**COMP4433 Take-home Examination: Part I Written Report**

By Chu Long Kiu (19051822D)

1. Data Pre-processing

Text, table

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Using df.info() to check the type of every attributes.

The datatime64[ns] “tdate” is dropped due to irrelevant to inter-stock clustering. The primary key “stock\_id” is dropped due to its systematic nature.

Checking zero value for the numerical attributes:

A picture containing text

Description automatically generated

It is observed that only the attribute “volume” contains missing data. As the early stage in clustering mainly focus for close price, it is non-essential to handle such missing value in current stage.

1. Feature Engineering

Graphical user interface, text, Word

Description automatically generated with medium confidence

Feature “1DayMovement” is the difference between the close price and open price for a stock data in a day, determining the exact growth price for each day.

Feature “Movement%” is the percentage change for “1DayMovement” under the original open price, determining the percentage growth price for each day.

1. Clustering

For clustering, I have used k-means clustering (from sklearn.cluster import KMeans) for the clustering results which are divided into two cases: Movement% vs 1DayMovement and close vs 1DayMovement.

* 1. Movement% vs 1DayMovement

To determine the number of k-means clusters used, sum of squared error (sse) is represented in the following graph.

Chart, line chart

Description automatically generated

After K-means cluster > 5, the change in sse drops less significant under a stable drop rate so I decided to set the number of k-means cluster = 5.

To visualize the relation between Movement% and 1DayMovement, the following graph is called by “plt.scatter(df[‘Movement%], df[‘1DayMovement])”.

Chart, scatter chart

Description automatically generated

Using km.fit\_predict to get clusters

A picture containing chart

Description automatically generated

After that, we create ‘cluster’ attribute by the fitted clustering result by y\_predicted.

Table

Description automatically generated

Clustering result for Movement% vs 1DayMovement:

Chart, scatter chart

Description automatically generated

Five clusters are generated in the graph.

Red cluster is the negative Movement% and the low 1DayMovement.

Pink cluster is the negative Movement% but lower average 1DayMovement.

Green cluster is the medium 1DayMovement, but it is polluted by the outliers due to the sensitivity in k-means clustering.

Yellow cluster is the positive Movement% and the higher average 1DayMovement.

Orange cluster is the positive Movement% and the high 1DayMovement.

* 1. Close vs 1DayMovement

To determine the number of k-means clusters used, sum of squared error (sse) is represented in the following graph.

Chart, line chart

Description automatically generated

After K-means cluster > 3, the change in sse drops less significant under a stable drop rate so I decided to set the number of k-means cluster = 3.

To visualize the relation between close price and 1DayMovement, the following graph is called by “plt.scatter(df.close, df[‘1DayMovement])”.

Chart, scatter chart

Description automatically generated

Like case 1.



Meanwhile, I used the Euclidean distance to measure the intra-similarity for each cluster, having the high intra-similarity for a good cluster.

Text

Description automatically generated

Clustering result for close vs 1DayMovement:

Chart, scatter chart

Description automatically generated

Three clusters are generated in the graph.

Pink cluster is the close price less than 40.

Green cluster is the close price between 40 to 80.

Red cluster is the close price higher than 80, but it might be polluted by outliers in the right side.

I have tried to use data transforming to get similar clustering result for close vs 1DayMovement. The following is to obtain the nominalized attributes: close and 1DayMovement.

Text

Description automatically generated

The clustering result is same as the original one with nominalized XY scaler.

Chart, scatter chart

Description automatically generated