ITCS-6100 Big Data for Computational Advantage

Group -18

Project Deliverable - 3

Team Members

Manasa Avula - 801307493

Nikhita Sai Boyidapu - 801327682

Srikar Chamarthy - 801317299

Rachana Gullipalli - 801311637

Aravind Pabbisetty - 801274519

7) Analytics, Machine Learning

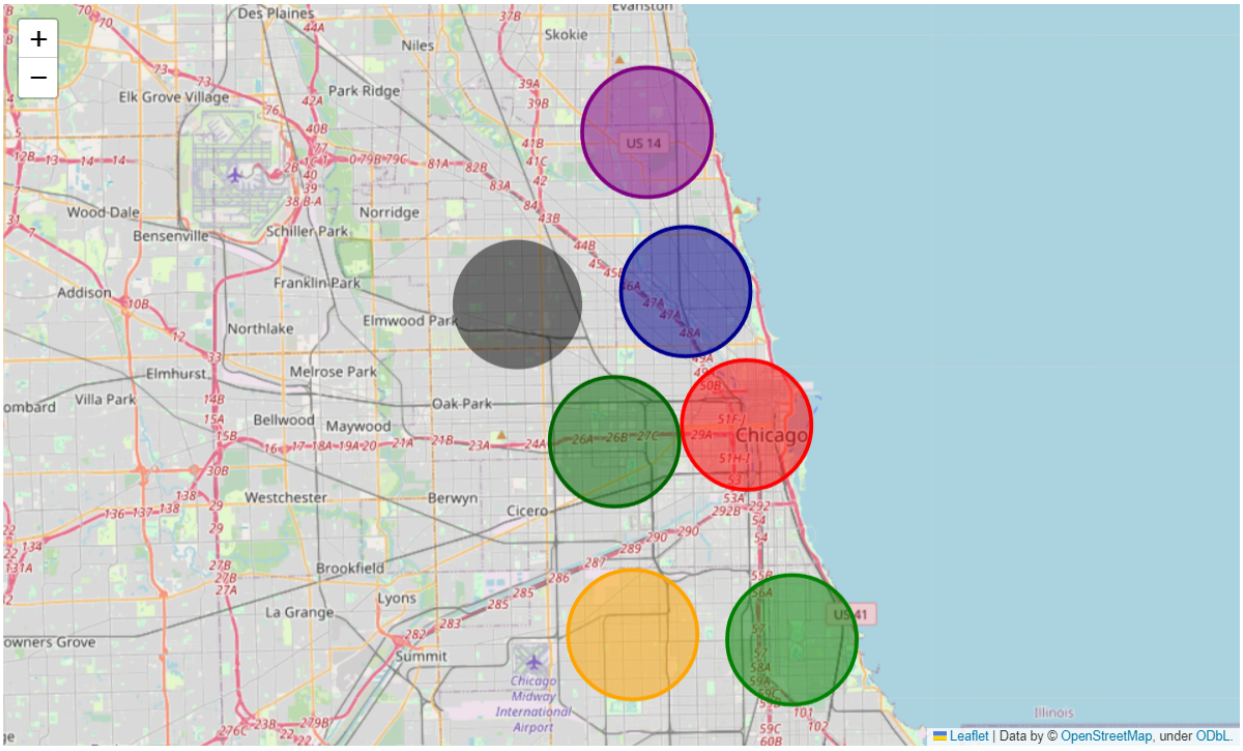
We have used two ml models random forest classifier and k means clustering to get the insights from the dataset. By using random forest classifier we were able to predict which type of bikes will be preferred in the next few years.

