**PHENIKAA UNIVERSITY**

**FALCULTY OF ELECTRONIC ENGINEERING**

Logo, company name

Description automatically generated

**MACHINE LEARNING MID-TERM PROJECT REPORT**

**Project: Predicting Cryptocurrency Direction using Classification Algorithms with 5 Classes**

1. Duong Doan Tung

2. Nguyen Trong Huy Hoang

3. Le Hoang Nam

**HA NOI, 03/2023**

TABLE OF CONTENTS

[Section 1: ABSTRACT 1](#_Toc129553994)

[Section 2: PROJECT OVERVIEW 2](#_Toc129553995)

[1. Inspiration 2](#_Toc129553996)

[2. Basic Idea 2](#_Toc129553997)

[Section 3: PROJECT DETAILS 3](#_Toc129553998)

[1. The dataset 3](#_Toc129553999)

[1.1. Acquiring the dataset 3](#_Toc129554000)

[1.2. Calculate the true labels 4](#_Toc129554001)

[2. Feature Engineering 6](#_Toc129554002)

[2.1. Technical Indicators 6](#_Toc129554003)

[2.1.1. Simple Moving Average (SMA) 6](#_Toc129554004)

[2.1.2. Exponential Moving Average (EMA) 6](#_Toc129554005)

[2.1.3. Relative Strength Index (RSI) 6](#_Toc129554006)

[2.1.4. Bollinger Bands 7](#_Toc129554007)

[2.1.5. Moving Average Convergence Divergence (MACD) 7](#_Toc129554008)

[2.1.6. On Balance Volume (OBV) 7](#_Toc129554009)

**WORKLOAD DISTRIBUTION**

|  |  |
| --- | --- |
| Full name | Task |
| Duong Doan Tung | o |
| Nguyen Trong Huy Hoang |  |
| Le Hoang Nam |  |

Project include:

1. Project report
2. Presentation file and video
3. Python files, Jupyter Notebook
4. Dataset sample

Full project material can be found here:

**Project repository:** <https://github.com/dtungpka/CPC5>

*Special thanks to our teacher Dr. Huy Minh Le for his help in this project*

# Section 1: ABSTRACT

The use of classification algorithms for predicting cryptocurrency prices has garnered significant interest in recent years. In this study, we focus exclusively on using the K-Nearest Neighbors (KNN) algorithm to predict the direction of cryptocurrency prices with five classes: small increase, large increase, small decrease, large decrease, and no change.

We used a dataset containing historical cryptocurrency price data and calculate some technical indicators from it. We evaluated the performance of the algorithm using accuracy, precision, recall, and F1 score metrics to determine the performance of KNN in predicting cryptocurrency price direction with five classes.

Our results demonstrate that the KNN algorithm is a viable tool for predicting cryptocurrency price direction with multiple categories. In particular, the KNN algorithm achieved an accuracy of 0.58, precision of 0.54, recall of 0.54, and F1 score of 0.54, surpassing the performance of other classification algorithms used for comparison.

The findings of this study could potentially benefit investors and traders in making more informed and nuanced decisions within the cryptocurrency market. This study provides valuable insights into the use of classification algorithms for predicting cryptocurrency prices, specifically highlighting the efficacy of the KNN algorithm for this purpose.

# Section 2: PROJECT OVERVIEW

## Inspiration

The advent of cryptocurrencies has introduced a new phase of digital financial transactions. The popularity and volatility of cryptocurrency prices have led to increased interest in predicting their direction. Accurate price predictions can potentially benefit investors and traders by providing insights into market trends and facilitating informed decision-making.

Despite the rapid growth of the cryptocurrency market, predicting cryptocurrency prices, or stock price in general, remains a challenging task. The complex and dynamic nature of this market presents unique challenges for traditional financial analysis techniques. Therefore, there is a growing need to explore alternative methods for predicting cryptocurrency prices. Because of that, we decided to start this project: **Cryptocurrency Price Classification with 5 classes (CPC5)**

The inspiration for this study stems from the need to explore new approaches for predicting cryptocurrency prices. By focusing on the KNN algorithm, we aim to contribute to the growing body of research on machine learning techniques for predicting cryptocurrency prices. The findings of this study have the potential to benefit investors and traders in making more informed and nuanced decisions within the cryptocurrency market.

## Basic Idea

To accomplish our objective of predicting cryptocurrency price direction using the K-Nearest Neighbors (KNN) algorithm, we utilized a dataset consisting of historical cryptocurrency price data and technical indicators. We first obtained the price data of a cryptocurrency from Binance Public Data, one of the most reliable, biggest sources for cryptocurrency data.

After obtaining the price data, we calculated several technical indicators such as the Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Boilinger Bands, Average Directional Movement Index (ADX), and others. These technical indicators provide valuable information on market trends, momentum, and potential price movements.

To ensure the optimal performance of our model, we employed a feature selection process to identify the most relevant technical indicators. This process involves filtering out irrelevant features and identifying the ones that provide the most significant predictive power for our model.

Once we obtained the most suitable features, we proceeded to calculate the true labels for our model. To achieve this, we used a clustering algorithm, such as k-means, to group the data into several clusters. We then assigned a label to each cluster, representing the expected direction of price movement for that cluster.

