Machine learning project

Team members:

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Group** |
| Rajab Hassan Abd Al-Monem  **(Team leader)** |  | 1 |
| Ziad Ahmed Saeed |  | 2 |
| Yusuf Sameh Hussen Mohammed |  | 1 |
| Mohammed Mahmoud Mohammed |  | 2 |
| Zienab Abdelsamea Mohamed Ahmed |  | 2 |

***K-means***

* **K-Means Clustering Algorithm**

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science

## **What is K-Means Algorithm?**

K-Means Clustering is an It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

* **The k-means clustering algorithm mainly performs two tasks**:
* Determines the best value for K center points or centroids by an iterative process.
* Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.
* **The below diagram explains the working of the K-means Clustering Algorithm:**



## **How does the K-Means Algorithm Work?**

The working of the K-Means algorithm is explained in the below steps:

**Step-1:** Select the number K to decide the number of clusters.

**Step-2:** Select random K points or centroids. (It can be other from the input dataset).

**Step-3:** Assign each data point to their closest centroid, which will form the predefined K clusters

**Step-4:** Calculate the variance and place a new centroid of each cluster.

**Step-5:** Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

**Step-6:** If any reassignment occurs, then go to step-4 else go to FINISH.

**Step-7**: The model is ready.

## **How to choose the value of "K number of clusters" in K-means Clustering?**

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

### **Elbow Method**

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

WCSS= ∑Pi in Cluster1 distance(Pi C1)2 +∑Pi in Cluster2distance(Pi C2)2+∑Pi in CLuster3 distance(Pi C3)2

In the above formula of WCSS,

∑Pi in Cluster1 distance(Pi C1)2: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

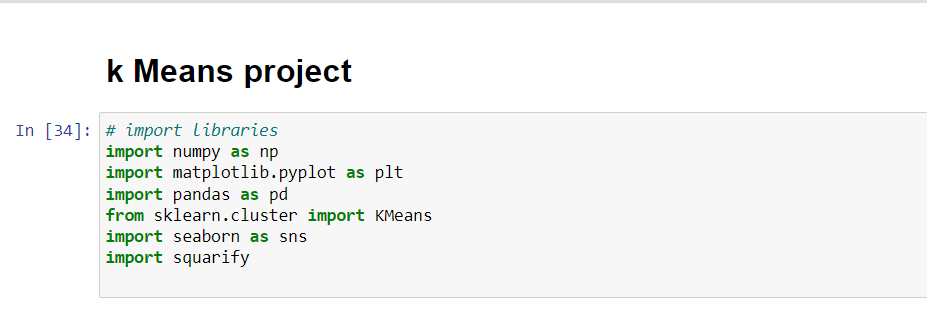
* It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
* For each value of K, calculates the WCSS value.
* Plots a curve between calculated WCSS values and the number of clusters K.
* The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value Of k .

**The steps to be followed for the implementation are given below:**

* **Data Pre-processing**
* **Finding the optimal number of clusters using the elbow method**
* **Training the K-means algorithm on the training dataset**
* **Visualizing the clusters**

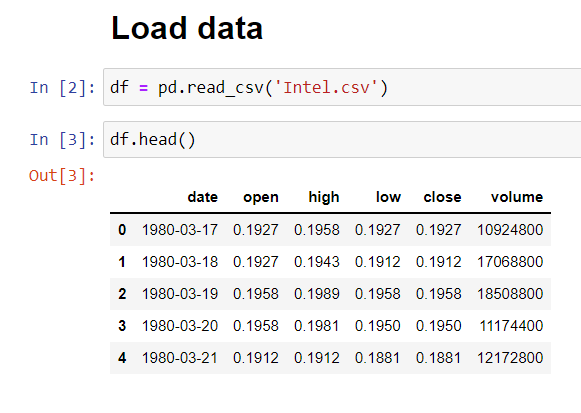
### **Step-1: Data pre-processing Step**

