Problem Statement

Driver recruitment and retention are critical challenges for Ola, with high churn rates making it easy for drivers to switch to competitors like Uber. This high turnover not only affects organizational morale but also increases costs, as acquiring new drivers is more expensive than retaining existing ones. To mitigate this, Ola has expanded its recruitment pool to include individuals without cars, though this approach is costly.

As a data Analyst in Ola's Analytics Department, your task is to analyze monthly data from 2019 and 2020, focusing on driver attributes, and predict whether a driver is likely to leave the company. The goal is to develop insights and predictive models that can help reduce driver attrition and optimize retention strategies.

Import Required Library

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Import data in dataset

```
data =  pd.read_csv('/content/ola_driver.csv')
```

 # Check Dataset and Apply some Basic Function

```
pd.set_option("display.max_rows",50)
pd.set_option("display.max_columns",50)
# These Function use to show at a time show 50 rows and 50 columns
data

data.head()
```

```
/usr/local/lib/python3.10/dist-packages/google/colab/
_dataframe_summarizer.py:88: UserWarning: Could not infer format, so
each element will be parsed individually, falling back to `dateutil`.
To ensure parsing is consistent and as-expected, please specify a
format.
  cast_date_col = pd.to_datetime(column, errors="coerce")
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 19104,\n
\"fields\": [\n    {\n      \"column\": \"Unnamed: 0\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
5514,\n        \"min\": 0,\n        \"max\": 19103,\n
```

```
\"num_unique_values\": 19104,\n        \"samples\": [\n
18299,\n          9376,\n          4518\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"MMM-YY\",\n      \"properties\":
{\n        \"dtype\": \"object\",\n        \"num_unique_values\": 24,\
n        \"samples\": [\n          \"03/01/20\",\n
\"10/01/19\",\n          \"01/01/19\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Driver_ID\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
810,\n        \"min\": 1,\n        \"max\": 2788,\n
\"num_unique_values\": 2381,\n        \"samples\": [\n          1663,\
n          1264,\n          1618\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 6.2579116861907345,\n
\"min\": 21.0,\n        \"max\": 58.0,\n        \"num_unique_values\":
36,\n        \"samples\": [\n          58.0,\n          41.0,\n
24.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.4933670037660394,\n        \"min\": 0.0,\n        \"max\":
1.0,\n        \"num_unique_values\": 2,\n        \"samples\": [\n
1.0,\n          0.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"City\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 29,\n        \"samples\": [\n
\"C22\",\n          \"C5\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Education_Level\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 2,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          2,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Income\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 30914,\n        \"min\": 10747,\n
\"max\": 188418,\n        \"num_unique_values\": 2383,\n
\"samples\": [\n          44273,\n          35370\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Dateofjoining\",\n
\"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 869,\n        \"samples\": [\n
\"14/09/19\",\n          \"01/06/18\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"LastWorkingDate\",\n
\"properties\": {\n        \"dtype\": \"date\",\n        \"min\":
\"2018-12-31 00:00:00\",\n        \"max\": \"2020-12-28 00:00:00\",\n
\"num_unique_values\": 493,\n        \"samples\": [\n
\"05/03/20\",\n          \"10/01/19\"\n        ],\n
```

```
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Joining Designation\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 1,\n        \"max\": 5,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          2,\n
5\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Grade\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1,\n        \"min\": 1,\n        \"max\": 5,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          2,\n
5\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Total Business Value\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1128312,\n        \"min\": -6000000,\n
\"max\": 33747720,\n        \"num_unique_values\": 10181,\n
\"samples\": [\n          431090,\n          720180\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Quarterly Rating\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 1,\n        \"max\": 4,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          1,\n
3\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"data"}
```


#1. Data Structure and Overview:

Question- What is the structure of the dataset (number of rows and columns)?

```
data.shape
```

```
(19104, 14)
```

Question- What are the data types of each column?

```
data.dtypes
```

```
Unnamed: 0                 int64
MMM-YY                    object
Driver_ID                  int64
Age                      float64
Gender                   float64
City                      object
Education_Level            int64
Income                     int64
Dateofjoining             object
LastWorkingDate           object
Joining Designation        int64
Grade                      int64
```

```
Total Business Value      int64
Quarterly Rating          int64
dtype: object
```