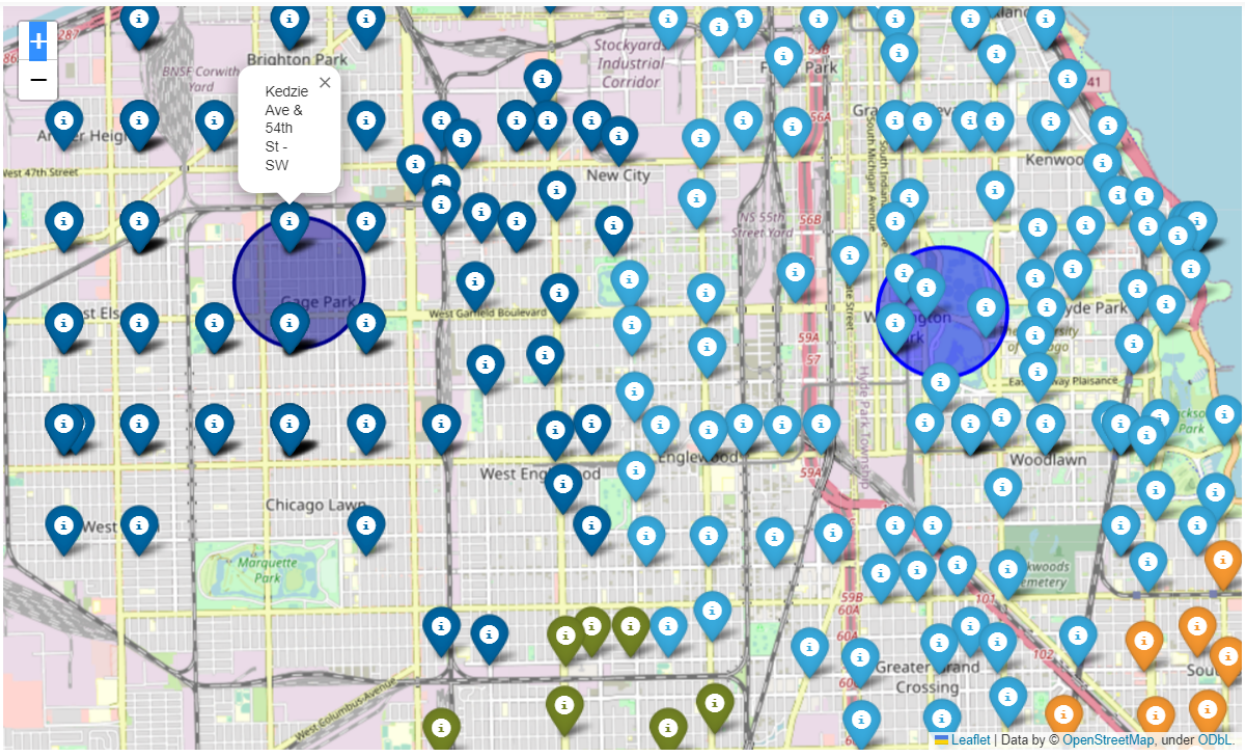
```
df['started_at_date'] = pd.to_datetime(df['started_at_date'])
df['month'] = df['started_at_date'].dt.month
df['year'] = df['started_at_date'].dt.year
df['rideable_type'] = df['rideable_type'].replace({'classic_bike': 0, 'electric_bike': 1,'docked_bike':2})

X = df[['month', 'year']]
y = df['rideable_type']
rfc = RandomForestClassifier()
rfc.fit(X, y)
future_months = pd.date_range(start='2022-10-01', end='2027-09-01', freq='MS')
future_data = pd.DataFrame({'month': future_months.month, 'year': future_months.year})

predictions = rfc.predict(future_data)
predicted_bike_type = 'electric' if sum(predictions) > len(predictions) / 2 else 'classic'

print(f"The preferred bike type for the next 5 years is {predicted_bike_type}.")
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

rfc = RandomForestClassifier()
rfc.fit(X_train, y_train)

y_pred = rfc.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"The accuracy of the model is {accuracy}")
```

```
The preferred bike type for the next 5 years is classic.
The accuracy of the model is 0.5828590909090909
```

From the above image we can see that the preferred bike is a classic bike. We have

also predicted the most demandable location(hotspot) using k-means clustering model and we were able to identify 7 hotspots around the city. The below image shows those locations.

