc164200348)

[Eq. 7 R-Squared 23](#_Toc164200349)

# Introduction

The cryptocurrency market, particularly Bitcoin, has emerged as a dynamic and influential sector in the global financial landscape. With its unprecedented volatility and intricate price movements, Bitcoin has captivated the attention of investors, traders, and researchers alike. Understanding and predicting Bitcoin price fluctuations has become a focal point for many, driven by the potential for lucrative investment opportunities and the desire to mitigate risks in this nascent yet rapidly evolving market.

This study aims to explore and develop a comprehensive framework for predicting Bitcoin prices using a combination of machine learning algorithms and a diverse array of technical indicators. By leveraging historical price data and extracting features from indicators such as moving averages, relative strength index (RSI), and stochastic oscillator, among others, we seek to uncover meaningful patterns and relationships that can inform future price movements.

The utilization of machine learning models, including but not limited to random forests and XGBoost Classifier, allows for the creation of predictive models capable of capturing complex market dynamics and adapting to changing conditions over time. Through rigorous evaluation using established metrics such as mean absolute error (MAE) and root mean square error (RMSE) on test data, we aim to assess the performance and effectiveness of our proposed approach.

Ultimately, this research endeavors to contribute to the growing body of knowledge surrounding cryptocurrency market analysis and provide valuable insights for investors, traders, and stakeholders navigating the dynamic landscape of Bitcoin and other digital assets. By harnessing the power of machine learning and technical indicators, we strive to enhance the accuracy and reliability of Bitcoin price prediction, empowering market participants to make informed decisions and navigate the complexities of the cryptocurrency market with greater confidence.

## Machine Learning

Machine learning (ML) is a branch of artificial intelligence (AI) that enables computers to "self-learn" from training data and improve over time, without being explicitly programmed. Machine learning algorithms can detect patterns in data and learn from them, to make their own predictions. In short, machine learning algorithms and models learn through experience.

In traditional programming, a computer engineer writes a series of directions that instruct a computer how to transform input data into a desired output. Instructions are mostly based on an if-then structure, when certain conditions are met, the program executes a specific action. Machine learning, on the other hand, is an automated process that enables machines to solve problems with little or no human input, and take actions based on past observations.

While artificial intelligence and machine learning are often used interchangeably, they are two different concepts. Al is the broader concept machines making decisions, learning new skills, and solving problems in a similar way to humans whereas machine learning is a subset of Al that enables intelligent systems to autonomously learn new things from data.

### Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system to train it, and on that basis, it predicts the output. The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

This is the most common and popular approach to machine learning. It's "supervised" because these models need to be fed manually tagged sample data to learn from. Data is labeled to tell the machine what patterns (similar words and images, data categories, etc.) it should be looking for and recognize connections with.

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested based on test data (a subset of the training set), and then it predicts the output.

### Unsupervised Learning

Unsupervised learning algorithms uncover insights and relationships in unlabeled data. In this case, models are fed input data, but the desired outcomes are unknown, so they have to make inferences based on circumstantial evidence, without any guidance or training. The models are not trained with the "right answer," so they must find patterns on their own.

One of the most common types of unsupervised learning is clustering, which consists of grouping similar data. This method is mostly used for exploratory analysis and can help you detect hidden patterns or trends. Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

## Introduction to Cryptocurrency

Artificial Intelligence (AI) is now at the heart of innovation economy and thus is the base for our work. The main problem in the development of image description started with object detection using static object class libraries in the image and modelled using statistical language models.

* + 1. Making use of CNN: It’s a Deep Learning algorithm that will intake in a 2D matrix input image, assign importance (learnable weights and biases) to different aspects/objects in the image, and be intelligent enough to be able to differentiate one from the other.
    2. This model was advantageous in naming the objects in an image but it could not tell us the relationship among them (that’s plain image classification).
    3. Present a generative model built on a deep recurrent architecture that unites recent advances in computer vision and machine translation and that can effectively generate meaningful sentences.
    4. Making use of an RNN: They are networks with loops in them, allowing information to persist. LSTMs are a particular kind of RNN, capable of learning long-term dependencies.
    5. Further an attention layer is used along with the RNN model to generate more relevant captions for any given image.

### Cryptocurrency

**Cryptocurrency** is a digital form of money designed to act as a medium of exchange. Unlike traditional currencies controlled by governments, cryptocurrency operates on a decentralized system called a blockchain. This blockchain is a public ledger of transactions that is secure and tamper-proof due to cryptography.

### Bitcoin

**Bitcoin** is the first and most well-known cryptocurrency, launched in 2009 by the mysterious Satoshi Nakamoto. Bitcoin transactions are verified and secured by a network of computers running specialized software. This process is called **mining**.

#### ****Bitcoin Generation (Mining)****

* Miners compete to solve complex mathematical puzzles that verify transactions on the Bitcoin network.
* The first miner to solve the puzzle is rewarded with a set amount of Bitcoin
* The difficulty of these puzzles increases as more Bitcoins are mined, controlling the rate at which new Bitcoins are created.

#### ****Storing Bitcoin****

* Bitcoin is not stored in a physical location but rather in a digital wallet.
* These wallets come in various forms, including software wallets on your computer or phone, hardware wallets that resemble USB drives, and even online wallets offered by cryptocurrency exchanges.

#### ****Bitcoin Price Volatility****

* The price of Bitcoin is highly volatile, meaning it can fluctuate significantly in a short period.
* Several factors influence Bitcoin's price, including:
  + Supply and demand: As Bitcoin has a limited supply (21 million), its value is influenced by market demand.
  + News and events: Positive news about Bitcoin adoption can drive prices up, while negative events can lead to sell-offs.
  + Regulations: Government regulations on cryptocurrency can impact market sentiment and price.

## Objective

This project aims to develop a robust predictive model for Bitcoin prices, leveraging the power of machine learning techniques. By analyzing historical data obtained from reputable sources like Yahoo Finance, the model will incorporate technical indicators such as EMA (Exponential Moving Average), RSI (Relative Strength Index), and MACD (Moving Average Convergence Divergence) to identify patterns and trends influencing Bitcoin's value.

# Literature Survey

The cryptocurrency market is transforming the world of money and finance [1], and has seen significant growth in the last years [1], [2]. In particular, the number of cryptocurrencies reached more than 7000 in 2021[3], and the crypto market capitalization hit $3 trillion the same year [3]. The banking and financial industry has taken notice of Blockchain benefits. The underlying technology behind every cryptocurrency is Blockchain technology. Blockchain is a distributed/decentralized database that is organized as a list of blocks, where the committed blocks are immutable. It has many attractive properties including transparency and security [2].