Question- Are there any missing values in the dataset? If so, which columns are affected?

```
data.isna().sum()
```

```
Unnamed: 0                    0
MMM-YY                        0
Driver_ID                     0
Age                          61
Gender                       52
City                          0
Education_Level               0
Income                        0
Dateofjoining                 0
LastWorkingDate           17488
Joining Designation           0
Grade                         0
Total Business Value          0
Quarterly Rating              0
dtype: int64
```

2. Descriptive Statistics:

Question- What are the basic statistics (mean, median, standard deviation) for numerical features like Age, Income, Total Business Value, and Quarterly Rating?

```
data[['Age','Income','Total Business Value','Quarterly
Rating']].describe([])
```


```
{"summary":"{\n  \"name\": \"data[['Age','Income','Total Business
Value','Quarterly Rating']]\",\n  \"rows\": 6,\n  \"fields\": [\n
{\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 7761.72301581364,\n
\"min\": 6.2579116861907345,\n        \"max\": 19043.0,\n
\"num_unique_values\": 6,\n        \"samples\": [\n          19043.0,\
n          34.668434595389385,\n          58.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Income\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 65468.0505610525,\
n        \"min\": 10747.0,\n        \"max\": 188418.0,\n
\"num_unique_values\": 6,\n        \"samples\": [\n          19104.0,\
n          65652.02512562813,\n          188418.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Total Business Value\",\n
```

\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
14348449.435504831,\n   \"min\": -6000000.0,\n   \"max\":
33747720.0,\n   \"num_unique_values\": 6,\n   \"samples\":
[\n   19104.0,\n   571662.074958124,\n
33747720.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Quarterly Rating\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 7798.357391778938,\n   \"min\":
1.0,\n   \"max\": 19104.0,\n   \"num_unique_values\": 6,\n
\"samples\": [\n   19104.0,\n   2.008898659966499,\n
4.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

#3. Temporal Analysis:

Question- How many unique drivers are there in the dataset?

```
data['Driver_ID'].nunique()
```

```
2381
```

#OR

```
data['Driver_ID'].value_counts().sort_index(ascending=True)
```

```
Driver_ID
1        3
2        2
4        5
5        3
6        5
        ..
2784    24
2785     3
2786     9
2787     6
2788     7
Name: count, Length: 2381, dtype: int64
```

Question- How many drivers joined and left each month?

Firstly, we aim to convert the date column's datatype to a datetime format.

```
data[['MMM-YY','Dateofjoining','LastWorkingDate']].dtypes
```

```
MMM-YY             object
Dateofjoining      object
LastWorkingDate    object
dtype: object
```

```
data['Reporting_date'] = pd.to_datetime(data['MMM-
YY'],format='%d/%m/%y',errors='coerce')
data['Date_of_joining'] =
pd.to_datetime(data['Dateofjoining'],format='%d/%m/%y',errors='coerce'
)
data['Last_Working_date'] =
pd.to_datetime(data['LastWorkingDate'],format='%d/%m/%y',errors='coerc
e')
```

Drop duplicate and unused columns such as Unnamed,MMM-YY, DateofJoining, and LastWorkingDate

```
data.drop(columns={'MMM-
YY','Dateofjoining','LastWorkingDate','Unnamed:
0'},inplace=True,axis=1)

data[['Reporting_date','Date_of_joining','Last_Working_date']].dtypes
```

```
Reporting_date       datetime64[ns]
Date_of_joining      datetime64[ns]
Last_Working_date    datetime64[ns]
dtype: object
```

Now that the datatype has been changed, it's time to calculate the number of drivers who joined and left each month.