* Import libraries



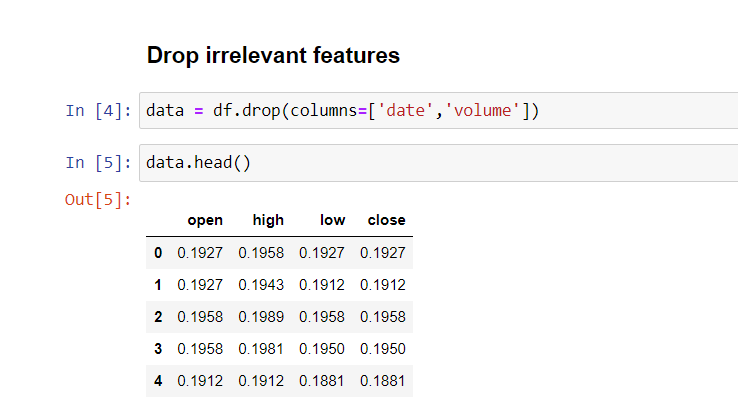
we need pandas to deal with data as data frame, using matplotlib and seaborn to visualize the graphs and display the plots between the features, squarify to mapping data in trees and we need sklearn to use the built-in function to build the model of K Means.

* Load data

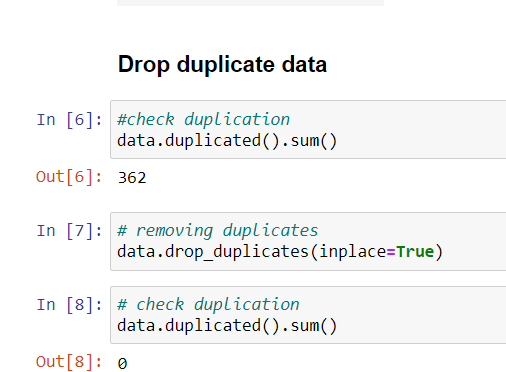


* Dealing with irrelevant data
* We have been removed irrelevant columns
* Volume attribute indicate to the number of buying and selling the stock of the company that happened between open and close time
* Since Intel is a famous company -almost every one use its products- , it’s normal to have a high volume per day.
* The mean on this attribute is 50718441.74527148 which is a huge number.
* Since k mean algorithm use Euclidean distance, this attribute will cause a high effect on the results (because high weight).
* Also, other four attributes have the same scale which is different from volume scale.
* so, if we applied the algorithm with volume attribute the algorithm will be biased to it and ignore the other features.

It’s better to remove Volume attribute

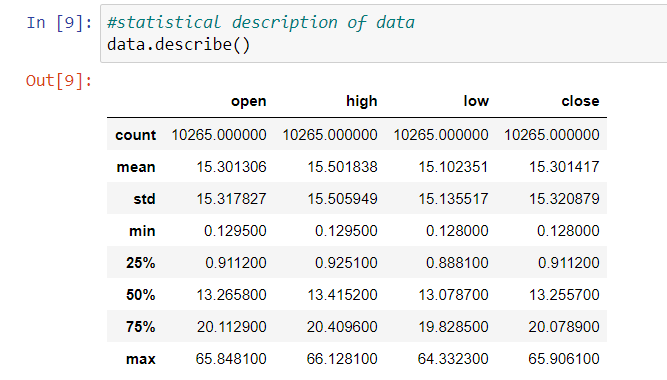


* Dealing with duplication

******

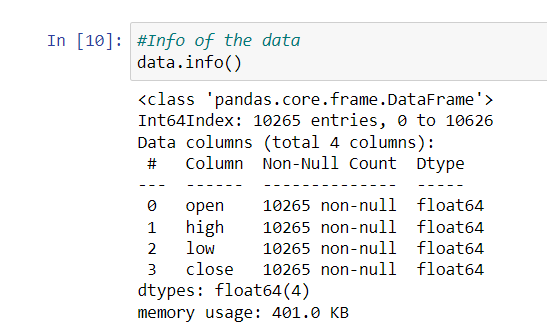
* **Describe statistics of the data**

The describe() function computes a summary of statistics pertaining to the DataFrame columns. This function gives the mean, std and IQR values. And function excludes the character columns and given summary about numeric columns.