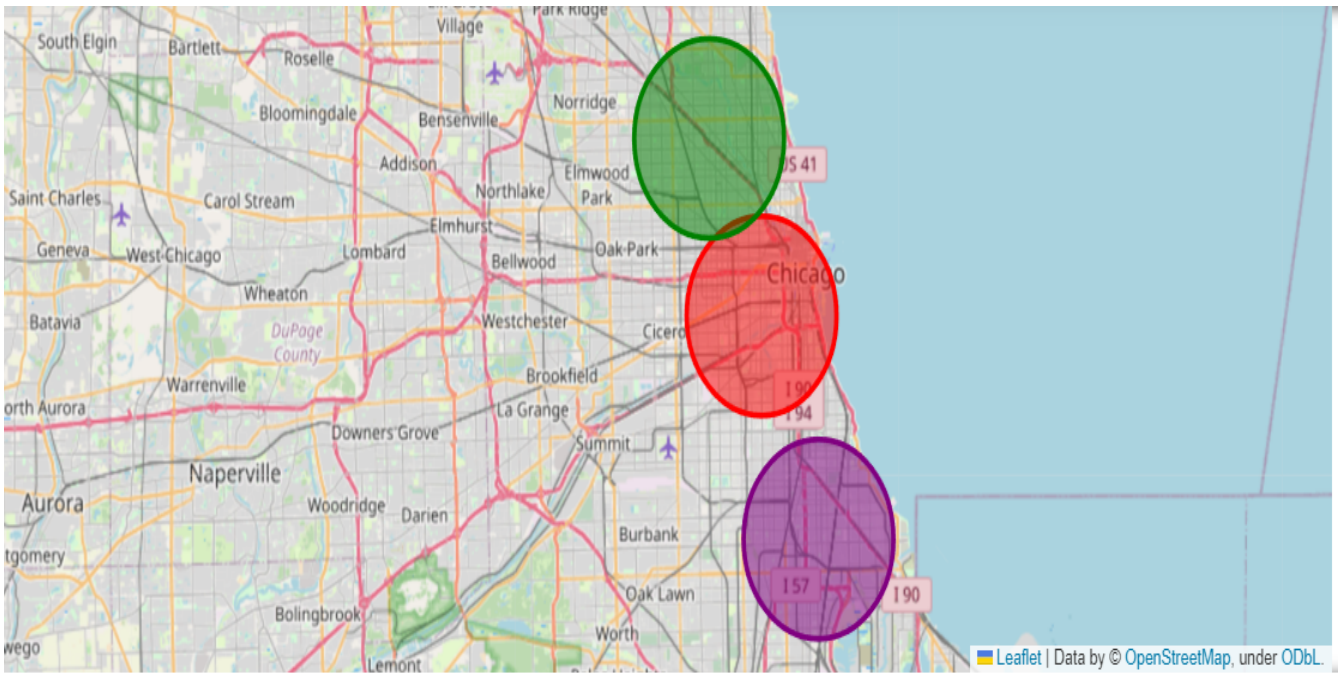




8) Evaluation and Optimization

We have evaluated the K-means Clustering model performance by checking whether the clusters are loosely or tightly coupled. In case of clustering models inertias and silhouette_scores are the parameters that evaluate the model performance. Using these scores we can find the optimal number of clusters required to do clustering. So we have optimized the model based on the optimal number of clusters and we can see a change in the number of clusters. The clusters are clustered in such a way that the points that are mostly similar and have too much variation with other clusters are placed in the same cluster.

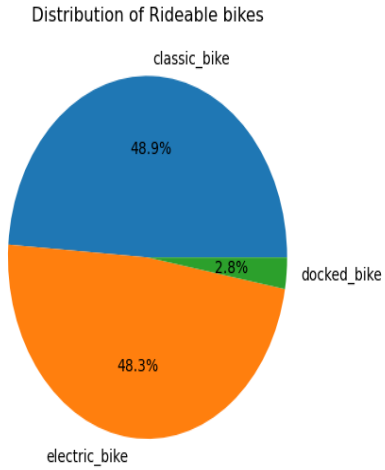
Most demandable locations/ hotspots after model Optimization



9) Results

- **Which bike type is most used among the classic ,docked and electric bikes?**

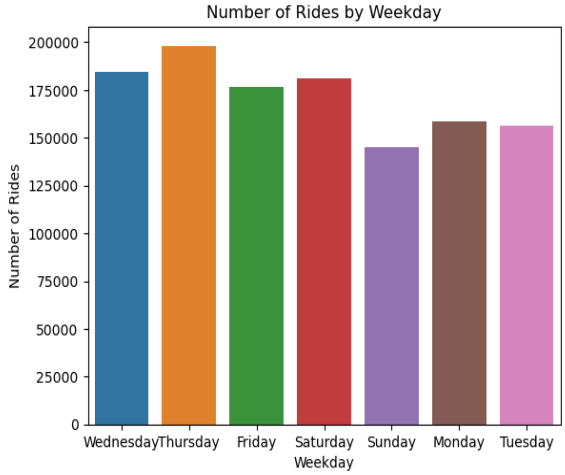
```
In [79]: ride_counts = datafile['rideable_type'].value_counts()
         plt.pie(ride_counts, labels=ride_counts.index, autopct='%1.1f%%')
         plt.title("Distribution of Rideable bikes")
         plt.show()
```



From the above graph we can say that classic bikes are most used.