Extract month name from Date_of_joining and Last_Working_date column

```
data['Joined_month'] = data['Date_of_joining'].dt.month_name()
data['Leave_Month'] = data['Last_Working_date'].dt.month_name()

join_counts = data['Joined_month'].value_counts().sort_index()
leave_counts = data['Leave_Month'].value_counts().sort_index()

Result = pd.DataFrame({'People Joined': join_counts,'People Left':
leave_counts}).fillna(0).astype(int)

month_order = ['January', 'February', 'March', 'April', 'May',
'June','July', 'August', 'September', 'October', 'November',
'December']

Result = Result.reindex(month_order)

Result
```

```
{"summary":"{\n  \"name\": \"Result\",\n  \"rows\": 12,\n  \"fields\":
[\n    {\n      \"column\": \"People Joined\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 520,\n
\"min\": 682,\n        \"max\": 2544,\n        \"num_unique_values\":
12,\n        \"samples\": [\n          1651,\n          1844,\n
```

```
1365\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"People Left\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 24,\n        \"min\": 80,\n
\"max\": 183,\n        \"num_unique_values\": 10,\n
\"samples\": [\n          150,\n          137,\n          139\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"Result"}
```


Question- Can we determine the average tenure of drivers in the dataset?

```
Average_working_day = (data['Last_Working_date'] -
data['Date_of_joining']).mean()
print('The Average Tenure Of Drivers In OLA Is :',Average_working_day)
```

```
The Average Tenure Of Drivers In OLA Is : 370 days 11:19:00.594059404
```

4. Feature Engineering:

Question- How can we create a target variable to indicate whether a driver has left the company based on LastWorkingDate?

We created a target variable/column to indicate the driver's current status: it shows 0 if the driver is still working with Ola and 1 if the driver has left.

```
data["Current_working_status"] =
data["Last_Working_date"].apply(lambda x: 0 if pd.notna(x) else 1)

data[['Date_of_joining','Last_Working_date','Current_working_status']]
```


```
{"summary":"{\n  \"name\":
\"data[['Date_of_joining','Last_Working_date','Current_working_status'
]]\",\n  \"rows\": 19104,\n  \"fields\": [\n    {\n      \"column\":
\"Date_of_joining\",\n      \"properties\": {\n        \"dtype\":
\"date\",\n        \"min\": \"2013-01-04 00:00:00\",\n        \"max\":
\"2020-12-28 00:00:00\",\n        \"num_unique_values\": 869,\n
\"samples\": [\n          \"2019-09-14 00:00:00\",\n          \"2018-
06-01 00:00:00\",\n          \"2016-05-13 00:00:00\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Last_Working_date\",\n
\"properties\": {\n        \"dtype\": \"date\",\n        \"min\":
\"2018-12-31 00:00:00\",\n        \"max\": \"2020-12-28 00:00:00\",\n
\"num_unique_values\": 493,\n        \"samples\": [\n          \"2020-
03-05 00:00:00\",\n          \"2019-01-10 00:00:00\",\n
\"2019-07-19 00:00:00\"\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Current_working_status\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
```

```
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe"}
```

```
data['Current_working_status'].value_counts()
```

```
Current_working_status
1    17488
0     1616
Name: count, dtype: int64
```

Question- What additional features can we extract from Dateofjoining, such as tenure or duration of employment?