In 2018, Saad et al. [4] provided a machine learning model to predict Bitcoin price. In particular, they made use of a regression model and involved many factors that impact the price of Bitcoin. However, they did not provide Buy and Sell signals, which are the most important in building a trading strategy/approach. Furthermore, they did not consider any kind of technical indicators. The use of technical indicators as features to feed machine learning models for financial trading has been successfully employed by many researchers [5], [6].

McNally et al. [7] proposed a machine learning model that makes a recurrent neural network, called Long Short Term Memory Model (LSTM). LSTM achieves an accuracy of 52%, for classification. However, this is not acceptable for building a trading strategy. Our approach achieves 86%, which is quite acceptable.

Recently, Jay et all. [8] proposed a stochastic neural network model for cryptocurrency price prediction. Precisely, they made use of random walk theory, and Multi-Layer Perceptron (MLP) and LSTM machine learning models [8]. The approach achieves good mean absolute percentage error (MAPE). However, they did not consider Buy and Sell signals. They also did not consider any kind of technical indicators to feed their machine learning models.

In a more recent work, Singh et al. [9] proposed three machine learning models to predict the price of cryptocurrency. They reported that Gated Recurrent Unit (GRU) provides a good accuracy compared to others with a MAPE of 0.2454% for Bitcoin. However, the authors did not provide Buy and Sell signals, and did not consider technical indicators to feed this machine learning model.

# Problem statement

Due to the extreme volatility of Bitcoin prices, traditional investment analysis methods often struggle to keep pace with daily fluctuations. This project tackles this challenge by developing a day-wise Bitcoin price prediction model using machine learning.

The model will incorporate a comprehensive dataset encompassing historical price data, trading volume data, sentiment analysis from social media and news articles, and technical indicators like RSI, MACD, and Bollinger Bands. By analyzing these factors, the model aims to identify patterns and trends that influence Bitcoin's value. We acknowledge the inherent volatility of the cryptocurrency market and the limitations of any model in perfectly predicting future prices. Our primary evaluation metric will be Mean Squared Error (MSE).

Despite these limitations, this project aims to create a valuable tool for investors navigating the complexities of the Bitcoin market. The model can potentially provide insights into price trends, market sentiment, and potential support and resistance levels, ultimately empowering investors to make more informed decisions.

# System Analysis

Bitcoin price prediction is a complex and ever-evolving field due to the inherent volatility of the cryptocurrency market. While there's no guaranteed way to predict future prices perfectly, several approaches attempt to forecast potential movements.

## Existing System

Existing systems for Bitcoin price prediction encompass various approaches, with a focus on technical analysis and machine learning. Technical analysis utilizes historical price data and indicators like RSI and MACD to identify potential trends. Machine learning algorithms are trained on historical data to learn complex relationships and predict future prices. However, these methods all face challenges due to the inherent volatility of the cryptocurrency market and the limitations of model accuracy.

### Limitations

Existing Bitcoin price prediction systems, whether based on technical analysis with indicators like RSI or machine learning algorithms, all struggle with the inherent volatility of the cryptocurrency market, making it difficult for models to translate historical data into perfectly accurate future predictions. Additionally, even the most advanced models have limitations and cannot guarantee perfect accuracy, and external events like news or regulations can throw predictions off course.

## Proposed System

Accurately predicting Bitcoin prices, notoriously volatile like the weather, remains a significant challenge. This project tackles this complexity with a novel machine learning approach. We build a system that analyzes a comprehensive dataset encompassing historical Bitcoin price data, trading volume, and even sentiment analysis gleaned from online sources. By analyzing these factors, the model aims to identify patterns and trends that can potentially influence future Bitcoin prices.

This project takes a unique two-pronged approach. First, the model attempts to predict the direction of future price movements, classifying them as up or down. This classification leverages the theory that historical trends and technical indicators can offer clues about future market direction, similar to how past weather patterns might suggest future conditions. However, the system doesn't stop there. It also attempts to predict the specific future price point of Bitcoin. This aspect is based on the theory that historical data and other factors can influence future prices to a certain degree.

It's important to acknowledge that the cryptocurrency market is inherently volatile, so these predictions won't be perfect. However, by combining direction signals and price predictions, this project strives to create a valuable tool for investors navigating the complexities of the Bitcoin market. The model can potentially empower investors with a more comprehensive picture, allowing them to make informed decisions based on both directional trends and potential price points.

## Architecture

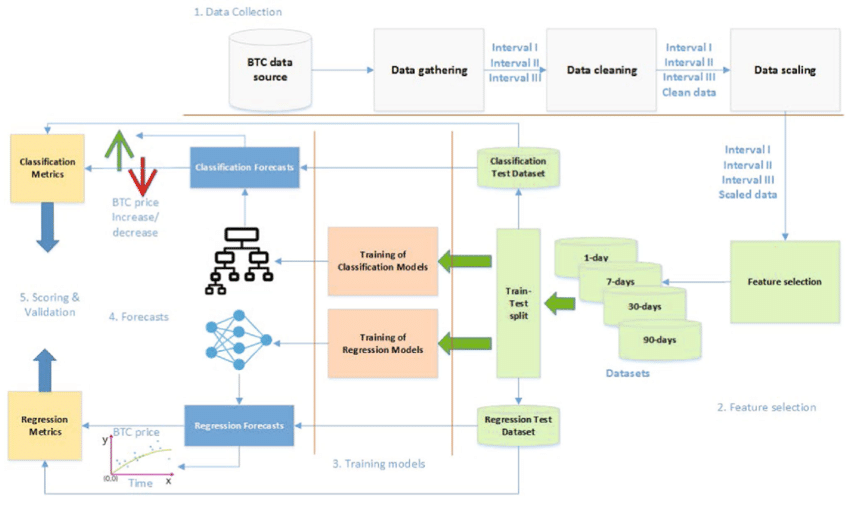
Bitcoin price prediction system powered by a two-fold machine learning approach. First, a classification model analyzes historical close price data (retrieved externally) and technical indicators (RSI, MACD) to predict the future direction (up or down) of Bitcoin prices. Additionally, the system incorporates a regression model (if applicable) to estimate the specific future price point of Bitcoin. While acknowledging limitations due to market volatility and model accuracy, this system aims to equip investors with a more comprehensive view of the Bitcoin market. By potentially providing both directional signals and price predictions.****

Figure 4.1 Combined Architecture of both Classification and Regression Technique

# Methodology

Methodologychapter describes the algorithms and architectures that were used in the proposed model. This chapter mainly focus on the methodology that required to develop a model to predict the price of bitcoin using machine learning techniques.

