# Google Data Analytics Professional Certificate Capstone Project: Bike Sharing data

# First we import all the required libraries:

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from datetime import datetime

import janitor

import numpy as np

# Importing the data sets:

data\_sep2020 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\Sep2020-tripdata.csv")

data\_oct2020 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\Oct2020-tripdata.csv")

data\_nov2020 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\Nov2020-tripdata.csv")

data\_dec2020 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\Dec2020-tripdata.csv")

data\_jan2021 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\Jan2021-tripdata.csv")

data\_feb2021 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\Feb2021-tripdata.csv")

data\_mar2021 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\Mar2021-tripdata.csv")

data\_apr2021 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\Apr2021-tripdata.csv")

data\_may2021 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\May2021-tripdata.csv")

data\_june2021 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\June2021-tripdata.csv")

data\_july2021 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\July2021-tripdata.csv")

data\_aug2021 = pd.read\_csv("D:\\google project data\\dataset\_csv\_format\\Aug2021-tripdata.csv")

# The column names of all the dataset is checked for confirming consistency:

print(data\_sep2020.columns)

print(data\_oct2020.columns)

print(data\_nov2020.columns)

print(data\_dec2020.columns)

print(data\_jan2021.columns)

print(data\_feb2021.columns)

print(data\_mar2021.columns)

print(data\_apr2021.columns)

print(data\_may2021.columns)

print(data\_june2021.columns)

print(data\_july2021.columns)

print(data\_aug2021.columns)

import pandas as pd

# To check the structure of all the columns mentioned in the datasets:

for dataset in [data\_sep2020, data\_oct2020, data\_nov2020, data\_dec2020, data\_jan2021,

data\_feb2021, data\_mar2021, data\_apr2021, data\_may2021, data\_june2021,

data\_july2021, data\_aug2021]:

print(dataset.info())

# Convert 'start\_station\_id' and 'end\_station\_id' to string type for all datasets

datasets = [data\_sep2020, data\_oct2020, data\_nov2020, data\_dec2020, data\_jan2021,

data\_feb2021, data\_mar2021, data\_apr2021, data\_may2021, data\_june2021,

data\_july2021, data\_aug2021]

for data in datasets:

data['start\_station\_id'] = data['start\_station\_id'].astype(str)

data['end\_station\_id'] = data['end\_station\_id'].astype(str)

# Now combining all the data for the last 12 months into a single dataframe named "trip\_data":

trip\_data = pd.concat(datasets, ignore\_index=True)

import pandas as pd

import numpy as np

from datetime import datetime

# To check the value in the rideable\_type column:

print(trip\_data['rideable\_type'])

# Now we need to separately create columns for months, year, day and day of the week:

trip\_data['months'] = pd.to\_datetime(trip\_data['started\_at']).dt.strftime('%m')

trip\_data['year'] = pd.to\_datetime(trip\_data['started\_at']).dt.strftime('%Y')

trip\_data['day'] = pd.to\_datetime(trip\_data['started\_at']).dt.strftime('%d')

trip\_data['day\_of\_the\_week'] = pd.to\_datetime(trip\_data['started\_at']).dt.strftime('%a')

# Converting time into the format of hours and minutes:

trip\_data['time'] = pd.to\_datetime(trip\_data['started\_at']).dt.strftime('%H:%M')

trip\_data['time'] = pd.to\_datetime(trip\_data['time'], format='%H:%M')

# Calculating the length of the ride(in mins) taken by the user:

trip\_data['ride\_length'] = (trip\_data['ended\_at'] - trip\_data['started\_at']).dt.total\_seconds() / 60

# To convert ride\_length column from "factor" to "numeric" so that we can use it

trip\_data['ride\_length'] = pd.to\_numeric(trip\_data['ride\_length'])

print(pd.api.types.is\_numeric\_dtype(trip\_data['ride\_length']))

# Now we need to check the minimum value of ride\_length and need to see whether it has any negative values or not :

print(trip\_data['ride\_length'].min())

# Checking for trip lengths less than 0

print(len(trip\_data[trip\_data['ride\_length'] < 0]))

# Checking for testrides that were made by company for quality checks

print(len(trip\_data[trip\_data['start\_station\_name'].str.contains('TEST', case=False, na=False)]))

# We create a new dataframe named "trip\_data\_pos" which will only have data containing positive ride\_length values and are not observations of tests undertaken by the company

trip\_data\_pos = trip\_data[trip\_data['ride\_length'] >= 0]

# Removing the test rides

trip\_data\_pos = trip\_data\_pos[~trip\_data\_pos['start\_station\_name'].str.contains('TEST|test', case=False, na=False)]

