Centre for infrastructure, Sustainable Transportation and Urban Planning

Indian Institute of Science (IISc), Bengaluru Summer Internship:

Round 1

Report on Test 1

Submitted by :- Sanish kumar

Analysis of bicycle usage patterns or sharing system around the city

&

Analyze mobility patterns and develop location-based applications.

Acknowledgement

I would like to thank you for providing me with the details of a bicycle rental programme and a dataset of 6,867 bicycle trips made in a single day. The dataset includes a number of column descriptions that are useful for writing a report on the bike-sharing programme. Your contribution is much appreciated, and I believe it will enable me to gain a better understanding of the system and conduct a more thorough analysis of the data.

Analysis of mobility patterns and the development of location-based applications. Their valuable insights and expertise have helped to make this report possible. Once again, thank you for your assistance.

TABLE OF CONTENTS

Sr.No.	Outlines	Page No.
1.	Introduction	4
2.	Question 1.1	5-7
3.	Question 1.2	8-9
4.	Question 1.3	10
5.	Question 2.1	11-12
6.	Question 2.2	13
7.	Question 2.3	14-15

Introduction

Bicycle-sharing systems have gained popularity in cities worldwide due to their convenience and environmental friendliness. These systems allow users to rent bicycles from a fixed docking station and return them to any other docking station in the city, providing an affordable and efficient mode of transportation for short trips. This report focuses on analyzing a dataset of 6,867 bicycle trips over one day .

mobility patterns and developing location-based applications

Examine the possibilities offered by the dataset provided for analyzing mobility patterns and developing location-based applications. The report will provide an overview of the dataset, including its data structure and characteristics. Additionally, the report will discuss the significance of analyzing mobility patterns and how this dataset can support such analysis.

QUE-1 Methodology:

To conduct this analysis, we used a dataset of 6,867 bicycle trips over one day, collected from a bicycle-sharing system in a major city. The dataset includes information such as the start and end time of each trip, the duration of the trip, the starting and ending docking stations.

1. Write a function that removes all trips of duration 0 minutes and prints the following values on the
Maximum duration of the trip (in minutes).
• Minimum duration of the trip (in minutes).
• Total number of trips corresponding to the minimum duration.
• Percentage of total circular trips. A trip is defined as circular if it starts and ends at the same
location.
• Total runtime for the function

Begin by opening a new Python script or notebook.
Import the required libraries, including pandas and numpy. Use the 'import' keyword followed by the library name and an optional alias using 'as'.
Import the datetime module for analysis.
Use the 'read_csv' function of pandas to read in the data from the CSV file named "bike_data_new.csv". Assign this dataframe to a variable called 'df'.

Steps:

Step1.
The first step is to remove all trips with 0 minutes of duration using the 'removeall_trips_zero_duration' function that takes the input dataframe 'df' as an argument.

Step2.
The function uses pandas data manipulation techniques to remove all rows where the 'ended_at' and 'started_at' values are the same, indicating that the trip has 0 minutes of duration.

Step3.
After removing the 0 duration trips, the function calculates the maximum and minimum duration of the remaining trips in minutes. This is done by calculating the difference between the 'ended_at' and 'started_at' values in seconds, dividing it by 60 to get the duration in minutes, and then using the 'max' and 'min' functions of pandas to calculate the maximum and minimum duration.

Step4.
The function then calculates the total number of trips corresponding to the minimum duration using the 'sum' function of pandas to count the number of rows where the duration is equal to the minimum duration.

Step5.
The function also calculates the percentage of total circular trips, where a trip is defined as circular if it starts and ends at the same location. This is done by creating a boolean mask using the 'start_lat', 'end_lat', 'start_lng', and 'end_lng' columns, and then using the 'sum' function of pandas to count the number of circular trips. The percentage is then calculated by dividing the number of circular trips by the total number of trips and multiplying by 100.

Step6.
Finally, the function calculates the total runtime of the function by subtracting the start time from the end time and converting it to seconds. This runtime is printed along with the other calculated values.

The methodology involves using pandas data manipulation techniques to clean and analyze the data, and then printing and returning the calculated values. The function returns the updated dataframe with the 0 duration trips removed.

Output:-

```
Maximum duration of the trip (in minutes): 518.0000000000001
Minimum duration of the trip (in minutes): 1.0000000000000002
Total number of trips corresponding to the minimum duration: 89
Percentage of total circular trips: 2.4776425744025805%
Total runtime for the function: 1.459528 seconds.
```

2.Filter the original dataset to include only the trips starting between 06:00 AM and 06:00 PM.
Find the total number of feasible pairs of trips. Two trips, A and B,
are defined as a feasible pair if theycan be served in succession by thesame bicycle, i.e., if the end location of trip A is the same as the start location of trip B and the start time of the trip B is greater than or equal to the end time of the trip A. For example, Trip Id 1733 and 1965 are feasible. In the report, mention the total feasible pairs of trips and runtime.

__Steps:-__

First, we filtered the original dataset to only include the trips that started between 6:00 AM and 6:00 PM. This was achieved using the code: df = df[(df['started_at'].dt.hour >= 6) & (df['started_at'].dt.hour < 18)]. By doing so, we ensured that only trips that start and end during the daytime were considered.