A. First we will search and find the data from different online sources like Finance.yahoo.com, coin market cap or from Kaggle for historical data.

B. After downloading we will pre-process the data that will help the machine learning algorithms. We will train the dataset and generate models separately to check the accuracy

C. We will fetch the real time data and then we apply the generated model to these fetched data and check if the sufficient data is to predict the price of the bitcoin accurately.

Methodologychapter describes the algorithms and architectures that were used in the proposed model. This chapter mainly focus on the methodology that required to develop a model to predict the price of bitcoin using machine learning techniques.

A. First we will search and find the data from different online sources like Finance.yahoo.com, coin market cap or from Kaggle for historical data.

B. After downloading we will pre-process the data that will help the machine learning algorithms. We will train the dataset and generate models separately to check the accuracy

C. We will fetch the real time data and then we apply the generated model to these fetched data and check if the sufficient data is to predict the price of the bitcoin accurately.

## Dataset

A dataset is a collection of data organized for analysis or processing. This data can be in a variety of forms, including numbers, text, images, or audio. Datasets are typically stored in a structured format, such as a table in a database or a CSV file. This makes it easy to access and manipulate the data for analysis.

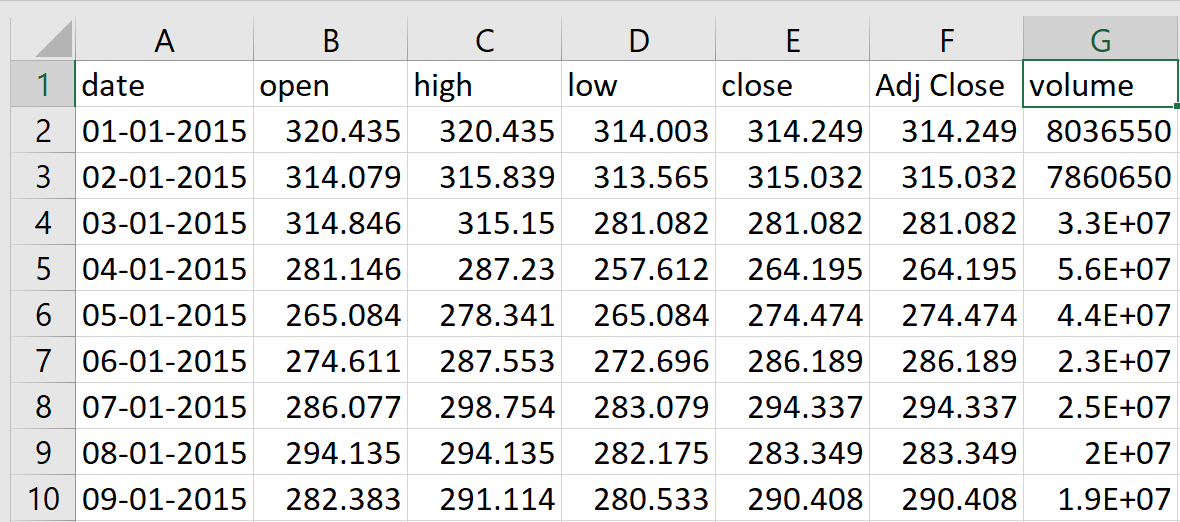


Figure 5.1 Dataset

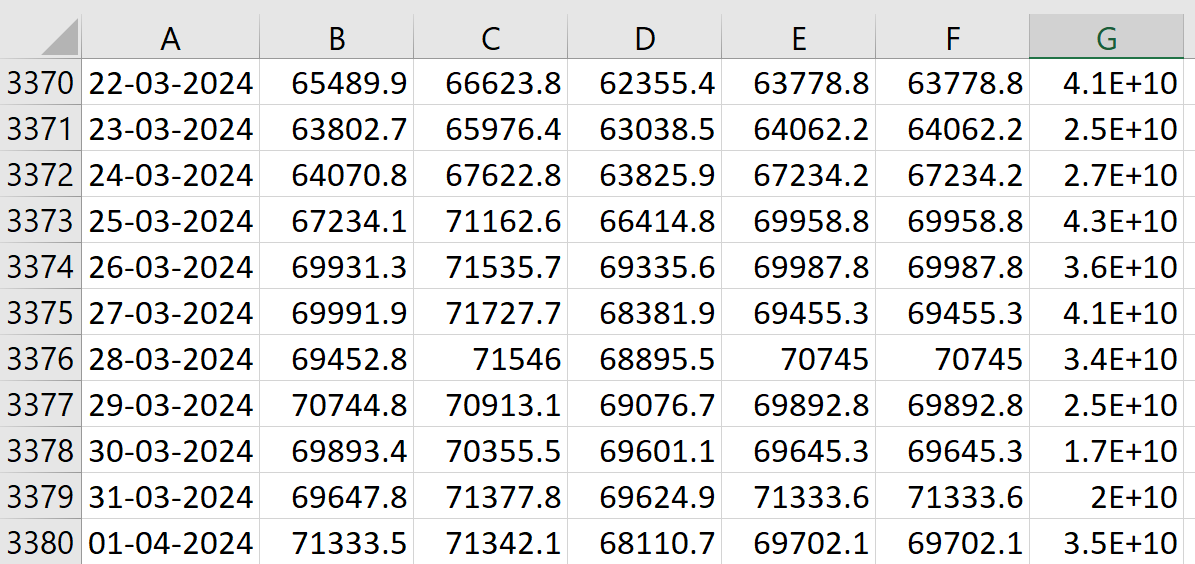


Figure 5.2 Dataset

## Preprocessing

Data preprocessing is the most important phase in prediction models as the data consists of ambiguities, errors, redundancy which needs to be cleaned beforehand. The data gathered from multiple sources first is aggregated and then cleaned as the complete data collected is not suitable for modeling purposes. The records with unique values do not have any significance as they do not contribute much in predictive modeling. Fields with too many null values also need to be discarded.

### Data Cleaning

The data cleaning process focuses on ensuring the quality and integrity of the dataset specifically tailored for predicting Bitcoin prices. This step involves addressing missing data, handling duplicate records, and rectifying errors present in the dataset.