By utilizing this approach, we created a dataset with true labels that accurately reflect the price direction of the cryptocurrency. This dataset was then used to train and evaluate our KNN model, with the aim of accurately predicting the direction of cryptocurrency prices.

Our ultimate goal is to create a trading bot that can leverage our KNN model's predictive power to make profitable trades. The trading bot will use the predicted price direction to determine when to enter or exit trades.

To increase the trading bot's profitability, we will incorporate a modified version of the Martingale strategy. The Martingale strategy is a popular money management technique that involves doubling the trading volume after each loss, with the aim of recouping all losses in a single winning trade.

However, the Martingale strategy has its limitations, and in its purest form, it can lead to significant losses. To address this, we will modify the Martingale strategy to incorporate risk management measures that limit losses and maximize profits.

By combining our KNN model with a modified Martingale strategy, we aim to create a robust and profitable trading bot that can navigate the volatile cryptocurrency market. This study represents a crucial step towards realizing this goal and has the potential to revolutionize the cryptocurrency trading landscape.

# Section 3: PROJECT DETAILS

## The dataset

### Acquiring the dataset

To procure the comprehensive dataset required for our study, we developed a script, *binance\_data.py*, which utilizes the Binance API to download historical cryptocurrency price data. The script is available in our GitHub repository.

To ensure the data downloaded is relevant and aligned with our study's goals, we specified the time range in the details.json file. This file provides the script with the necessary information to download data for the specified time range.

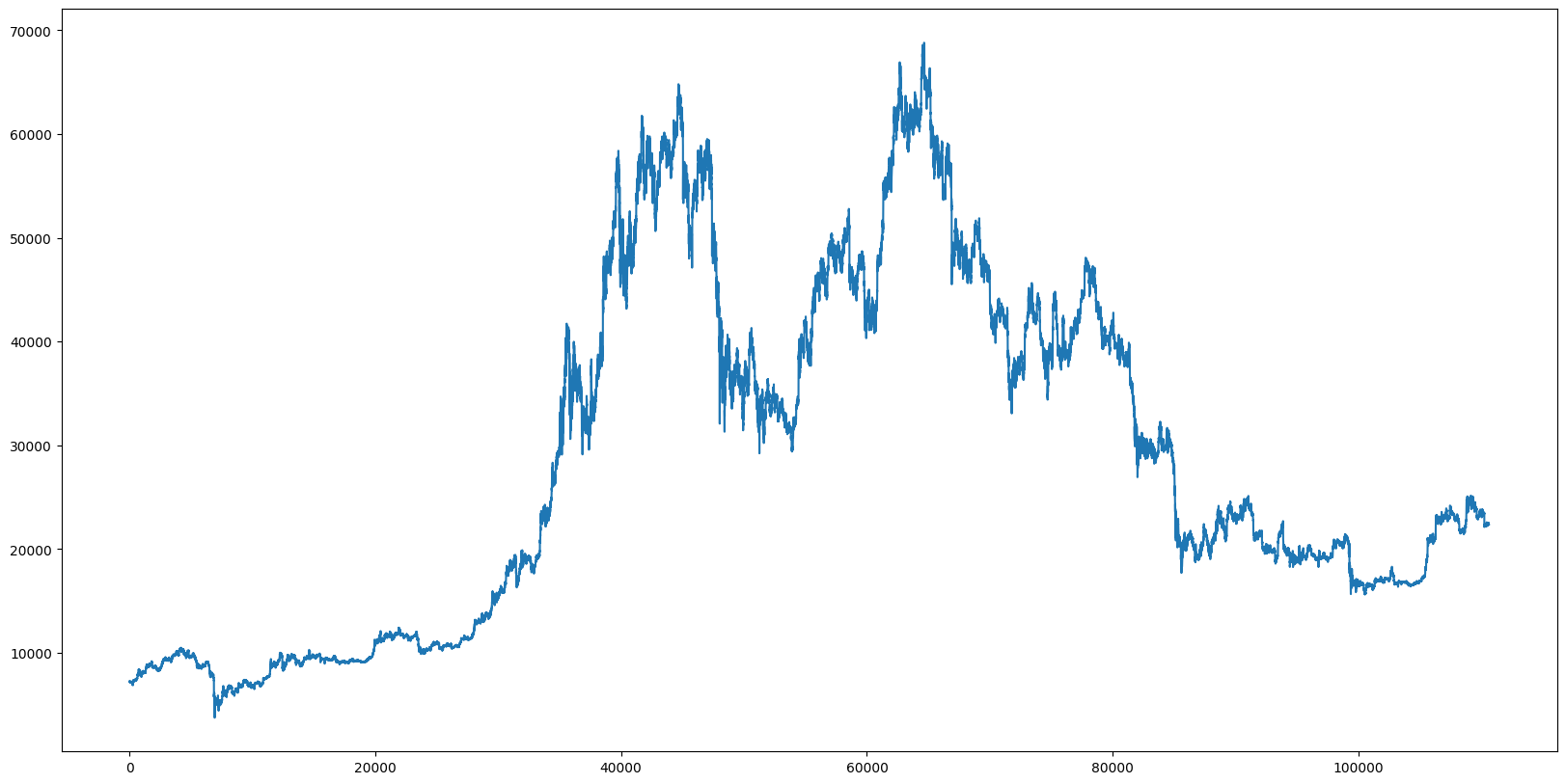


Figure . BTCUSDT in 15 minute interval

Figure 1 above shows the BTCUSDT price in 15 minute interval, from 31/12/2019 to 06/03/2023, with total of 110607 data points.