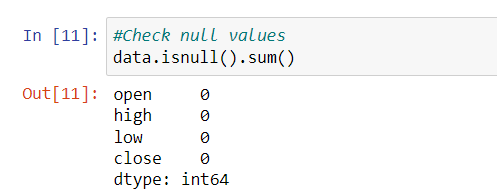


* Knowing the info data

The info() method prints information about the DataFrame. The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-null values).



* Checking null values

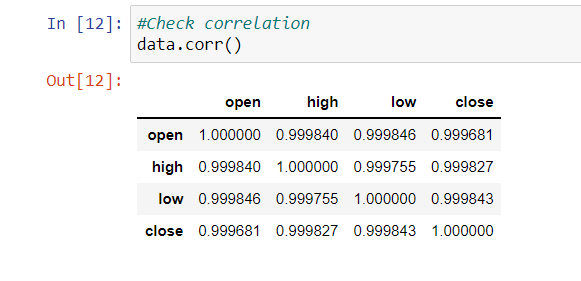


* Correlation of the data

Data is highly correlated, that is because a lot of reasons

* It’s known that international exchange is correlated with Every local exchange and every local exchange follows NASDAQ (Famous American exchange),
* Intel is an American company Opening and closing happen in the same day
* Investors don’t invest in any company unless they do a lot of research related to the investment.
* And this research is highly correlated to each one.
* For example, if war happened in a country which has a lot of famous companies then no one will invest in any company in this country because there is a high chance that these companies will be destroyed
* Sometimes a terrible problem happens in one market and causes a mess in another market, famous example in 2008
* The collapse of the American real estate market led to the collapse of the global stock market in all its markets

