Andrew THENEDI
18079969D

- Software Libraries:
  - NumPy
  - Matplotlib
  - Pandas
  - Scikit-learn
  - SciPy
- Data Preprocessing
  - Import dataset
    - "SixHKStockData.xlsx"
  - Drop irrelevant columns (including "trend" column)
  - Drop the records which all of the corresponding column values are empty
  - Rename the columns in the following format:
    - "stock_id: xxx"
    - "open: xxx"
    - "trend: xxx"
    - "close: xxx"
    - "high: xxx"
    - "low: xxx"
    - "volume: xxx"
    - xxx = stock ID
  - Drop the redundant columns
    - "t_date"
    - "stock_id"
  - Reposition the stock "857" columns into the last position
  - It is observed that there exists missing data from stock "857" as its IPO (initial public offering) date is later at 7 April 2000.
  - In addition, all the stocks have missing transaction volume values at a certain period sporadically.
  - To make an accurate prediction, in advance, the missing transaction volume values have to fill in using a mean value from the other transaction volume values, within the same volume column, that is filled properly.
  - Rather than replacing the missing values with a general mean value from all the filled values within the column, to be more precise, the following conditional mean algorithm to fill the missing data is as follow:
    - The first loop is applied to iterate through all 6 transaction volume columns.

- The second loop is applied to iterate through the corresponding transaction volume column.
- If the filled value is found, the third loop is not applied.
- Else, If the missing value is found, notably as 0, the third loop is applied as follow:
  - Let the following assumption be specified:
    - ‘j’ represents the variable that stores the current position.
    - ‘j_next’ represents the variable that stores the next position relative to the current position
    - ‘j_prev’ represents the variable that stores the previous position relative to the current position
    - ‘Col’ represents the variable that stores the corresponding column series.
    - ‘sum’ represents the variable that stores and updates the summation throughout the corresponding iteration.
  - If the values of the Col at index j_next, which is the column value at the next position, and Col at index j_prev, which is the column value at the previous position, are larger than zero:
    - The summation will be performed with Col at index j_next, and Col at index j_prev.
  - Else, if the values of the Col at index i+1, which is the column value at the next position, and Col at index i-1, which is the column value at the previous position, are less than zero:
    - The summation will not be performed.
  - After the aforementioned control flows finished the execution, the value of j_next will be incremented and the value of j_prev will be decremented, to move into the next position, given that both the value of j_next is less than the length of Col and the value of j_prev is larger than -1.
  - Else, if either the value of j_next is larger or equal to the length of Col or the value of j_prev is smaller or equal to -1, the third loop finishes its execution.

➢ It is observed that the stock “857” has missing data as its IPO (initial public offering) the date is later at 7 April 2000.

- ➢ Therefore, in order to cluster the time series dataset based on the movements, the prediction for the open and close column values in stock “857” has to be performed beforehand.
- ➢ Assuming that all the missing values in all of the volume columns are filled, it can be used as one of the features, in addition to the other features of open, high, close, and low column values from the stock “001”, “011”, “293”, “857”, “13”, and “23”.
- ➢ Predict ‘open: 857’ and ‘close: 857’
    - Feature Scaling
        - Min-max normalization can be performed from 0 or -1 as the minimum value to 1 as the maximum value. It is not only efficient in both reducing the scaling of all the stock’s data, but also computationally efficient as compared to other methods.
        - On the other hand, standard scaler normalization provides useful information about outliers and makes the algorithm less sensitive to them, which comes with a cost of higher computational cost than min-max normalization.
        - Thus, either method can be applied depending on the specific circumstances. In this case, min-max normalization is applied mainly due to the GPU computational power limitation on Google Collaboratory.
    - Model Training, Evaluation, and Prediction
        - Grid Search 5-Fold-Cross-Validation is applied to get the best parameters on a specific algorithm, given the input.
        - Algorithms Considered:
            - Support Vector Regression
            - Multi-layer Perceptron (Neural Network) Regression
            - Decision Tree Regression
            - Random Forest Regression
            - K-Nearest Neighbour Regression
            - Gaussian Process Regression
            - XGBoost Regression
        - Training Evaluation Result on the ’open: 857’ column:

	rmsle_score	Standard_Deviation
XGBRegressor	0.097721	0.053736
KNeighborsRegressor	0.108997	0.073404
RandomForestRegressor	0.115158	0.063305
GaussianProcessRegressor	0.131545	0.094872
SVR	0.205882	0.061490
DecisionTreeRegressor	0.205895	0.073504
MLPRegressor	0.220610	0.100150

- XGBoost Regression has the best performance for both the rmlsle and standard deviation score, which is used as the chosen model for predicting the missing values on the 'open: 857' column.

