Methodology:

The application makes extensive use of the Coinbase API and various data science tools. They are two classes in the application, which are GetData and ComputeStatistic. As the name suggests the GetData is used to acquire the data set that is needed. The ComputeStatistic computes the statistics of the retrieved data set.

The workhorse of the GetData class is the historical rate function. The function connects to the Coinbase API and makes the get request to retrieve the historical rate of products within a time frame. The historical rates data set is available for public use through the Coinbase API gateway, which makes it the suitable data set for this program. To fully understand the data retrieved by the historical rate function a crash course on crypto trading is needed, which is out of the scope of this article. But in summary, the function retrieves the high, low, open, and close prices of a ticker in a particular time interval. Additionally, the function does preliminary data preprocessing, it changes the data set type from text to float for computational purposes. The resulting data is used as the input of the Compute\_statistic class. The Compute\_statistic class is where the magic happens. The class preprocesses the data set, trains the model, and makes predictions. The next chapters give detailed steps on how these actions are performed in the Compute\_statistic class.

Data preprocessing

Various steps were required to transform the data set into a usable form for prediction. This is the case because of the stochastic nature of the crypto market. These are the preprocessing steps:

1. The data is split into target and domain data. The target data set is the close price, while the domain data set are the time, the open price, the low price, and the close price.
2. Smooth the data, the smooth\_data function applies an exponential moving average on the data. The exponential moving average is a weighted moving average that gives more weight to recent price data (Maverick, 2022). The output of this function is a data set that is smoother with less extreme values. Figure 1, the orange trend line is the output of the action on the data set.
3. Find the change in price between different time intervals. The function that performs this task is the change\_calculator. The input of this function is the output from the smooth\_data function. The change in price data is the needed data as it shows the relationship between changes in the different types of prices.



**Figure 1:** Action of EMA on the data set. (Binance, 2022)

1. The next step is to standardize the data, the step is important as it allows the calculation of permutation testing (i.e. p-values) for the data columns. The standardizer function performs this task. The input of the standardizer function is the output of the change\_calculator function.
2. In other to get variables that are significant to the target data, the p-values of the data set were calculated. The condition for acceptance is that the variable must have a p-value less than **0.5**. The variables that meet this condition are used as the data independent variables.

These are the five preprocessing that were done on the data set before it was trained with the linear models. This project uses Sklearn linear regression model to make close price predictions and the logistic regression model to predict the trend direction.

Price and outcome prediction:

As stated earlier, Sklearn linear regression model was used to train the data and the function in the code that does the task is the ln\_reg function. The process involves firstly, splitting the data into train and test sets. The training set was **80%**, and the test was **20%** of the preprocessed data. The target data set was structured so that the predicted price was the most recent data. In the best cases, the model had **81%** prediction accuracy. The trend direction prediction is a classification problem, hence, the usage of the logistic regression model. The function that performs this task in the code is the logit\_reg function. The trend is either up or down, the uptrend is represented by the numeric value **1** and the downtrend is represented by the numeric value **0**. In the best cases, the prediction accuracy was **97%.**

**NOTE:** The above report is a summary of everything I did on the code. I tried to keep it as short as possible, hence, I omitted the details. If you think there is something that needs to be added, please add it.