### Feature Selection

Feature selection involves identifying and selecting the most relevant features from the dataset to improve model performance, reduce overfitting, and enhance interpretability.

We have majorly seven features are there. They are Date, Open, Close, Low, High, Adj Close, Volume. From these features we selecting the most important features like Date, Closing price along with the Volume.

### Feature Engineering

Feature engineering is essential for predicting cryptocurrency market movements using machine learning techniques. By extracting insights from historical market data and technical indicators, feature engineering creates informative attributes that capture the market's intrinsic dynamics. Key features such as close price and volume provide foundational understanding of market sentiment and trading activity.

Additionally, technical indicators like exponential moving average (EMA), moving average convergence divergence (MACD), and relative strength index (RSI) offer deeper insights into trends, momentum, and volatility, enriching the dataset with actionable information for modeling.

Furthermore, technical indicators such as momentum, price rate of change (PROC), stochastic oscillator, bollinger bands, and moving averages (MA) along with the enhance the dataset by quantifying aspects like velocity, rate of change, and market conditions of overbought or oversold. These features, alongside those derived from moving averages, are pivotal in generating buy and sell signals, facilitating trend identification and market timing. Through meticulous feature engineering, the model becomes adept at capturing the intricate patterns inherent in the cryptocurrency market, thereby bolstering its predictive prowess and reliability.

### Normalization

Normalization refers to scaling numerical features to a range between 0 and 1, while standardization involves scaling features to have a mean of 0 and a standard deviation of 1. These techniques help in making different features comparable and prevent features with larger scales from dominating the analysis.

### Data Splitting

Finally, the preprocessed dataset is often split into training, validation, and testing sets. The training set is used to train the model, the validation set is used to fine-tune hyperparameters and evaluate model performance during training, and the testing set is used to evaluate the final model performance.

## Algorithms

The proposed method's final stage, called classification, involves feeding extracted features into a classifier for training and testing algorithm before it is used in the prediction phase. The proposed system makes use of machine learning algorithms. Logistic Regression, Random Forest and XGBClassifier are the algorithms we employ in our proposed system. These algorithms are supervised Machine Learning algorithms which are used for classification problems. For Regression we are using Random Forest Regressor to predict the future Price . Using these algorithms, the models are developed and evaluated. Using the same dataset to these machine learning models were assessed. The data after being pre-processed and extraction of features is passed into these models.

### Logistic Regression

Logistic Regression is a widely used statistical technique for binary classification problems, where the outcome variable has two possible outcomes. Despite its name, logistic regression is a classification algorithm rather than a regression algorithm. It works by estimating the probability that a given input belongs to a particular class.

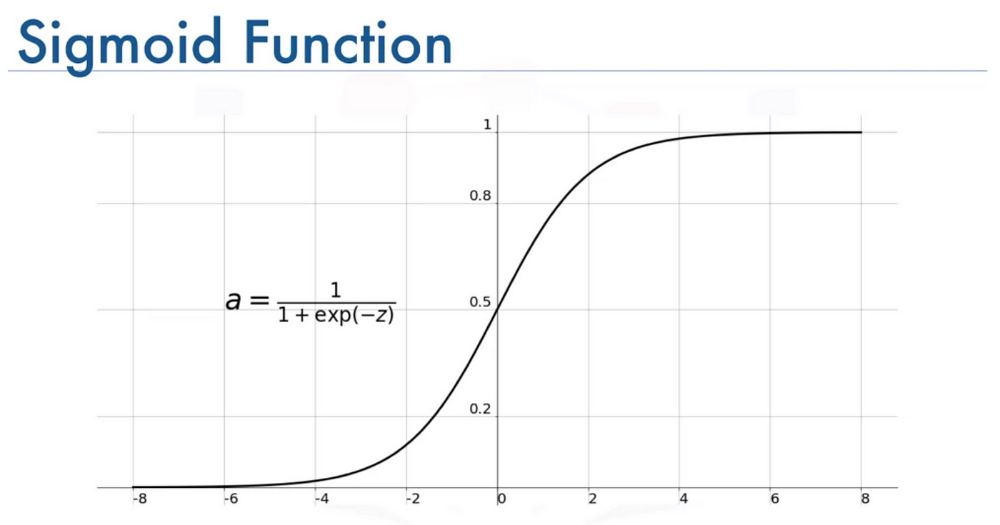


Figure 5.3 Sigmoid function

One of the key features of logistic regression is its use of the sigmoid function (also known as the logistic function) to model the relationship between the input features and the probability of the outcome. The sigmoid function is an S-shaped curve that maps any real-valued number to the range [0, 1]. This allows logistic regression to output probabilities that are bounded between 0 and 1, making it suitable for classification tasks.

The logistic function used in logistic regression is sigmoid function, where sigmoid function represents the linear combination of input features and their associated coefficients. This linear combination is then transformed into a probability score using the logistic function. If the probability exceeds a certain threshold (usually 0.5), the input is classified as belonging to one class, otherwise, it is classified as belonging to the other class.

Logistic regression estimates the parameters (coefficients) of the model using a technique called maximum likelihood estimation. The goal is to find the set of parameters that maximizes the likelihood of observing the given dataset under the assumed logistic regression model. This is typically achieved using optimization algorithms like gradient descent.

### Random Forest

Random Forest is a versatile machine learning algorithm capable of addressing both classification and regression tasks. It operates on the principle of ensemble learning, a strategy that combines multiple models to enhance overall performance. Specifically, Random Forest is a type of ensemble classifier that leverages a collection of decision trees, each trained on different subsets of the dataset.



Figure 5.4 Random Forest

The strength of Random Forest lies in its ability to mitigate the shortcomings of individual decision trees by aggregating their predictions. Instead of relying on the outcome of a single tree, Random Forest considers the collective wisdom of multiple trees. During training, each decision tree in the ensemble is constructed using a random subset of the features and data points. This diversity ensures that the trees are relatively independent and capture different aspects of the data.

When making predictions, Random Forest aggregates the outputs of all decision trees and generates the final prediction based on a majority or weighted vote. This ensemble approach improves the robustness and generalization ability of the model, making it less susceptible to overfitting and noise in the data. Additionally, Random Forest provides a measure of feature importance, indicating which features contribute most significantly to the predictive performance of the model.

### XGBoost

XGBoost, short for eXtreme Gradient Boosting, is an advanced and highly efficient implementation of the gradient boosting algorithm. It is a powerful machine learning technique used for both classification and regression tasks, known for its speed, performance, and scalability. XGBoost has gained widespread popularity and is widely regarded as one of the state-of-the-art algorithms for structured data problems.