- Training Evaluation Result on the 'close: 857' column:

	rmsle_score	Standard_Deviation
GaussianProcessRegressor	0.008464	0.005146
XGBRegressor	0.030429	0.038576
RandomForestRegressor	0.081463	0.054031
KNeighborsRegressor	0.092839	0.062418
DecisionTreeRegressor	0.197886	0.112145
SVR	0.199909	0.106413
MLPRegressor	0.223727	0.097372

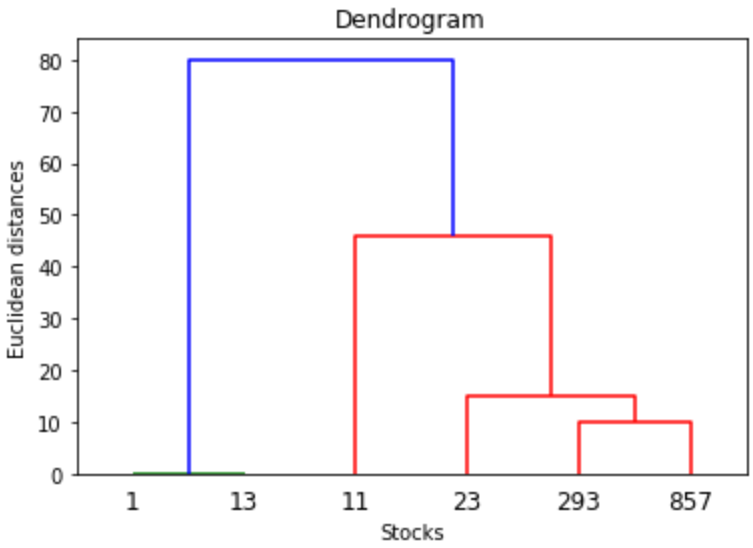
- Gaussian Process Regression has the best performance for both the rmlsle and standard deviation score, which is used as the chosen model for predicting the missing values on the 'close: 857' column.

❖ Clustering

➢ Dictionary 'd' is defined where the 'key' is the stock's name and the 'value' is the stock's movements ('movements_xxx').

- 'movements_xxx' is defined as the difference in opening and closing prices of a particular day.
- The positive movement suggests buying the stock (buy) and the negative movement suggests shorting the stock (sell).

➢ To help on deciding the k value for the K-means clustering algorithm, the hierarchical clustering algorithm is performed which display the following dendrogram:



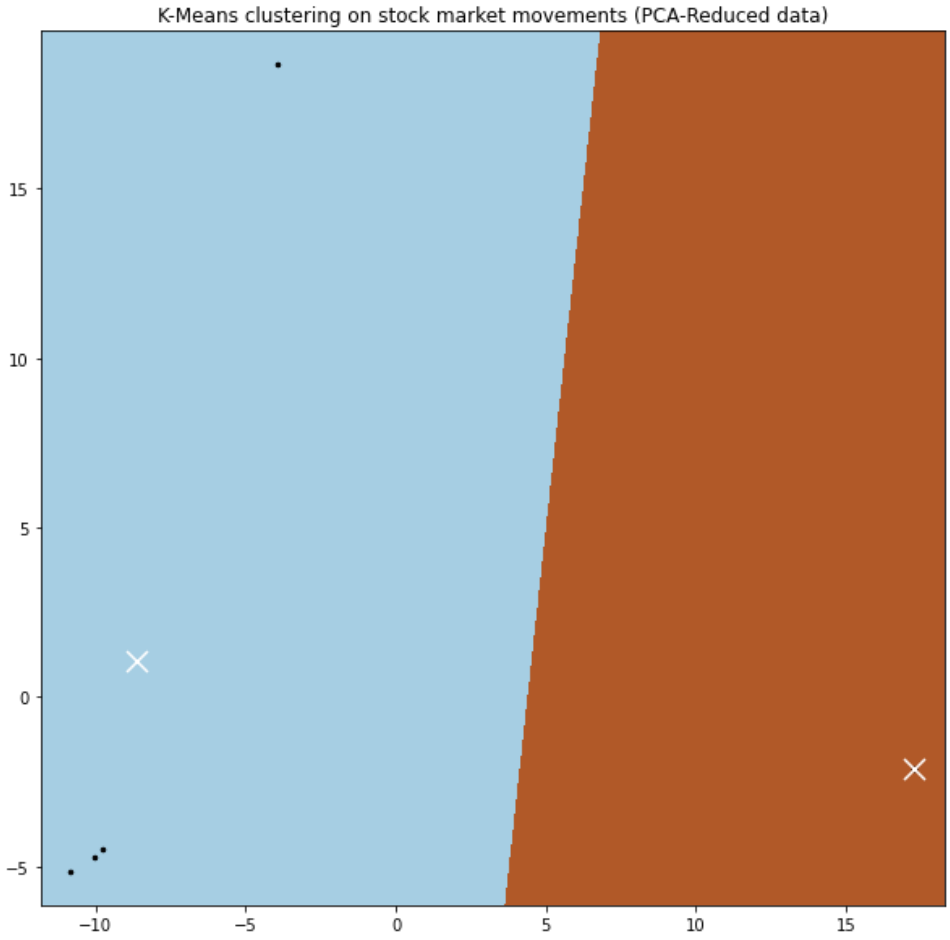
- According to the dendrogram above, it is observed that having 2 clusters is the most accurate clustering decision with a height of around 35, followed by 3 clusters with a height of around 30.

➢ Given the insight from the dendrogram above, the resulting clusters from the K-means algorithm are as follow:

	labels	stocks
1	0	11
2	0	293
4	0	23
5	0	857
0	1	1
3	1	13

- It is observed that stocks “11”, “293”, and “23” are grouped together as label 0, and stocks “1” and “13” are grouped together as label 1.

➢ The high dimensional data ‘norm_movement’, normalized movements, is reduced to 2-dimensional data with 2 features (2 days), and then K-means clustering is applied. Such that, it is possible to plot the clustered stocks on a two-dimensional graph, as follow:



- Each black point represents a stock. Each white cross represents the centroid of the respective cluster.