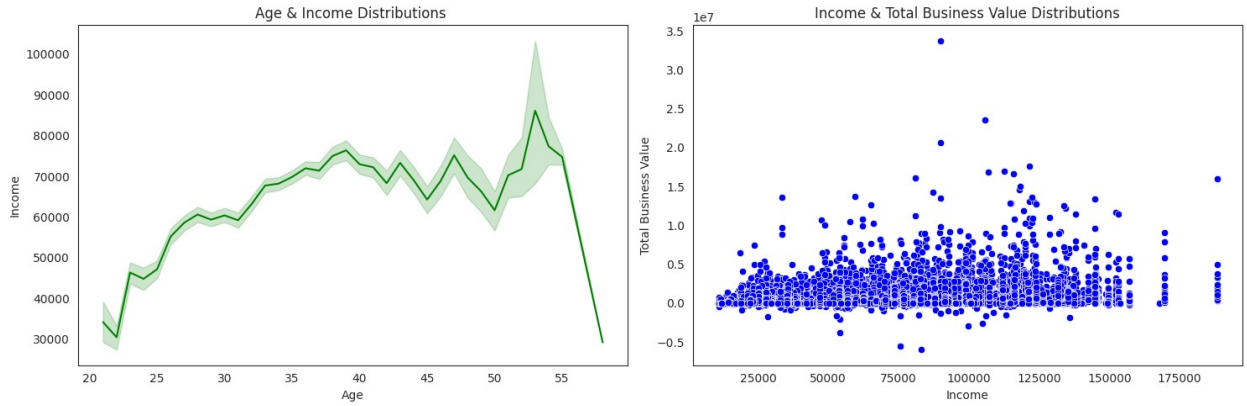
```
from datetime import datetime

data['current_date'] = datetime.now()
data['Duration_of_employee_in_day'] =
((data['Last_Working_date'].fillna(data['current_date'])) -
(data['Date_of_joining'])).dt.days

data[['Date_of_joining','Last_Working_date','Duration_of_employee_in_d
ay']]
```

```
{"summary":"{\n  \"name\":
\"data[['Date_of_joining','Last_Working_date','Duration_of_employee_in
_day']]\",\n  \"rows\": 19104,\n  \"fields\": [\n    {\n
\"column\": \"Date_of_joining\",\n      \"properties\": {\n
\"dtype\": \"date\",\n        \"min\": \"2013-01-04 00:00:00\",\n
\"max\": \"2020-12-28 00:00:00\",\n        \"num_unique_values\":
869,\n        \"samples\": [\n          \"2019-09-14 00:00:00\",\n
\"2018-06-01 00:00:00\",\n          \"2016-05-13 00:00:00\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Last_Working_date\",\n      \"properties\": {\n        \"dtype\":
\"date\",\n        \"min\": \"2018-12-31 00:00:00\",\n        \"max\":
\"2020-12-28 00:00:00\",\n        \"num_unique_values\": 493,\n
\"samples\": [\n          \"2020-03-05 00:00:00\",\n          \"2019-
01-10 00:00:00\",\n          \"2019-07-19 00:00:00\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Duration_of_employee_in_day\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
904,\n        \"min\": -274,\n        \"max\": 4368,\n
\"num_unique_values\": 1624,\n        \"samples\": [\n          3129,\
n          1148,\n          2313\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```

5. Exploratory Data Analysis (EDA):

Question- What are the distributions of Age, Income, and Total Business Value?

```
sns.set_style('white')
plt.figure(figsize=(15,5))
plt.subplot(1,2,2)
sns.scatterplot(y=data['Total Business
Value'],x=data['Income'],color='blue')
plt.title('Income & Total Business Value Distributions')
plt.subplot(1,2,1)
sns.lineplot(x=data['Age'],y=data['Income'],color='green')
plt.title('Age & Income Distributions ')
plt.tight_layout()
plt.show()
```



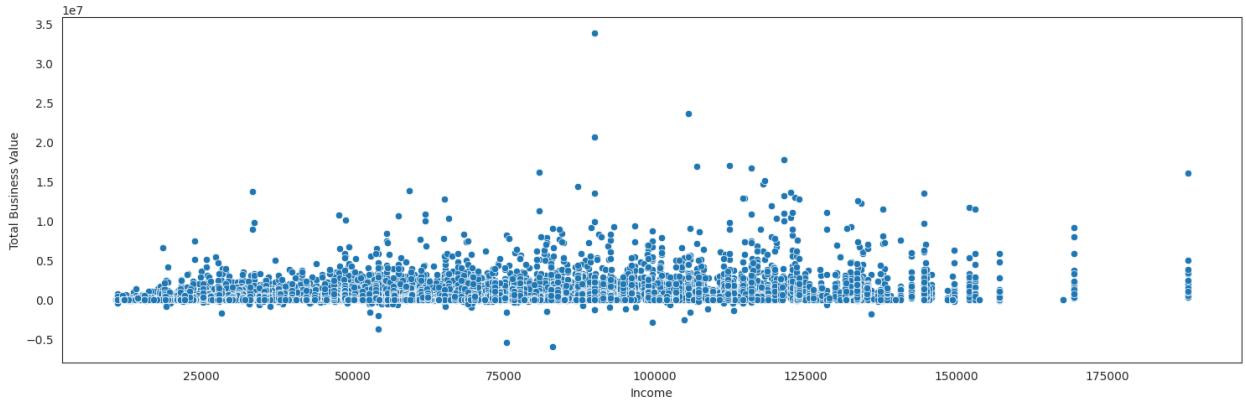
Question- How does Quarterly Rating vary across different drivers and time periods?