Figure 5.5 XGBoost

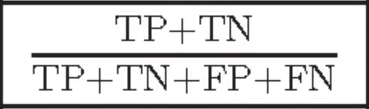
The core concept behind XGBoost is gradient boosting, a technique that builds an ensemble of weak learners (typically decision trees) sequentially to minimize a predefined loss function. However, what sets XGBoost apart is its focus on optimizing both the computational efficiency and predictive accuracy of the model.

One key feature of XGBoost is its innovative regularization techniques, which help prevent overfitting and improve generalization. These techniques include L1 and L2 regularization (also known as "Lasso" and "Ridge" regularization), as well as tree pruning and maximum depth constraints. By incorporating regularization, XGBoost is able to build more robust models that perform well on unseen data.

## Evaluation of Algorithms

### ****Classification Model Evaluation****

* **Accuracy:** This metric simply calculates the proportion of correct predictions made by the model.



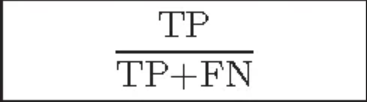
Eq. 1 Accuracy

* **Precision:** This metric focuses on the proportion of positive predictions that were actually correct (upward movement predicted correctly).



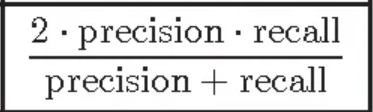
Eq. 2 Precision

* **Recall:** This metric focuses on the proportion of actual upward movements that were correctly predicted by the model.



Eq. 3 Recall

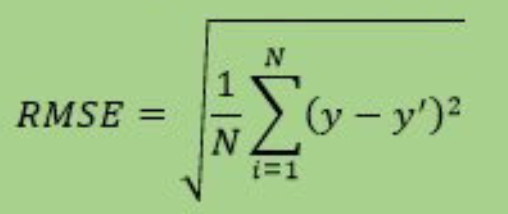
* **F1 Score:** This metric combines precision and recall into a single score, providing a balance between the two. It's calculated using the harmonic mean of precision and recall.



Eq. 4 F1 Score

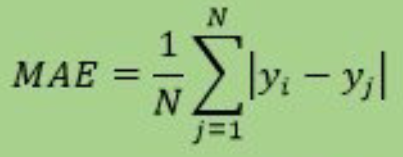
### Regression Models Evaluation

* **Root Mean Squared Error (RMSE):** This metric measures the average squared difference between the predicted price points (ŷ) and the actual price points (y). Lower MSE indicates better performance, as it reflects a smaller average difference between predictions and actual values.



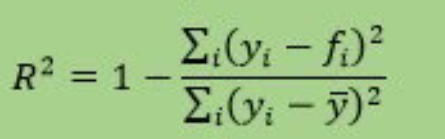
Eq. 5 RMSE

* **Mean Absolute Error (MAE):** This metric measures the average absolute difference between the predicted price points (ŷ) and the actual price points (y). MAE is less sensitive to outliers compared to MSE, as it considers the absolute difference instead of the squared difference.



Eq. 6 MAE

* **R-squared (coefficient of determination):** This metric indicates the proportion of variance in the actual price points (y) that can be explained by the model's predictions (ŷ). A value closer to 1 signifies a better fit, where the model explains a larger proportion of the variance in the actual data.



Eq. 7 R-Squared

# System Requirements and Specifications

This chapter contains brief description about the software and hardware requirements and specifications in the project. It also consists of the information about t the libraries used in the proposed work.

System Requirement Specification is a fundamental document, which forms the foundation of the software development process. It not only lists the requirements of a system but also has a description of its major feature An SRS is basically an organization's understanding of a customer or potential client's system requirements and dependencies at a particular point in time prior to any actual design or development work The SRS also functions as a blueprint for completing a project with as little cost growth as possible. It is important to note that an SRS contains functional and non-functional requirements only.

## Hardware Requirements

Processor: Intel

Hard disc: 500GB

RAM: 4GB or above

System with all standard accessories like monitor, keyboard, mouse, etc.

## Software Requirements

Operating System: Windows

Framework: Jupyter

Language: Python

IDE: Anaconda

AI Tools: ChatGPT , Google Bard

### Software Libraries

**NumPy:** NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

**Matplotlib:** Matplotlib is a cross-platform, data visualization and graphical plotting brary for Python and its numerical extension NumPy. As such, it offers a viable open- source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

**Sci-kit Learn:** Scikit-learn is a popular machine learning library because it is easy to use and provides a wide range of algorithms. It is also well-documented and has a large community of users who can provide support.

# System Design

Execution of the entire program takes place in 4 major steps. The implementation of the four major modules is as follows:

## UML Diagrams

### Use case diagram

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The [Fig. 7.1](#_bookmark25) shows the use case representation of the system