Also, it’s recommended to invest in stock which has a high correlation



* The shape of data

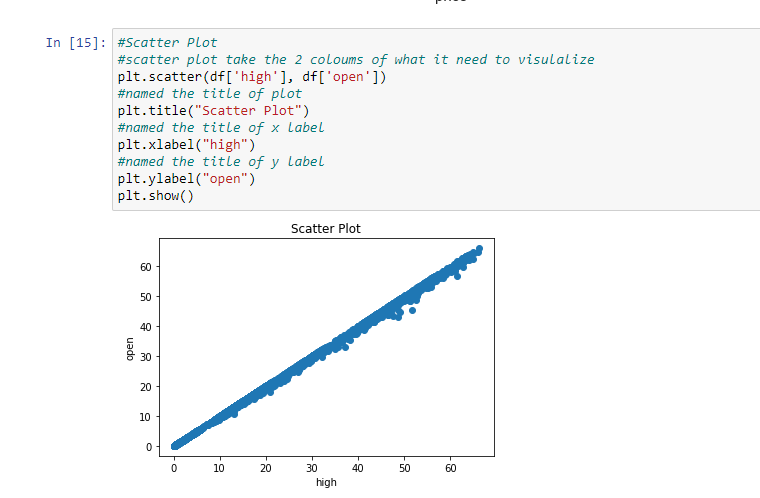


* Explore the data and relationships between columns

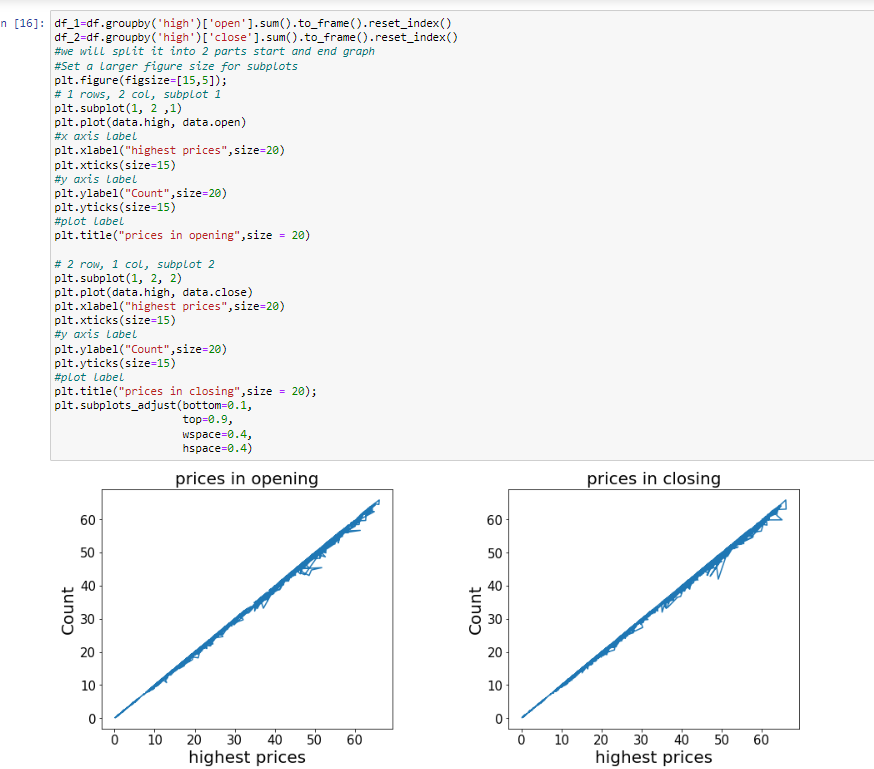
Histogram compares the high price with the opening price we notice that there is no change between opening and high price

****

A Scatter plot between high and open refer to high Positive Correlation between Open and high



Compare high price in opening and closing price

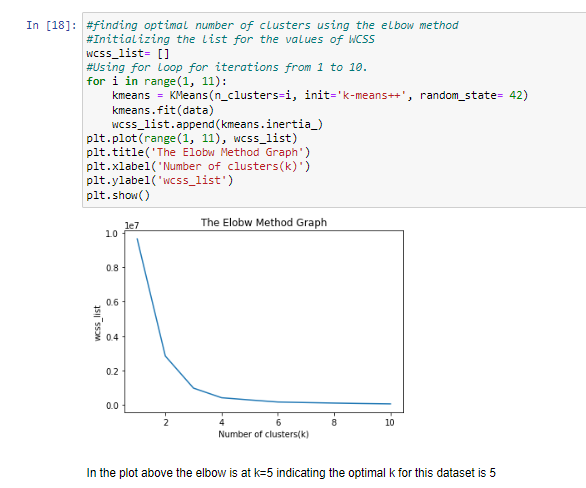


**Step-2:** **Finding the optimal number of clusters using the elbow method**

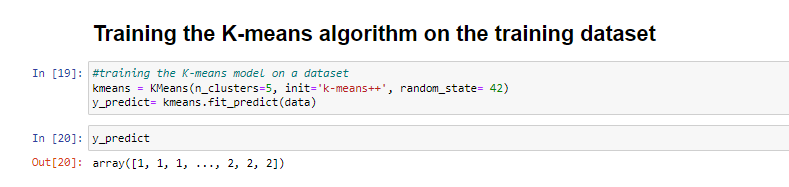
we will try to find the optimal number of clusters for our clustering problem. So, here we are going to use the elbow method for this purpose.

As we know, the elbow method uses the WCSS concept to draw the plot by plotting WCSS values on the Y-axis and the number of clusters on the X-axis. So we are going to calculate the value for WCSS for different k values ranging from 1 to 10.