- **Which weekday is preferable for people to ride the bike?**

```
In [67]: sns.countplot(x='weekday', data=datafile)
         plt.title('Number of Rides by Weekday')
         plt.xlabel('Weekday')
         plt.ylabel('Number of Rides')
         plt.show()
```



From the above graph we can see that thursday is more preferable for people to ride the bike

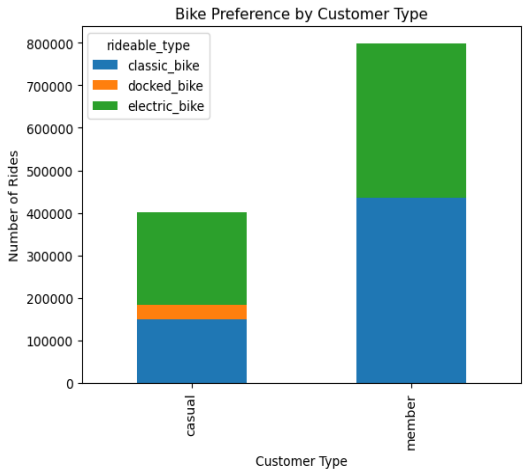
- **What is the most used end station for returning the bike?**

```
In [64]: from wordcloud import WordCloud as wd
         end_station_data = datafile["end_station_name"].value_counts()
         wordcloud = wd(width=300,height=100,background_color="white").generate_from_frequencies(end_station_data)
         plt.figure(figsize=(8,8))
         plt.imshow(wordcloud)
         plt.axis("off")

Out[64]: (-0.5, 299.5, 99.5, -0.5)
```

From the above visualization we can see that Michigan Ave & Oak St is the most used end station to return the bike

- **What types of bikes are preferred by different customer types?**

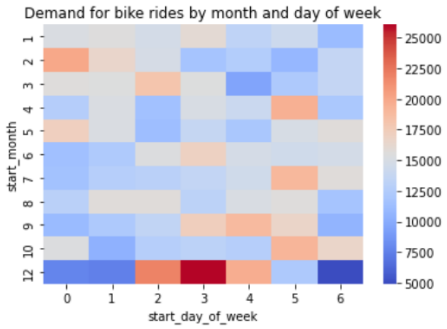
```
In [65]: # Group the data by member/casual and bike type, then count the number of occurrences
         bike_preference = datafile.groupby(['member_casual', 'rideable_type']).size().unstack()

         bike_preference.plot(kind='bar', stacked=True)
         plt.xlabel('Customer Type')
         plt.ylabel('Number of Rides')
         plt.title('Bike Preference by Customer Type')
         plt.show()
```



From the above visualization we can see that members use classic bikes more and casual people use electric bikes more.