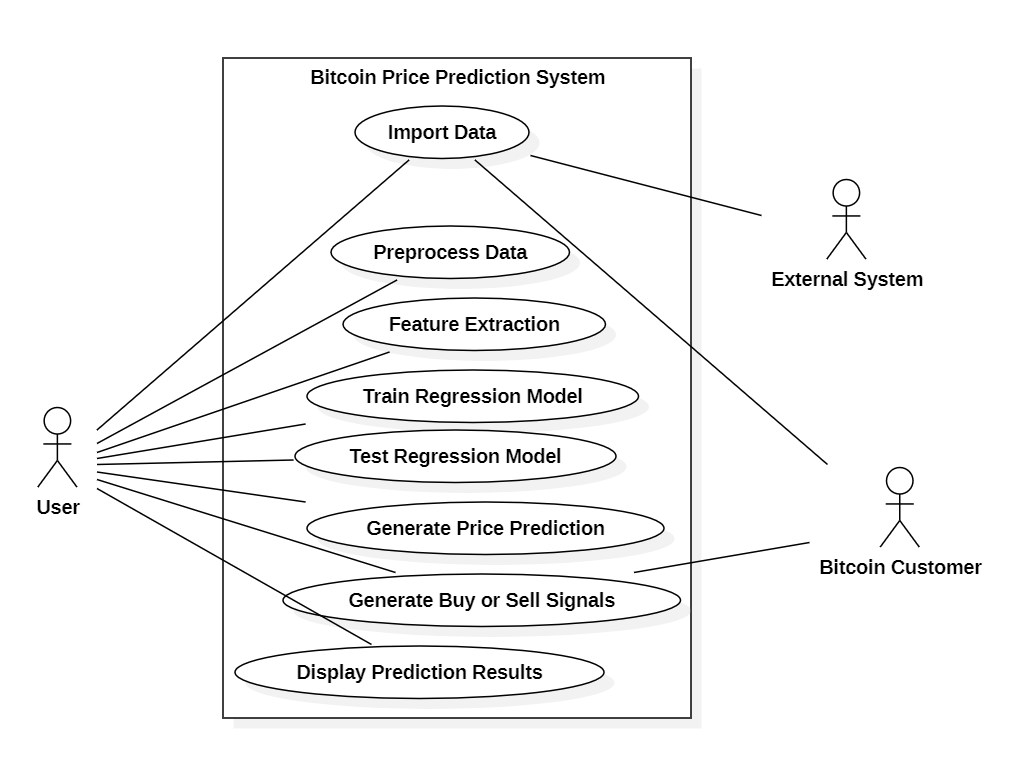


Figure 7.1 Use case Diagram

### Class diagram

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. The [Fig. 7.2](#_bookmark27) shows the class diagram representation of the system.

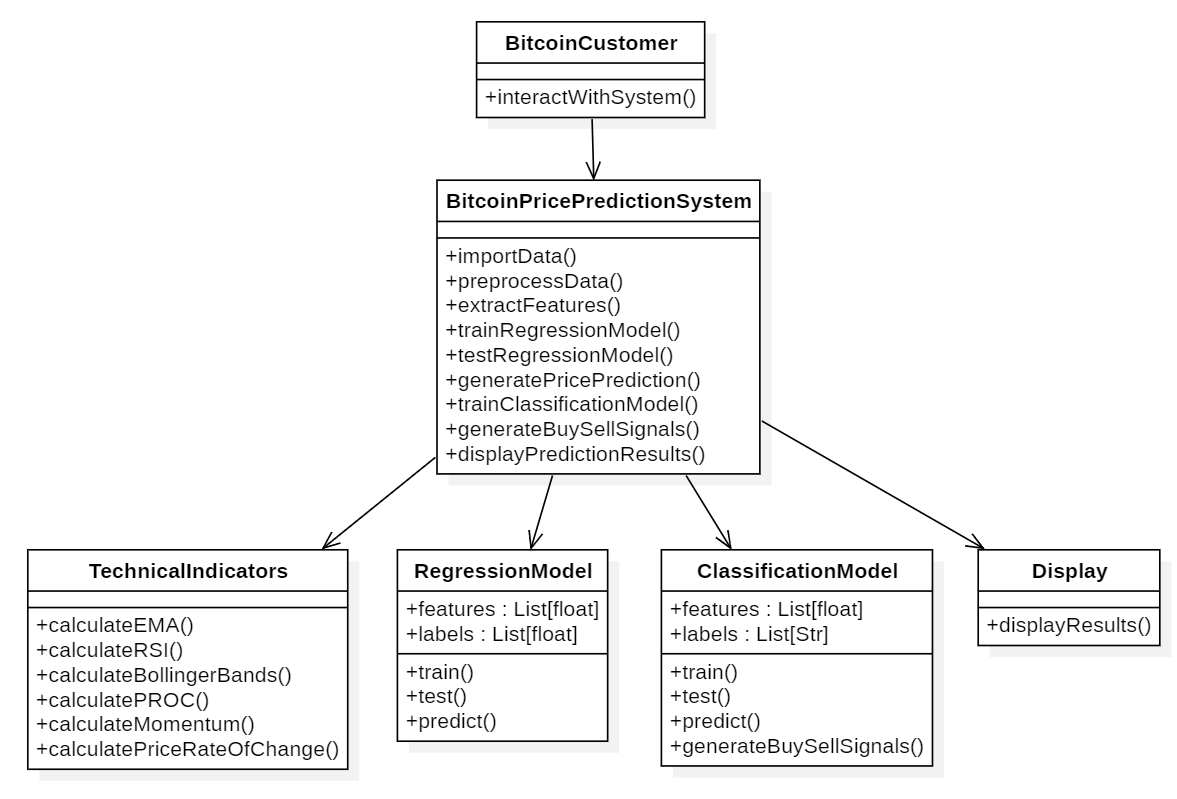


Figure 7.2 Class Diagram

### Activity diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. In UML, an activity diagram provides a view of the behavior of a system by describing the sequence of actions in a process. The [Fig. 7.3](#_bookmark29) shows the activity diagram representation of the system

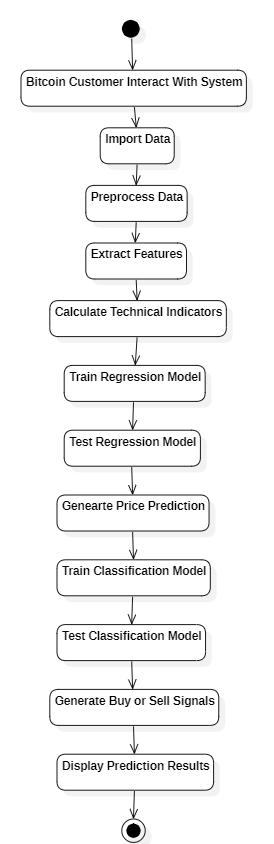


Figure 7.3 Activity Diagram

### Sequence diagram

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. The [Fig. 7.4](#_bookmark31) shows the sequence diagram representation of the system. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them.