```
data.groupby(['Driver_ID','Date_of_joining'])['Quarterly
Rating'].mean()
```

```
Driver_ID  Date_of_joining
1          2018-12-24         2.000000
2          2020-06-11         1.000000
4          2019-07-12         1.000000
5          2019-09-01         1.000000
6          2020-07-31         1.600000
                                 ...
2784       2015-10-15         2.625000
2785       2020-08-28         1.000000
2786       2018-07-31         1.666667
2787       2018-07-21         1.500000
2788       2020-08-06         2.285714
Name: Quarterly Rating, Length: 2381, dtype: float64
```

Question- Are there any trends or patterns in the monthly income or business value acquired?

```
sns.set_style('white')
plt.figure(figsize=(15,5))
sns.scatterplot(x=data['Income'],y=data['Total Business Value'])
plt.tight_layout()
plt.show()
```



6. Missing Values Handling:

Question- How should missing values in LastWorkingDate be treated, considering it indicates whether a driver has left?

In the Last_Working_date column, if a date is present, it indicates the date the driver left the OLA company. If the column contains NaT or null values, it means the driver is currently working with OLA. To represent this scenario, we initially considered using a string value like 'Currently Working'. However, this approach caused errors during calculations because string values cannot be used in date-based computations.

To solve this problem, we replace the null (NaT) values in the Last_Working_date column with the current date. This allows us to maintain a consistent datetime format in the column, enabling accurate calculations.

```
data['current_date'] = datetime.now()
data['Last_Working_date'].fillna(value=data['current_date'],
inplace=True)
```

```
<ipython-input-29-067cea666cac>:2: FutureWarning: A value is trying to
be set on a copy of a DataFrame or Series through chained assignment
using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
```

```
original object.


  data['Last_Working_date'].fillna(value=data['current_date'],
inplace=True)

data.Last_Working_date.head()

0   2024-12-20 11:22:56.687448
1   2024-12-20 11:22:56.687448
2   2019-11-03 00:00:00.000000
3   2024-12-20 11:22:56.687448
4   2024-12-20 11:22:56.687448
Name: Last_Working_date, dtype: datetime64[ns]
```

Successfully handled missing values in the Last_Working_date column without causing errors during calculations.

7. Correlation and Relationships:

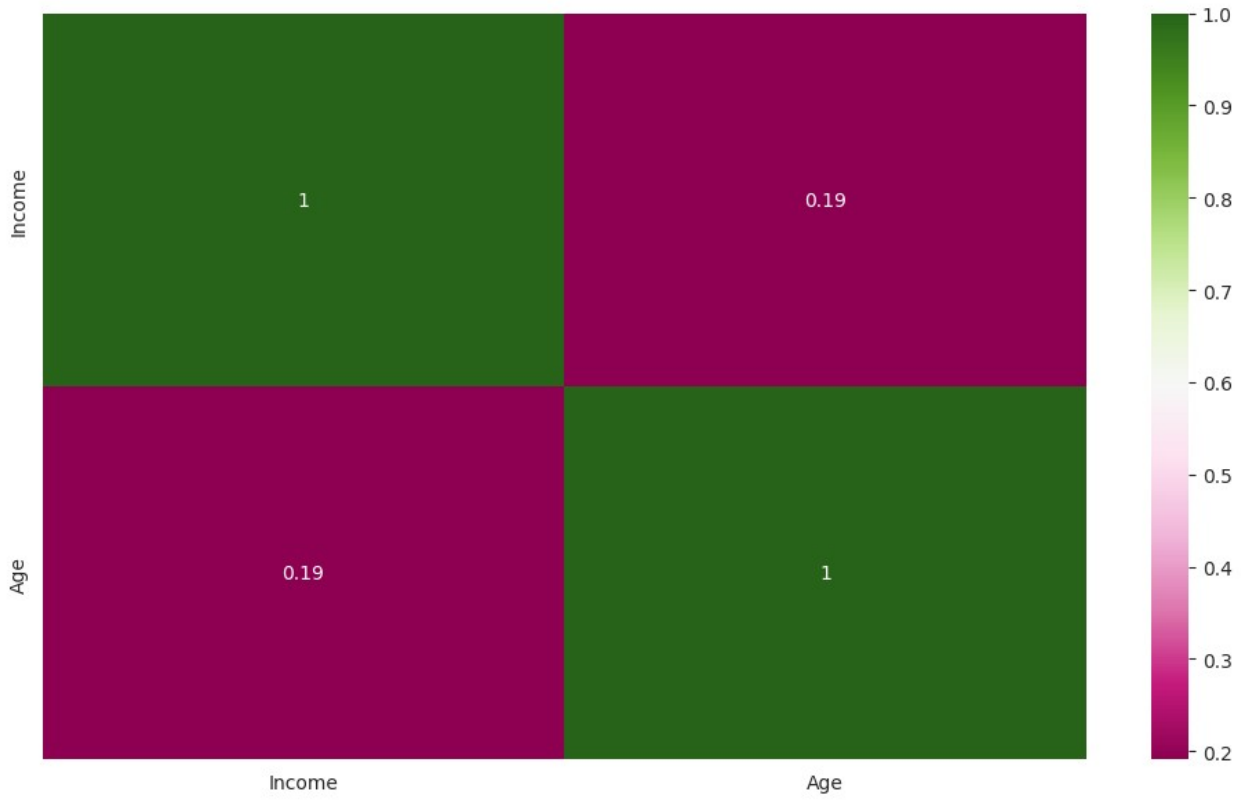
Question- Is there a correlation between Age and Income?

