IE555 Final Project

Project Overview

The CitiBike program, launched in New York City in 2013, provides a convenient, affordable, and eco-friendly mode of transportation for residents and tourists alike. With over thousands of bikes and hundreds of docking stations, CitiBike has become an integral part of the city's public transportation system. The large volume of trip data collected offers valuable insights into user behavior, bike utilization, and station demand patterns.

In this final project, you will use CitiBike trip and real-time data to explore various analytical questions, create visualizations, and apply machine learning techniques to uncover meaningful patterns and predictions. This project will allow you to showcase your skills in data processing, statistical analysis, and machine learning, while gaining a deeper understanding of urban mobility trends.

Data Description

There two datasets included:

1) Historical Trip Data from 2013-2024:
   https://s3.amazonaws.com/tripdata/index.html
   Check this website for detailed trip data description:
   https://citibikenyc.com/system-data
2) General Bikeshare Feed Specification:
   https://gbfs.citibikenyc.com/gbfs/2.3/gbfs.json
   The detailed description is provided:
   https://github.com/MobilityData/gbfs/blob/master/gbfs.md

Requirements:

- **Final report format:** a Jupyter notebook with necessary markdown notes to answer questions
- **Contribution:** for each question below, put the name initials of your group members who made contributions to the question. For a group of three students, you do not need to complete certain questions specified in each tasks.
- **Data Cleaning and Preprocessing:** Appropriate handling of missing values, outliers, and data transformation.
- **Quality of Visualizations:** Clarity, effectiveness, and appropriateness of visualizations to convey insights.
- **Analytical Depth:** Rigor and depth of analysis, including interpretation of results.
- **Presentation and Documentation:** Clear communication of findings, well-organized code, and comprehensive documentation.

Task 0: Program basic functions (25 points)

1. Write a function named `countTrips`. It will take a pandas bike trips history dataframe as its input. The function should return an integer indicating the number of trips that are contained in the dataframe.
    - NOTE: Each row corresponds to a "trip" (i.e., it has a single origin and a single destination.
2. Write a function named `avgTripDuration` that will return the average trip duration (in seconds) for a given pandas bike trips history dataframe.
3. Write a function named `maxTripDuration` that will return the maximum trip duration (in seconds) for annual members who were born on or after a given (input) year.
    - This function will take two inputs, in this order:
        - A 4-digit integer year (e.g., 1999), and
        - A pandas bike trips history dataframe.
4. Write a function named `countAllStations` that will return the number of unique stations (across both start and end stations) in a given bike trips history dataframe.
5. Write a function named `avgTripsByDayOfWeek` that will return a 7-element list containing the average number of trips taken on each day of the week. The first element of the list should represent the average number of trips taken on Sundays; the last (7th) element of the list should represent the average number of trips taken on Saturdays. The function will have one input: a given pandas bike trips history dataframe.

- HINT 1: Add a column to the dataframe that contains the day of the week corresponding to the start date of each row.
- HINT 2: The groupby() operation may be helpful.

6. Write a function named topXdepartures that will find the x stations that have the most departures, sorted in descending order. If there are ties, include those as well. For example, your function may be asked to find the 10 stations with the most departure; if there is a 3-way tie for 10th place, then your function should return 12 stations. Your function should return a pandas dataframe containing three columns:

   - rank (where rank 1 has the highest number of departures. If two or more stations have the same number of departures, then those stations should have the same rank);
   - start station id; and
   - number of departures.

   This function will take two inputs, in this order:
   - An integer indicating the top x number of departures; and
   - A pandas bike trips history dataframe.

Task 1: Citibike Trip Data Analysis (20points)

1. As of August 2024, how many active stations are currently in service (i.e., have recorded trips starting or ending at them), and how many bikes are in service in total?
2. In 2023, what is the total number of trips taken, the average trips per day, the average duration per trip, and the percentage of trips taken by annual members?
3. Based on 2023 trip records, create plots showing the total number of trips taken in: 1) each month of the year, 2) each day of the week, and 3) each hour of the day.
4. Find the top 10 most popular stations and top 10 most common bike trip routes in 2023. Show the results using bar charts.
5. Create a heatmap depicting the average usage of the top bike station identified in the last question at different times of the day.
6. Is there a relationship between trip duration and the distance between the start and end stations? Present your results visually.

For the group of three students, you DO NOT need to complete the questions below.

7. Plot the usage (total number of trips) of all stations on a map of NYC. Consider using different colors or dot sizes to represent the usage levels.

Task 2: Real-Time Data Analysis (20 points)

1. Import the station status data from station status link and the station information data from station information link. These JSON files are part of the "General Bikeshare Feed Specification" and provide real-time records of station status. Create dataframes to store the current station status and station information.
2. Merge the two dataframes using a common identifier, and visualize the number of available bikes based on the geographical areas provided in the station information data.
3. Determine the current utilization rate for each station by calculating the percentage of bikes in use relative to the total capacity of each station. Display the utilization rate on a map.
4. Using the station information data, group the stations by neighborhood or area (region_id). Create a pivot table to show the total number of available bikes, available docks, and overall capacity for each neighborhood. Identify which neighborhood has the highest total bike availability and which has the highest total dock availability.

For the group of three students, you DO NOT need to complete the questions below.

5. Calculate the total number of disabled docks and bikes across all stations. Create a summary table that includes the station information, total number of disabled docks, total number of disabled bikes, and the percentage of disabled bikes relative to the total bikes available at each station.
6. Is there correlation between the number of disabled docks and bikes?

Task 3: Trip Prediction (20 points)

The number of bike-sharing trips is a classic example of a time series, as it reflects the demand levels at various stations over time. Leveraging historical trip data allows us to train models and forecast future demand.

Import the Trip data from 2013-2023

1. Load the trip data from 2013 to 2023
2. Select an appropriate machine learning method and train your model using data from 2013 to 2022. Test your model's performance by evaluating its prediction accuracy using the 2023 data. In addition to presenting numerical metrics such as MAE or RMSE, plot the predicted values against the actual values for visual comparison.
3. Retrain and test your model by excluding the data collected during the COVID-19 period. Report the model's prediction performance again and compare it to the results obtained using the complete dataset.

For the group of three students, you DO NOT need to complete the questions below.

4. Examine the daily demand change pattern using the total number of trips recorded at each hour of the day (Task 1, Question 3). You may observe different patterns across stations.
    - Group the stations into distinct clusters based on the observed patterns using time series clustering. Choose the number of clusters according to the variation in patterns you identified. You can refer to the tslearn documentation for guidance on conducting time series clustering.
    - Plot the clustering results on a map, using different colors to represent stations belonging to different clusters.

Task 4: Optimization for Bike Rebalancing (15 points)

Bike rebalancing refers to the process of redistributing bicycles across the bike-sharing system to ensure an even availability of bikes and docking stations throughout a city. This is crucial for maintaining user convenience and satisfaction, as it addresses imbalances caused by high demand in certain areas and low usage in others.

The rebalancing process typically involves monitoring bike usage data in real-time, identifying stations with high demand that are running low on bikes, and those with excess bikes that need to be moved. Operators may use trucks or electric vehicles to transport bikes from oversupplied stations to those in need, optimizing the system's overall efficiency.

Question: Pick 20 stations of Citibike, a delivery person needs to start from Station A (any station you pick) and visit all other stations exactly once before returning to Station A. What is the shortest possible route the delivery person can take to complete this journey, minimizing the total distance traveled? Use Python to solve this Traveling Salesman Problem (TSP). Note: The distance between the stations can be calculated based on the longitude and latitude given.