**Bitcoin History Project (CMPSC 463) By Sam Axler**

This project utilizes various data analysis techniques to evaluate the trends, anomalies, and maximum gains for the price of Bitcoin over a certain time period. The following problems are addressed and answered.

Below is a simple flowchart of the program and how it works: A diagram of data processing

Description automatically generated

**Raw Imported Data:** Data from yfinance is imported. Users can change the period and interval values to get more or less data with more or less specific time frames.

**Data Cleaned and Split:** The raw data is in an unfriendly dataframe, so the dataframe is split into a list containing the numericized date, and the actual price data.

**Data Sorted with Merge Sort:** Even though the dataset comes pre-sorted by date, we need to double check to make sure it’s sorted correctly. Merge Sort is used to accomplish this.

**Outliers Found with KNN:** KNN from sklearn is implemented to find outliers. Users have access to the k value as well as the percentage value to find more or less outliers. A k value of 2 and a percentage value of 95 is most accurate in my findings.

**MaxGain found and printed:** Kadane’s Algorithm is used to find the maximum gain. For Bitcoin, this is a lot, given that 5 years ago, it started at almost nothing.

**Data is graphed:** The data and outliers are graphed through MatPlotLib. An example graph can be found below:

A graph with a line and a line

Description automatically generated



The Orange line represents the outliers, and the blue line is the raw data graphed over time.

A screen shot of a computer code

Description automatically generated

This is the only function users need to edit. Users can edit the k, p, and two values under ‘getData’. Here are examples of graphs with different k and p values:

K=2,P=90

A line graph with orange and blue lines

Description automatically generated

K=2,P=50

A line graph with blue and orange lines

Description automatically generated

K = 10, P = 99

A line graph with blue and orange lines

Description automatically generated

In this project, I had a good amount of trouble approaching transforming the dataset into a usable set of data by Merge Sort, KNN, ETC. At first, I had no clue where to start. However, I quickly found out that I was able to turn the timestamp into a string, then split it into two parts, and take only the first part, THEN remove all non-numerical data from the string, and turn it back into an integer. It was an unorthodox approach.

The second problem was graphing the data in the first place. Since the X value, the dates, were formatted by Year, Month, Day, there would be big jumps if I simply used the date as the X axis when plotting the data. Instead, I opted to just use the index of the element in the list instead of the day, as it was not only more visually appealing, but made more sense.

If I had more time with this project, I would’ve found a way to better graph the dates and time.