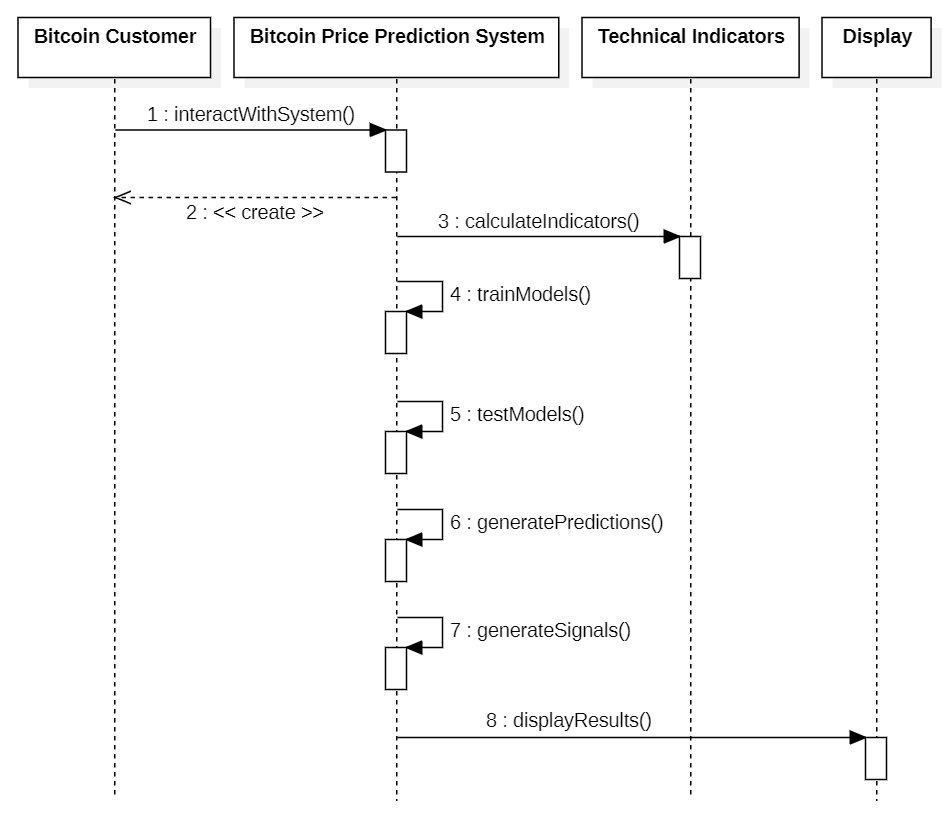


Figure 7.4 Sequence Diagram

# Testing

Software Testing is a process of executing the application with an intent to find any software bugs. It is used to check whether the application met its expectations, and all the functionalities of the application are working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

## Unit Testing

Unit testing refers to tests conducted on a section of code to verify the functionality of that piece of code. This is done at the function level.

## Integration Testing

Integration testing is any type of software testing that seeks to verify the interfaces between components of a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.

## System Testing

System testing tests a completely integrated system to verify that the system meets its requirements.

## Acceptance Testing

Acceptance testing tests the readiness of application, satisfying all requirements.

## Performance Testing

Performance testing is the process of determining the speed or effectiveness of a computer, network, software, program or devices such as response time or millions.

# Implementation

## Loading the Dataset

The code begins by loading the Bitcoin price dataset from a CSV file named "BTC-USD (1).csv" using Pandas' **read\_csv()** function.

## ****Dropping Unnecessary Columns****

After loading the dataset, the code drops unnecessary columns such as "open", "high", "low", and "Adj Close" using the drop() function along the column axis.

## Converting Date to Datetime

The "date" column in the dataset is converted from string format to datetime format using Pandas' **to\_datetime()** function. This allows for easier manipulation and extraction of date-related features.

## Extracting Date Features

Date features such as "DayOfWeek", "Month", and "Year" are extracted from the datetime column using Pandas' **dt.dayofweek**, **dt.month**, and **dt.year** attributes, respectively.

## Feature Engineering

### Technical Indicators

#### Calculating Exponential Moving Average(EMAs)

The code calculates Exponential Moving Averages (EMAs) for different window sizes (e.g., 10, 30, 60, 200) using a custom function named **calculate\_ema()**. These EMAs are commonly used as technical indicators in financial analysis.

#### Calculating Relative Strength Index (RSI)

Another custom function named **calculate\_rsi()** is likely used to calculate the Relative Strength Index (RSI) for different window sizes (e.g., 14, 30, 200). RSI is a momentum oscillator that measures the speed and change of price movements.

#### Calculating Momentum

The code calculates momentum indicators for different periods (e.g., 10, 30) using a custom function named calculate\_momentum(). Momentum measures the rate of price change and can indicate the strength of a trend.

#### Calculating Moving Average Convergence Divergence (MACD)

The MACD indicator is likely calculated using a custom function such as **calculate\_macd().** MACD is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price.

#### Calculating Price Rate of Change (PROC)

The code calculates the Price Rate of Change (PROC) using a custom function named **calculate\_proc().** PROC measures the percentage change in price over a specified period and can indicate the direction and strength of price trends.

#### Calculating Stochastic Oscillator

The Stochastic Oscillator values are calculated for different periods (e.g., 10, 30, 200) using a custom function named **calculate\_stochastic\_oscillator().** The Stochastic Oscillator is a momentum indicator that compares a security's closing price to its price range over a specific period.

#### Calculating Bollinger bands

Calculate Bollinger Bands using a custom function like **calculate\_bbands()** to determine the upper and lower bands based on the moving average and standard deviation.

#### Signal Generation

Trading signals are likely generated based on the calculated technical indicators and features. These signals indicate whether to buy, sell, or hold Bitcoin based on predefined criteria or trading strategies.

## Splitting Data and Model Training

* Split the dataset into training and testing sets, ensuring that features (including the calculated technical indicators) and target variables are appropriately separated.
* Train a machine learning model (classification or regression) using the training data. Features are selected based on their relevance to predicting future Bitcoin prices.

## Predicting future prices

Use the trained model to predict future Bitcoin prices based on the selected features and technical indicators.

## Evaluating Model Performance

Evaluate the performance of the trained model using appropriate metrics such as accuracy, precision, recall, F1-score, Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), etc.

## Generating Trading Signals (Signal Generation)

Based on the predicted future prices and technical indicators, generate trading signals to guide investment decisions. These signals provide actionable insights for investors to buy, sell, or hold Bitcoin.

## Implementation GitHub link

* <https://github.com/Tarunutla15/Bitcoin_Price_Prediction.git>

# Result

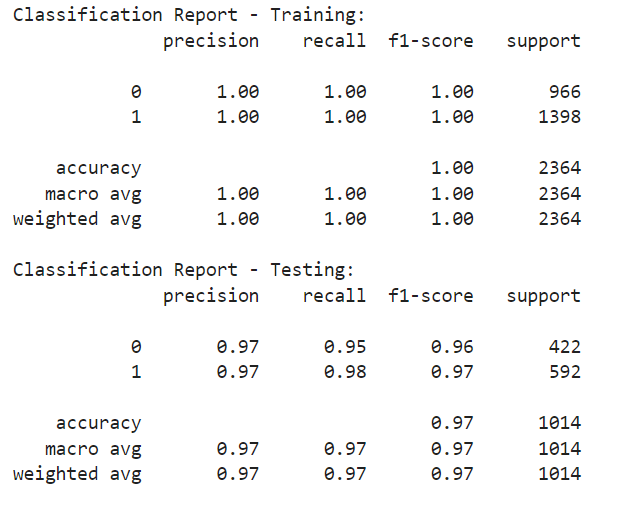
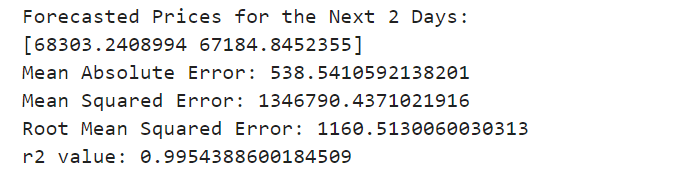


Figure 10.1 Classification report



# Conclusion

In this project, we developed a comprehensive approach to predict Bitcoin prices using machine learning techniques and technical indicators. We began by preprocessing the data, which involved loading the dataset, removing unnecessary columns, and extracting date features.