Figure . Data structure

Each line of the dataset contains 10 values, but we only use 5 of them: open,high,low,close,volume.

### Calculate the true labels

To classify cryptocurrency price movements into 5 different labels: "Large decrease", "Small decrease", "No change", "Small increase", "Large increase", we first calculate the average price each row by add the opening price (open) and closing price (close) for each row in the dataset and divide by 2.

With is the open price, is the close price

Then, the percentage change in price can be calculated as follows:

We can then visualize it:

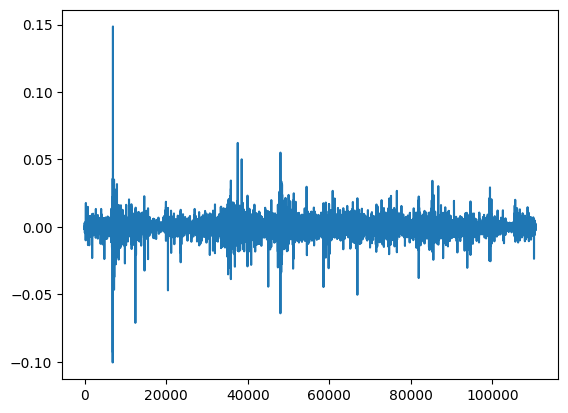


Figure . Percentage change in price

Based on this data, we used k-means clustering algorithm to cluster the data into 5 groups based on their similarity.[[1]](#footnote-1)

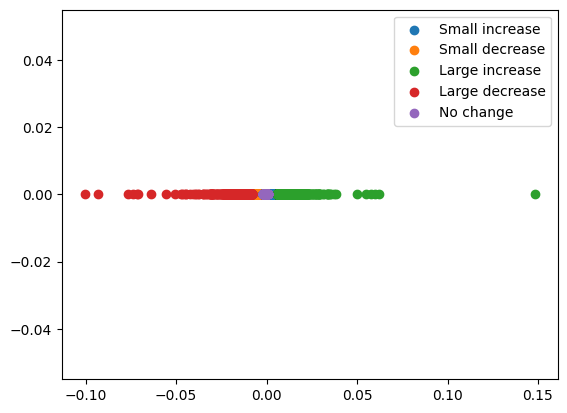


Figure . K-mean clustering

## Feature Engineering

Our next step is to select, calculate technical indicators, and choose what to use in our model.

### Technical Indicators

Exclude the first 5 value we already have from the dataset: open, close, low, high, volume; we need to calculate the remaining technical indicators. For better understanding, let is the time period for calculating some of the indicators.[[2]](#footnote-2)

#### Simple Moving Average (SMA)

The Simple Moving Average (SMA) is a commonly used technical analysis indicator that helps to identify trends in the price of an asset, it can be calculated by:

#### Exponential Moving Average (EMA)

The Exponentially Moving Average (EMA) is a quantitative or statistical measure used to model or describe a time series.

With

#### Relative Strength Index (RSI)

The Relative Strength Index (RSI) is a momentum oscillator that measures the speed and change of price movements.

Where AG, AL is the average gain, average loss:

#### Bollinger Bands

Bollinger Bands are volatility bands placed above and below a moving average. Volatility is based on the standard deviation, which changes as volatility increases and decreases.

First we calculate the standard deviation:

Calculate the upper Bollinger bands:

Calculate the lower Bollinger bands:

#### Moving Average Convergence Divergence (MACD)

Calculate the short term exponential moving average (SEMA):

With

Then the moving average convergence divergence:

#### On Balance Volume (OBV)

Let be the volume of the current period.

#### Average true range (ATR)

Let be the current high price of the current period, be the current low price of the current period.

Then:

#### Fibonacci Retracement

Fibonacci retracement is a popular technical analysis tool used to identify potential levels of support and resistance in a financial market. It is based on the idea that prices will often retrace a predictable portion of a move, after which they will continue to move in the original direction. The key Fibonacci retracement levels are 23.6%, 38.2%, 50%, 61.8%, and 78.6%. These levels are calculated by taking the difference between the high and low of a price move, multiplying it by the Fibonacci ratios and then adding or subtracting the result from the starting price of the move.

### Feature Selection

1. The k-means we implement in this project is from sklearn. Note that this is diffirent from our KNN implementation, which is not from sklearn library. This is due to lack of time in the mid-term, and we will use numpy/cuda entirely instead of sklearn in the final-term report. [↑](#footnote-ref-1)
2. We choose value 96 because the dataset is 15-min period, so 96\*15 = 1440 minutes which is 1 day. [↑](#footnote-ref-2)