Md Mehedi Hasan
mehedibinhafiz@gmail.com

Report on problem 1:

Task1: Segregate the data based on line id and day.

Problem Analysis

1. There are two json file of vehicle position. So I combined them.
2. Concatenated dataframe contains one column called data. each row of this column is a dictionary.
3. Each dictionary contains date as a timestamp of 13 digit represented by 'time' key, and vehicle position information as 'Responses' key.
4. Each response contains multiple lines
5. Each line contains lineId and vehiclePositions.

Now I need to segregate the vehicle positions based on this lineId and timestamps.

Algorithm

1. *Convert timestamp to date*
2. *If folder already exist:*
   Then go to 3
   Else:
   Create a folder named as the date value
   Then go to 3
3. *For each line of "Response" key:*
   For each dictionary of each line:
   a. *Store value 'lineId' key*
   b. *Make dataframe of 'vehiclePositions' key*
   c. *Export this dataframe as csv file to it's timestamp folder and named this file as lineid value.*
4. *Repeat 1 to 3 of each line of data column of concatenated dataframe. I used Lamda function for this purpose.*

Task 2: Identify the vehicle id which is missing here. Use the stop_times.txt file.

Problem Analysis:

1. pointId is the last stop crossed by a vehicle that means last point represented by pointId.
2. On the other hand, in stop_times.txt file the different stop sequence of stop id is given. From this file we can easily find last stop id by the help of last stop sequence number.
3. Form 1 and 2 I found a relationship.

4. The relationship is:
   The last stop id of each vehicle should match any of pointId. If not match any of pointId then we can say that this vehicle is missing here.
5. But another problem is there is no vehicle id is given both of dataset. But I have seen that the trip_id of each route are same. Here route is place behind strating stop_id and ending stop_id in specific arrival_time and departure_time. and trip_id in a route is same and unique for each route.
6. From 5 no point it can be said that each vehicle got unique trip_id. So based on trip_id we can assign vehicle_id is such way that each trip id got a unique vehicle_id.

Algorithm

1. Create a vehicle position dataframe by combining csv file of all lineId of all days.
2. This dataframe contains all pointId of all days
3. Access pointId of vehicle position dataframe then convert it's type to 'str' as stopId are 'str', in future we need to compare them.
4. From stop_times.txt file access all vehicle_id and stop_id based on last stop_sequence and create a dataframe.
5. This dataframe represents the all vehicle_id's last stop_id.
6. Access all stop_id and by looping check "is stop_id exist in pointId or not?"
7. If any stop_id not found in pointId that means corresponding vehicle_id is missing here.
8. If any stop_id is found in pointId that means corresponding vehicle_id is available here.