```
correlation = data[['Income','Age']].corr()
correlation
```


```
{"summary":"{\n  \"name\": \"correlation\",\n  \"rows\": 2,\n
\"fields\": [\n    {\n      \"column\": \"Income\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.5719703496724837,\n        \"min\": 0.19111177421789194,\n
\"max\": 1.0,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0.19111177421789194,\n          1.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.5719703496724837,\n
\"min\": 0.19111177421789194,\n        \"max\": 1.0,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1.0,\n
0.19111177421789194\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"correlation"}
```


Based on the correlation coefficient of 0.19 between the Age and Income columns, there is a very weak correlation between Age and Income.

```
plt.figure(figsize=(10,6))
sns.heatmap(correlation, annot=True, cmap='PiYG')
plt.tight_layout()
plt.show()
```



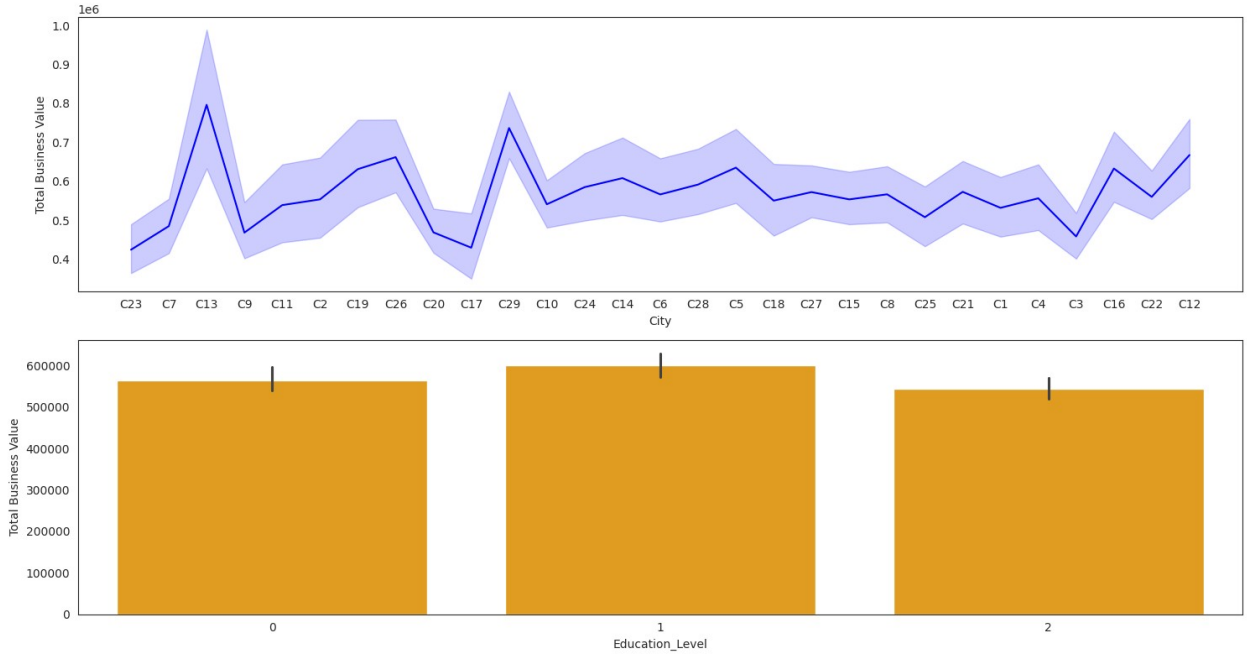
Question- How do Education_Level and City affect Total Business Value?