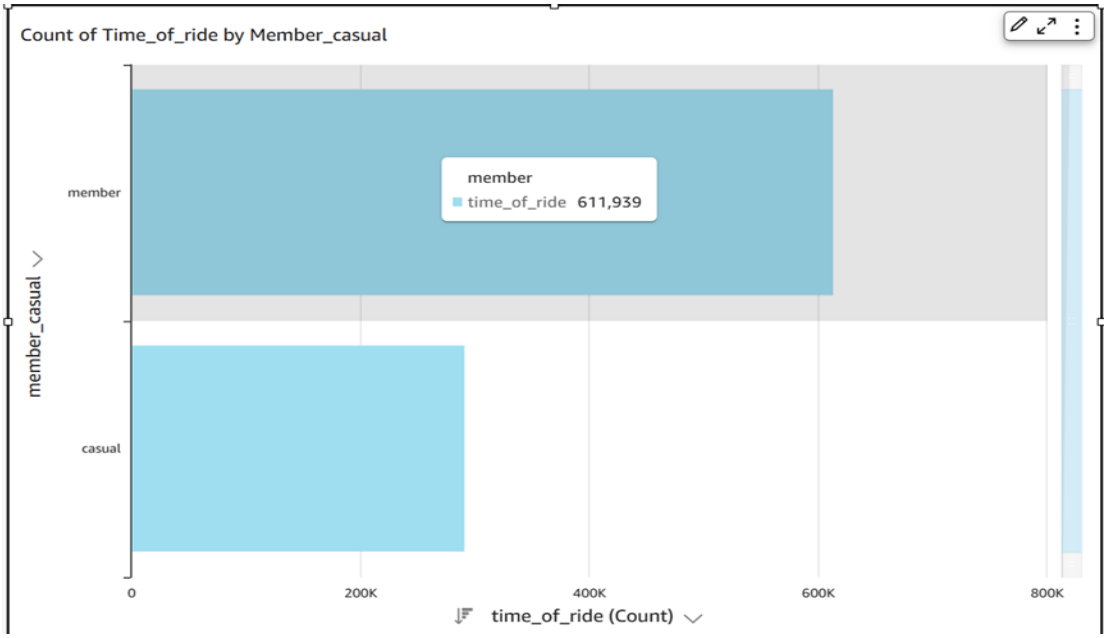
- **How does the demand for bike rides vary across different months and days of the week based on start date and time?**

```
In [41]: df = datafile.copy()
         df['started_at_date'] = pd.to_datetime(df['started_at_date'])

         df['start_month'] = df['started_at_date'].dt.month
         df['start_day_of_week'] = df['started_at_date'].dt.dayofweek

         heatmap_data = df.pivot_table(index='start_month', columns='start_day_of_week', values='ride_id', aggfunc='count')
         sns.heatmap(heatmap_data, cmap='coolwarm')
         plt.title('Demand for bike rides by month and day of week')
         plt.show()
```



This visualization shows the demand for bike rides vary across different months and days of the week. X-axis represents the day of the week and Y-axis represents the month. Most demand is on the Thursdays of december.

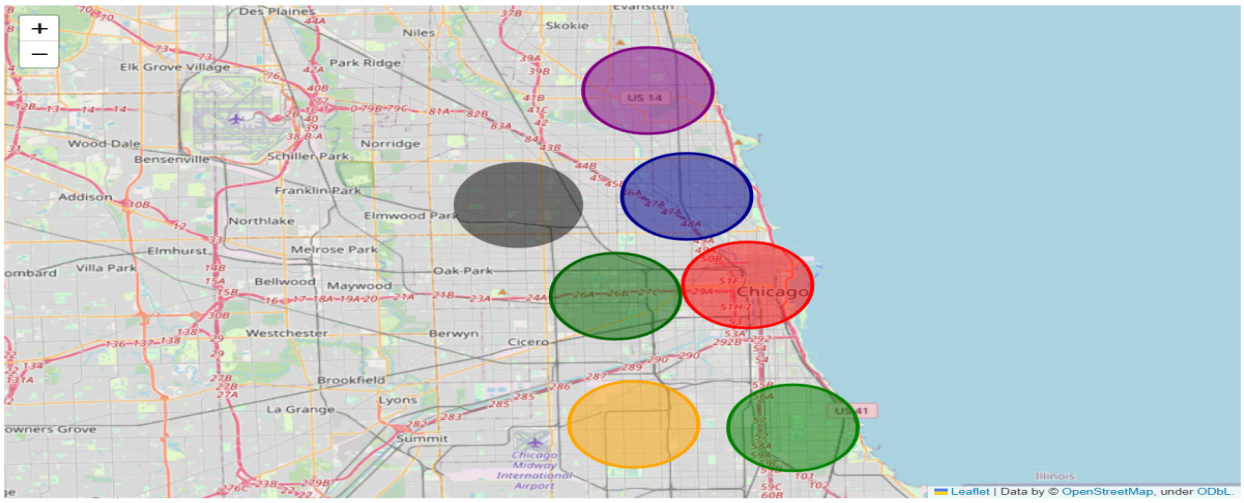
- **Is there a difference in the duration of rides between members and casual riders?**



We can say that there is a difference in the duration of rides between members and casual riders. Members take rides for a longer duration of time.

- **What will be the most demandable locations for the next 3 years?**

  In the below visualization we can see the top most hotspot areas for the bike pickups. With this bike companies can increase their number of stations and bikes in those hotspots to meet the demand and increase their revenue.



