SYE8027_Eccentric_report

Kshitiz Kumawat | Sreeya

- **Steps we followed in code: Exploratory Data Analysis → Data Cleaning & Pre-Processing →** Created **follows** function → output CSV file

follows(bus1list,bus2list,sf_types,followed_by) function: **bus1list**: It is the list from which we take bus 1 **sf_types**: It is the list that contains seat fare types that we consider
When **followed_by**=True: function appends the following column data and when **followed_by**=False: function gives us the data frame of followed_by data

- **Steps to determine which bus follows which bus:**

1. For a bus1 in 'bus1list', for a particular bus2 in 'bus2list', for a seat fare type in sf_type list (Let's say for seat fare type 2) & for a particular service date, we have created 2 data frames df1 for bus 1 & df2 for bus2.

2. We appended a 'fare_change_df1_Seat Fare Type 2' column to df1 & 'fare_change_df1_Seat Fare Type 2' column to df2. If fare increases we append 1, if decreases then -1 else 0. For example: Here change_in_df

Seat Fare Type 2	Bus	RecordedAt	fare_change_df1_Seat Fare Type 2
12	A	2020-06-15 18:41:00	0
14	A	2020-07-09 07:32:00	1
13	A	2020-07-09 07:33:00	-1
12	A	2020-07-09 07:59:00	-1
12	A	2020-07-14 09:33:00	0
12	A	2020-07-14 13:12:00	0
15	A	2020-07-14 13:13:00	1
15	A	2020-07-14 13:37:00	0
15	A	2020-07-15 08:15:00	0
14	A	2020-07-15 10:33:00	-1

Seat Fare Type 2	Bus	RecordedAt	fare_change_df2_Seat Fare Type 2
30	B	2020-06-11 19:20:00	0
25	B	2020-07-12 10:11:00	-1
25	B	2020-07-14 06:54:00	0
25	B	2020-07-14 11:08:00	0
26	B	2020-07-14 12:26:00	1
25	B	2020-07-14 15:09:00	-1
25	B	2020-07-14 17:15:00	0
25	B	2020-07-15 02:16:00	0

1_due_to_df2 = [-1,1,1] ;
time_difference =[Timedelta('2 days 03:02:00'),Timedelta('0 days 00:47:00'),Timedelta('0 days 19:24:00')]

3. Then we made two lists **change_in_df1_due_to_df2** & **time_difference**. If -1/1 has appeared in the df2 fare change column then we see the df1 change column (immediately after the Recorded At time of df2 change) if -1/1 appears there then we append 1(reward) to change_in_df1_due_to_df2, if 1/-1 appears in df1 fare change column then we append -1(penalize). In time_difference we appended the time after a change has happened in the df1 fare change column.Then we appended the average(change_in_df1_due_to_df2) & average(time_difference) in s_dates_corrs list for all service dates.

4. Metric used for ranking of bus2 is:**[abs(Avg(Avg(change_in_df1_due_to_df2)))*len(s_dates_corrs)]** If this metric is same for 2 buses then we take bus with smaller Average(Average(time_difference)). Here first average is for a single service date then next average is for all service dates. Here we have considered: **Avg(Avg(change_in_df1_due_to_df2)) := Conditional Probability of Change in Bus 1 when change in Bus 2 has occurred (or we can say Confidence Score). len(s_dates_corrs)=>No. Of days followed**

5. After ranking & selecting the bus2 with the maximum metric score we looped this procedure to all buses in bus1list & created a data frame 'bus1_corrs'.

- **Steps to determine the 'Is followed by' column (Just the reverse way):**

1. We picked up buses one by one from the data frame, used them as bus2 & seen among all bus1s which bus has the maximum value of the above metric. Inserted in the data frame. If a clash happens then we have taken the bus with a smaller Average(Average(time_difference)). For Independent buses (with empty change_in_df1_due_to_df2), we skipped them or filled a blank space or 0.