```
new_matrix =
data.pivot_table(index='City',columns='Education_Level',values='Total
Business Value')
new_matrix
```


```
{"summary":"{\n  \"name\": \"new_matrix\",\n  \"rows\": 29,\n
\"fields\": [\n    {\n      \"column\": \"City\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 29,\n        \"samples\": [\n          \"C8\",\
n          \"C24\",\n          \"C20\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": 0,\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 194978.0469309622,\n
\"min\": 158471.66666666666,\n        \"max\": 907502.5974025974,\n
\"num_unique_values\": 29,\n        \"samples\": [\n
390270.6763285024,\n          561527.7401129943,\n
592213.1513647643\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 1,\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
153556.746355644,\n        \"min\": 231839.5,\n        \"max\":
1037106.387434555,\n        \"num_unique_values\": 29,\n
\"samples\": [\n          633606.393442623,\n
467602.64705882355,\n          336556.5789473684\n        ],\n
```

\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": 2,\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 153590.65203418076,\n
\"min\": 225343.46456692912,\n   \"max\": 755015.5522388059,\n
\"num_unique_values\": 29,\n   \"samples\": [\n
643064.8659003831,\n   674646.329588015,\n
425067.6696165192\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"new_matrix"}

```
sns.set_style('white')
plt.figure(figsize=(15,8))
plt.subplot(2,1,1)
sns.lineplot(x=data['City'],y=data['Total Business
Value'],markers='o',color='blue')
plt.subplot(2,1,2)
sns.barplot(x=data['Education_Level'],y=data['Total Business
Value'],color='orange')
plt.tight_layout()
plt.show()
```



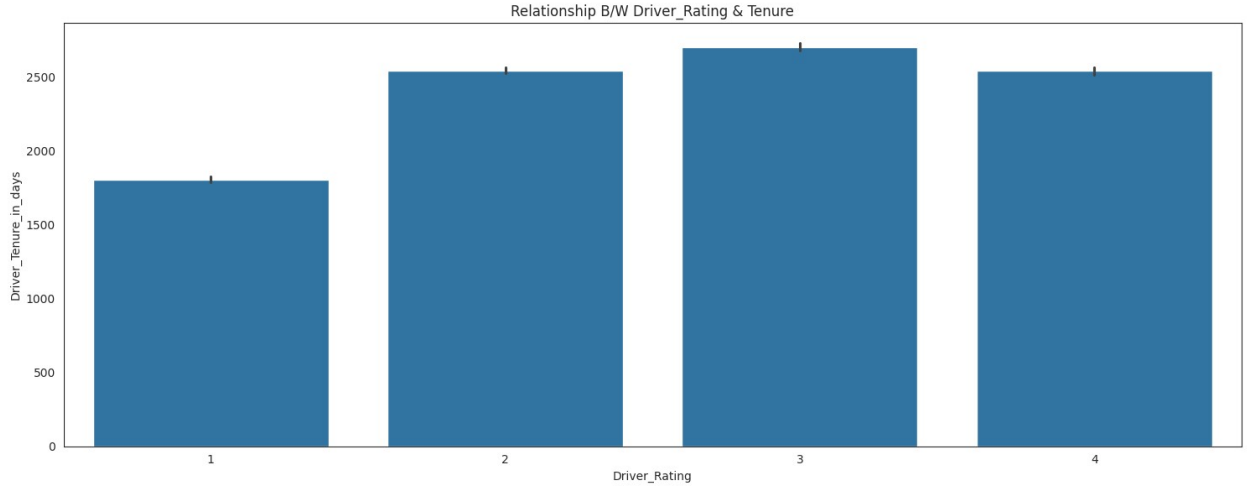
Question- Are drivers with higher Quarterly Rating more likely to stay longer?