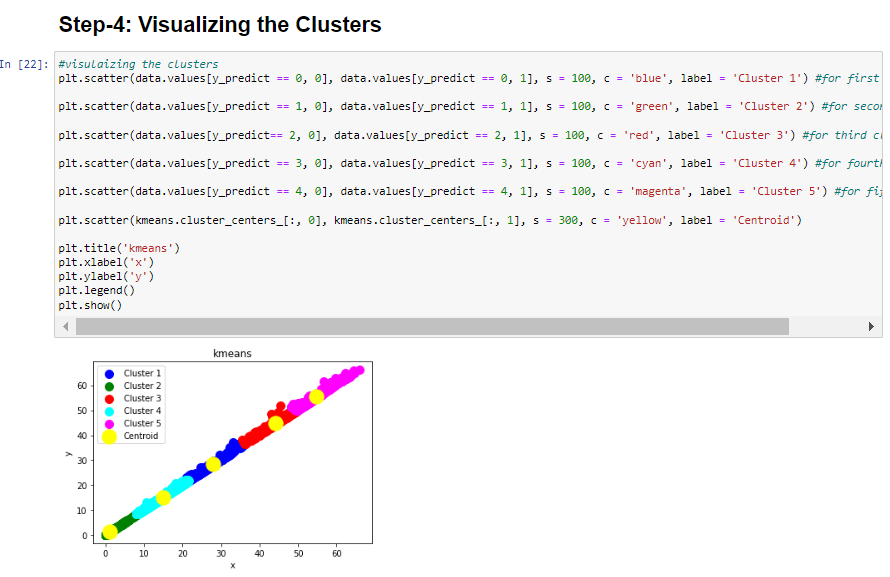
WCSS is the sum of the squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K value, the plot looks like an Elbow. As the number of clusters increases, the WCSS value will start to decrease. WCSS value is largest when K = 1.



**Step 3: Training the K-means algorithm on the training dataset**

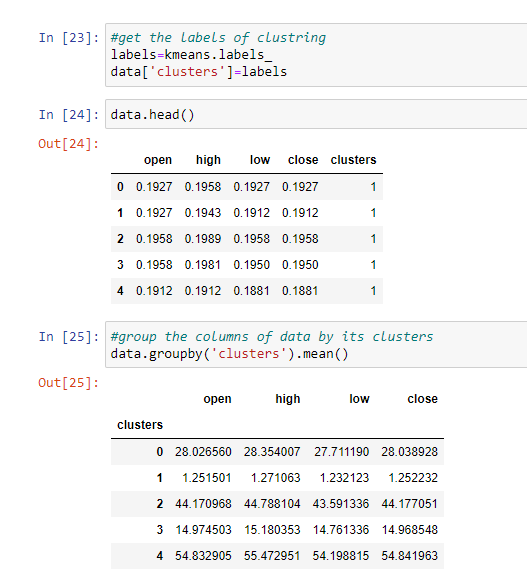


**Step 4: Visualizing the clusters**

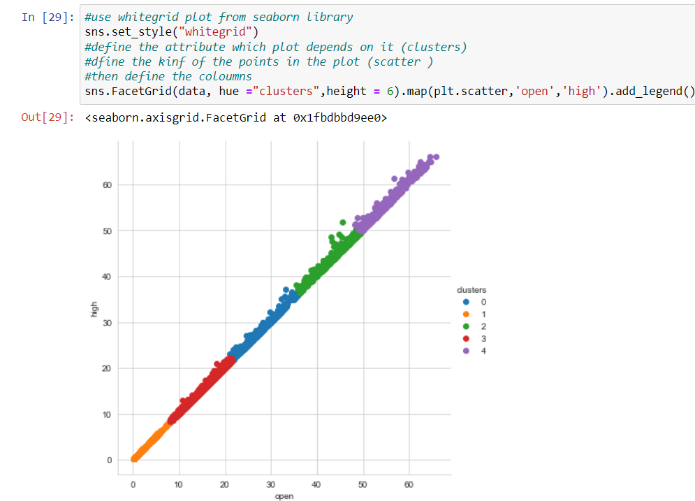


so we reached the final step of K means now we have a 5 clusters for our data.

Now we concatenate the clusters for each record in the data to make some visualization



* **Visualization of clusters with each variable**
* First one is between the open and high
* Second one is between open and low
* Third one is between the close and high
* The forth one is between the close and low

****

****

* **Mapping clusters**

We show the mapping of high and low prices in clustering

Notice that Cluster 3 biggest Cluster and Cluster 1 is the smallest