- **What will be the preferred bike in the next few years?**

```
df['started_at_date'] = pd.to_datetime(df['started_at_date'])
df['month'] = df['started_at_date'].dt.month
df['year'] = df['started_at_date'].dt.year
df['rideable_type'] = df['rideable_type'].replace({'classic_bike': 0, 'electric_bike': 1,'docked_bike':2})

X = df[['month', 'year']]
y = df['rideable_type']
rfc = RandomForestClassifier()
rfc.fit(X, y)
future_months = pd.date_range(start='2022-10-01', end='2027-09-01', freq='MS')
future_data = pd.DataFrame({'month': future_months.month, 'year': future_months.year})

predictions = rfc.predict(future_data)
predicted_bike_type = 'electric' if sum(predictions) > len(predictions) / 2 else 'classic'

print(f"The preferred bike type for the next 5 years is {predicted_bike_type}.")
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

rfc = RandomForestClassifier()
rfc.fit(X_train, y_train)

y_pred = rfc.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"The accuracy of the model is {accuracy}")
```

```
The preferred bike type for the next 5 years is classic.
The accuracy of the model is 0.5828590909090909
```

Using the model we were able to predict that for the next 5 years the preferred bike type is classic bike

10) Future Work, Comments - students may want to consider the following questions

1. **What was unique about the data? Did you have to deal with imbalance? What data cleaning did you do? Outlier treatment? Imputation?**
   - The dataset consists of unique attributes that describe the information of the trip such as trip duration, type of bike used, customer type, start and end stations. These features are helpful to analyze, derive various patterns.
   - These patterns are helpful for the Cyclistic company to forecast the future demand and ensure that they meet the demand by increasing the number of bikes or stations in the busiest localities.
   - We have performed data cleaning by removing the null columns and removing duplicate entries.
   - We have detected outliers entries which are stations far outside and removed them.
   - Next we found some null values in different attributes and imputed them by finding the mean for numerical attributes and finding the mode for categorical attributes.

2. **Did you create any new additional features / variables?**

   We have created some new features and added them to the dataset. We have encoded the Categorical variables such as member_type and bike type to numerical variables. Also we have added a new feature trip distance which is done by computing the distance between latitudes and longitudes of start and end point.

3. **What was the process you used for evaluation? What was the best result?**

   For the classification of which type of bike is most preferred we have used Random Forest Classifier. We have split the dataset into training and test dataset. With the training dataset we have trained the model and using the test dataset we were able to calculate the accuracy and able to predict which bike is preferred in the next 5 years. The model we have chosen is random forest classifier because the accuracy using this model is more compared to other models like the SVM classifier, Decision tree classifier. We have used Kmeans clustering to find the most demandable location. Using that we were able to create clusters where most rides are started and identified the hotspots in the city. We have evaluated it based on the clustering score and optimized the number of clusters.

4. **What were the problems you faced? How did you solve them?**

   The main challenge we faced was extracting the features that can be used to get the insights. We also had to clean a lot of data, remove null columns, duplicate rows, impute the null values and also detect, remove the outliers. We faced difficulties in choosing the right model for finding the predictions. We worked on different classification and clustering models, evaluated model accuracy and chose the best model to derive our predictions.

5. **What future work would you like to do?**

In the future we would like to optimize the model, improve its performance, accuracy and derive many other useful insights and predictions that will be useful for the bike sharing organization while making decisions. So that they can provide much better services to its customers and stand ahead of their competitors.

6. **Instructions for individuals that may want to use your work?**
    - **Requirements**: Amazon S3, Amazon Sagemaker, Python, Jupyter Notebook
    - **Packages**: Numpy, Seaborn, Pandas, Matplotlib
    - **Steps to run:**
        - The first step is to download the Cyclistic Dataset from Kaggle or from the link provided below.
        - After downloading the dataset, create a S3 bucket to store the data in AWS.
        - Upload the csv file that consists of Cyclistic data into the S3 bucket.
        - Next with the help of Jupyter notebook in AWS Sagemaker, perform data understanding and cleaning, preparation activities such as removing duplicate values and imputation null values and adding new features.
        - Finally with the help of AWS Quicksight create useful visualizations and add them to the dashboard.

Dataset link:

https://www.kaggle.com/datasets/jasfre/gcc-cyclistic-case-study-present-report-prompt

You can also copy the merged csv files of each month from the this link:

https://drive.google.com/file/d/1I-RdrIHhjEAKdMJM_P_zSjIVKHISg-2d/view?usp=share_link

GitHub repository URL:

https://github.com/aravindpabbisetty/BigDataGroup18