# Checking the dataframe:

print(trip\_data\_pos.info())

# Checking count of distinct values

print(trip\_data\_pos['member\_casual'].value\_counts())

# Aggregating total trip duration by customer type

print(trip\_data\_pos.groupby('member\_casual')['ride\_length'].sum().reset\_index().rename(columns={'ride\_length': 'total\_ride\_length'}))

# To find the mean, median, maximum and minimum values of ride length for casual and regular members:

print(trip\_data\_pos.groupby('member\_casual')['ride\_length'].agg(['mean', 'median', 'max', 'min']))

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Checking the dataframe trip\_data\_pos:

print(trip\_data\_pos.info())

# Creating a summary table for number of rides with respect to customer types and day of the week:

trip\_data\_pos\_summary = trip\_data\_pos.groupby(['member\_casual', 'day\_of\_the\_week']).agg(

no\_of\_rides=('ride\_length', 'count'),

average\_duration=('ride\_length', 'mean')

).reset\_index().sort\_values(['member\_casual', 'no\_of\_rides'], ascending=[True, False])

# To check the new summary dataframe:

print(trip\_data\_pos\_summary.info())

# VISUALIZATIONS:

# Bar graph representing the relation between customer\_type and day of the week:

summary = trip\_data\_pos.groupby(['member\_casual', 'day\_of\_the\_week']).size().reset\_index(name='no\_of\_rides')

plt.figure(figsize=(12, 6))

sns.barplot(x='day\_of\_the\_week', y='no\_of\_rides', hue='member\_casual', data=summary, dodge=True)

plt.title('Total trips by customer type Vs. Day of the week')

plt.ylabel('Number of rides')

plt.ticklabel\_format(style='plain', axis='y')

plt.xticks(rotation=45)

plt.show()

# Average number of trips by customer types and month:

monthly\_summary = trip\_data\_pos.groupby(['member\_casual', 'months']).agg(

no\_of\_rides=('ride\_length', 'count'),

average\_duration\_mins=('ride\_length', 'mean')

).reset\_index().sort\_values(['member\_casual', 'no\_of\_rides'], ascending=[True, False])

print(monthly\_summary)

# Visualization for total trips by customer type vs. Months

plt.figure(figsize=(12, 6))

sns.barplot(x='months', y='no\_of\_rides', hue='member\_casual', data=monthly\_summary, dodge=True)

plt.title('Total trips by customer type Vs. Months')

plt.ylabel('Number of rides')

plt.ticklabel\_format(style='plain', axis='y')

plt.xticks(rotation=30)

plt.show()

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Average trip duration by customer type on each day of the week:

(trip\_data\_pos.groupby(['member\_casual', 'day\_of\_the\_week'])

.agg({'ride\_length': 'mean'})

.reset\_index()

.pipe((sns.catplot, 'data'), x='day\_of\_the\_week', y='ride\_length', hue='member\_casual', kind='bar', dodge=True)

.set(title='Average trip duration by customer type Vs. Day of the week'))

plt.show()

# Average trip duration by customer type in each month:

(trip\_data\_pos.groupby(['member\_casual', 'months'])

.agg({'ride\_length': 'mean'})

.reset\_index()

.pipe((sns.catplot, 'data'), x='months', y='ride\_length', hue='member\_casual', kind='bar', dodge=True)

.set(title='Average trip duration by customer type Vs. Month'))

plt.xticks(rotation=30)

plt.show()

# Demand of bikes over a period of 24 hours (1 day):

(trip\_data\_pos.groupby(['member\_casual', 'time'])

.size()

.reset\_index(name='no\_of\_trips')

.pipe((sns.lineplot, 'data'), x='time', y='no\_of\_trips', hue='member\_casual')

.set(title='Demand over 24 hours of a day', xlabel='Time of the day'))

plt.xticks(rotation=90)

plt.show()

# Type of rides VS no of trips with respect to the customer types:

(trip\_data\_pos.groupby(['rideable\_type', 'member\_casual'])

.size()

.reset\_index(name='no\_of\_trips')

.pipe((sns.barplot, 'data'), x='rideable\_type', y='no\_of\_trips', hue='member\_casual')

.set(title='Ride type Vs. Number of trips'))

plt.show()

# Now at the end, we create a csv file consisting of the clean data which can be used for further analysis and visualizations:

clean\_data = (trip\_data\_pos.groupby(['member\_casual', 'day\_of\_the\_week'])

.agg({'ride\_length': 'mean'})

.reset\_index())

clean\_data.to\_csv("Clean Data.csv", index=False)