Next, we created a new column called 'end_location', which contains a tuple of the end latitude and longitude coordinates. This was done using the code: df['end_location'] = list(zip(df['end_lat'], df['end_lng'])). This step allowed us to group trips based on their end locations.

We then created a dictionary called 'end_location_dict' that maps end locations to the trip IDs of all trips that end at that location. To do this, we looped over each trip in the dataset and added its trip ID to the corresponding end location in the dictionary.

Finally, we looped over each trip in the dataset and checked if there were any other trips that ended at the same location and started after the current trip ended. If such a trip was found, we considered it as a feasible pair and incremented a counter. The total number of feasible pairs was then returned as the output.

Print the total number of feasible pairs of trips found, along with the total runtime of the function.

Output:-

```
Total feasible pairs of trips: 52335
Total runtime for the function: 8290353.61 seconds
```

52335.

3. Filter the original dataset to include only the first 100 trips (i.e., trip id 1 to 100).

- Find the nearest node in the graph corresponding to each depot.

This methodology involves filtering a subset of data from a given pandas DataFrame by selecting only the first 100 trips using Boolean masking. Then, the start and end coordinates (latitude and longitude) of each trip are concatenated to create a new DataFrame. Duplicate rows are removed using the drop_duplicates method to get only unique depots. Finally, the number of unique depots is calculated by finding the length of the unique_depots DataFrame using the len function, which is then printed. This approach is useful for analyzing the distribution of trips or locations used in a dataset.

Output:-

```
Number of unique depots used: 169
```

QUE-2 Methodology:

- Load the dataset using pandas read_csv method.
- Clean the data by removing rows with missing values.
- Calculate the distance traveled by each user using the haversine formula.
- Aggregate the distance by user and return the total distance traveled by each user.

1. Write a function to calculate the total distance traveled by each user in the dataset.

Steps:

Step1.
Load the dataset using the read_csv method from pandas library and store it in a variable named df.

Step2.
Clean the data by removing any rows that contain missing values using the dropna method from pandas library.

Step3.
Define a function named calculate_distance_traveled that takes the cleaned DataFrame df and user ID user_id as input.

Step4.
Filter the DataFrame to select only the rows corresponding to the given user_id.

Step5.

Define a function haversine that takes the start and end points (latitude and longitude) as input and returns the distance between them using the haversine formula.

Step6

Calculate the distance traveled between each consecutive pair of location points using the haversine function and sum up the distances to get the total distance traveled by the user.

Step7.

Return the total distance traveled by the user.

Step8.

Iterate over all unique user IDs in the DataFrame, call the calculate_distance_traveled function for each user ID, and store the results in a dictionary with the user ID as key and the total distance traveled as value.

2.Write a function to extract and visualize the spatial and temporal hotspots of Bejing City.

Load the location data from the dataset into a Pandas DataFrame.
Filter the data by latitude and longitude coordinates to focus on a specific geographic region.
Group the filtered data by latitude, longitude, date, and hour to create a summary DataFrame with counts of location points at each time interval.
Create a Folium map centered on the specified geographic region.
Create a heatmap layer using the summary DataFrame.
Add the heatmap layer to the Folium map.
Add a layer control to the Folium map to allow users to toggle the heatmap layer on and off.
Return the Folium map object.

Que3.Imagine that you have access to a GPS-tracking dataset containing the trajectories of thousands of individuals over an extended period of time. The dataset includes anonymized information such as latitude, longitude, altitude, date, and time. In 500 words, describe a problem that you would like to solve using this data and what methodology you would use to solve it. You could focus on solving an issue that interests you.

GPS-tracking datasets provide a wealth of information that can be used to analyze human mobility patterns and gain insights into a wide range of topics, from urban planning to public health. One potential problem that could be addressed using such a dataset is the identification of popular outdoor recreation areas and the evaluation of their accessibility.

With the increasing interest in outdoor activities, it is becoming increasingly important to identify and promote recreational areas that are popular with the public. However, the accessibility of these areas can be a significant factor that influences their usage. For example, a park located far from residential areas may not be used as much as one that is within walking distance of homes.

To solve this problem, we could use a combination of clustering and network analysis. First, we would cluster the GPS data to identify areas that are frequently visited by individuals. We could use an algorithm such as DBSCAN, which is designed to identify clusters of arbitrary shape, to cluster the data based on spatial proximity. This would allow us to identify areas that are frequently visited and have a high density of GPS points.

Next, we would use network analysis to evaluate the accessibility of the identified recreational areas. We could create a network of roads and pathways that connect the residential areas to the recreational areas, and calculate the shortest path between each residential area and the nearest recreational area. We could then use measures such as betweenness centrality to identify the most important pathways and nodes in the network, and identify areas where additional infrastructure may be needed to improve accessibility.

To further evaluate the accessibility of the recreational areas, we could also incorporate demographic data into the analysis. By identifying the demographic characteristics of the individuals who visit the recreational areas, we could evaluate whether the areas are equally accessible to all segments of the population or whether certain groups face greater barriers to access. This information could be used to inform policy decisions related to park and recreation planning.

Overall, the combination of clustering and network analysis provides a powerful methodology for identifying popular recreational areas and evaluating their accessibility. By using this approach, we can gain insights into the factors that influence park usage and identify areas where additional infrastructure may be needed to improve access. This information can be used by policymakers to promote outdoor recreation and ensure that recreational areas are accessible to all members of the community.