Feature engineering played a crucial role in enhancing the predictive power of our model. We calculated various technical indicators such as Exponential Moving Averages (EMA), Relative Strength Index (RSI), Momentum, Moving Average Convergence Divergence (MACD), Price Rate of Change (PROC), and Stochastic Oscillator. These indicators provided valuable insights into market trends and momentum, which were utilized by our predictive model.

For model training, we utilized random forest to generate buy/sell signals based on EMA comparison. Additionally, we employed RandomForestRegressor to forecast future Bitcoin prices for the next 2 days. A confidence threshold was defined to generate buy/sell signals based on predicted price changes.

While our approach demonstrates promising results, it's essential to acknowledge certain limitations and areas for improvement. Firstly, the predictive performance of our model should be thoroughly evaluated using appropriate performance metrics such as accuracy, precision, recall, and F1-score. Moreover, further optimization and fine-tuning of model hyperparameters may enhance the model's predictive accuracy and robustness.

# Future Enhancement

Integrating Quantum Machine Learning (QML) techniques into your Bitcoin price prediction project holds significant promise for advancing financial forecasting methodologies. Quantum computing's exceptional computational power offers novel avenues for analyzing complex datasets and deriving insights that traditional approaches may overlook. By leveraging quantum algorithms for feature selection, your model can identify the most relevant indicators from a plethora of options, enhancing the robustness and accuracy of predictions.

Moreover, developing machine learning models using quantum computing frameworks such as Qiskit or TensorFlow Quantum introduces a new paradigm in predictive modeling. Quantum models have the potential to outperform classical algorithms by harnessing the parallelism and optimization capabilities inherent in quantum systems. Additionally, hybrid quantum-classical approaches offer a compelling strategy for combining the strengths of classical and quantum computing. These hybrid models can leverage quantum processing for specific tasks while utilizing classical resources for data pre-processing and post-processing, resulting in more efficient and accurate predictions.

Furthermore, quantum-inspired optimization algorithms provide efficient solutions for fine-tuning model parameters and hyperparameters, further enhancing predictive performance. By integrating these QML techniques into your project, you not only advance the state-of-the-art in financial forecasting but also contribute to the broader field of quantum computing applications. Ultimately, these advancements empower future generations of traders and investors with sophisticated tools and insights to navigate the dynamic cryptocurrency market with confidence and precision.

# References

|  |  |
| --- | --- |
| [1] | G. Hileman and M. Rauchs, “Global cryptocurrency benchmarking study,” Cambridge Centre for Alternative Finance, vol. 33, no. 1, pp. 33–113, 2017. |
| [2] | P. Treleaven, R. G. Brown, and D. Yang, “Blockchain technology in finance,” Computer, vol. 50, no. 9, pp. 14–17, 2017. |
| [3] | CoinGecko, “Cryptocurrency prices, charts, and crypto market cap,” Available at https://www.coingecko.com/, Accessed: 02-Jul-2022. |
| [4] | M. Saad, J. Choi, D. Nyang, J. Kim, and A. Mohaisen, “Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions,” IEEE Systems Journal, vol. 14, no. 1, pp. 321–332, 2019. |
| [5] | Y. Shynkevich, T. M. McGinnity, S. A. Coleman, A. Belatreche, and Y. Li, “Forecasting price movements using technical indicators: Investigating the impact of varying input window length,” Neurocomputing, vol. 264, pp. 71–88, 2017. |
| [6] | P. Oncharoen and P. Vateekul, “Deep learning for stock market prediction using event embedding and technical indicators,” in 2018 5th international conference on advanced informatics: concept theory and applications (ICAICTA). IEEE, 2018, pp. 19–24. |
| [7] | S. McNally, J. Roche, and S. Caton, “Predicting the price of bitcoin using machine learning,” in 2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP). IEEE, 2018, pp. 339–343. |
| [8] | P. Jay, V. Kalariya, P. Parmar, S. Tanwar, N. Kumar, and M. Alazab, “Stochastic neural networks for cryptocurrency price prediction,” Ieee access, vol. 8, pp. 82 804–82 818, 2020. |
| [9] | H. Singh and P. Agarwal, “Empirical analysis of bitcoin market volatility using supervised learning approach,” in 2018 Eleventh International Conference on Contemporary Computing (IC3). IEEE, 2018, pp. 1–5. |
| [10] | S. B. Achelis, “Technical analysis from a to z,” 2001. |
| [11] | ] J. M. Lucas and M. S. Saccucci, “Exponentially weighted moving average control schemes: properties and enhancements,” Technometrics, vol. 32, no. 1, pp. 1–12, 1990. |
| [12] | G. Appel, Technical analysis: power tools for active investors. FT Press, 2005. |
| [13] | J. W. Wilder, New concepts in technical trading systems. Trend Research, 1978. |
| [14] | L. K. Chan, N. Jegadeesh, and J. Lakonishok, “Momentum strategies,” The Journal of Finance, vol. 51, no. 5, pp. 1681–1713, 1996. |
| [15] | ] C. Cortes and V. Vapnik, “Support-vector networks,” Machine learning, vol. 20, no. 3, pp. 273–297, 1995. |
| [16] | T. K. Ho, “Random decision forests,” in Proceedings of 3rd international conference on document analysis and recognition, vol. 1. IEEE, 1995, pp. 278–282. |
| [17] | A. Geron, ´ Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. ” O’Reilly Media, Inc.”, 2019 |
| [18] | Binance, “Binance api,” Available at https://www.binance.com/en/binance-api, Accessed: 16-Jul-2022. |
| [19] | G. Biau and E. Scornet, “A random forest guided tour,” Test, vol. 25, no. 2, pp. 197–227, 2016. |