```
data[['Quarterly Rating','Duration_of_employee_in_day']]
```

{"summary":"{\n \"name\": \"data[['Quarterly
Rating','Duration_of_employee_in_day']]\",\n \"rows\": 19104,\n
\"fields\": [\n   {\n   \"column\": \"Quarterly Rating\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":

```
1,\n        \"min\": 1,\n        \"max\": 4,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          1,\n
3,\n          2\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Duration_of_employee_in_day\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 904,\n        \"min\": -274,\
n        \"max\": 4368,\n        \"num_unique_values\": 1624,\n
\"samples\": [\n          3129,\n          1148,\n          2313\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```

```
sns.set_style('white')
plt.figure(figsize=(15,6))
sns.barplot(x=data['Quarterly
Rating'],y=data['Duration_of_employee_in_day'])
plt.xlabel('Driver_Rating')
plt.ylabel('Driver_Tenure_in_days')
plt.title('Relationship B/W Driver_Rating & Tenure')
plt.tight_layout()
plt.show()
```



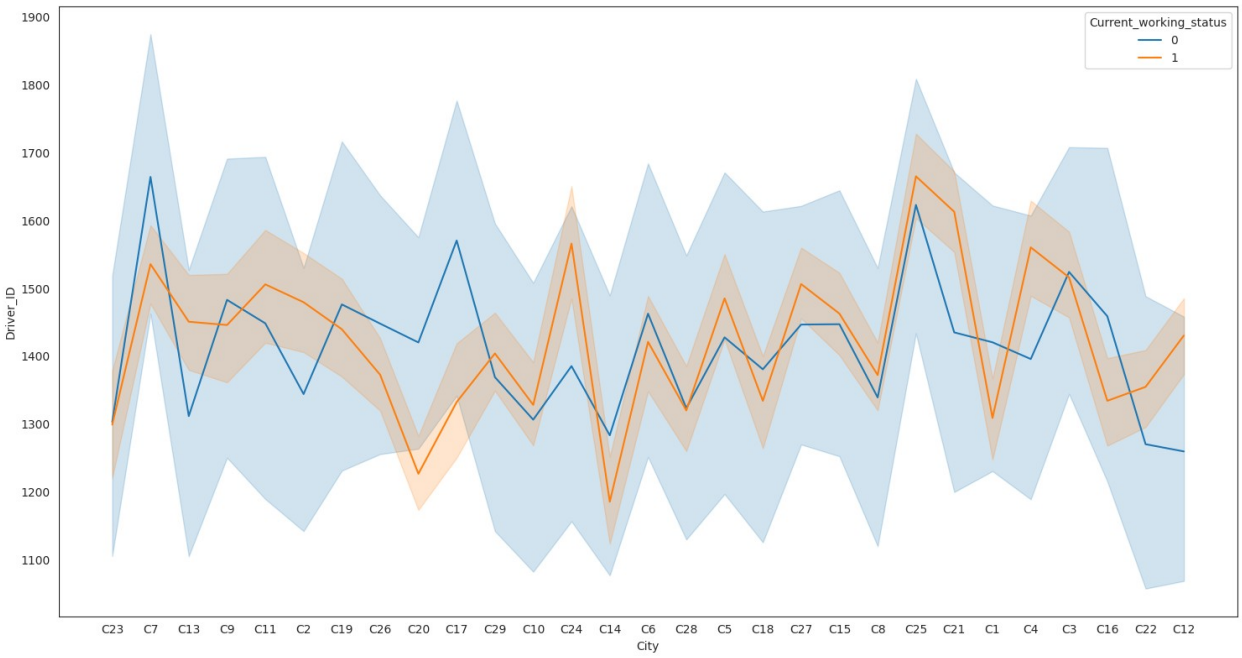
8. Predictive Analysis(Optional) :

Question- Can we predict which drivers are likely to leave based on their demographic and performance attributes?

Driver leave based on their demographic and performance

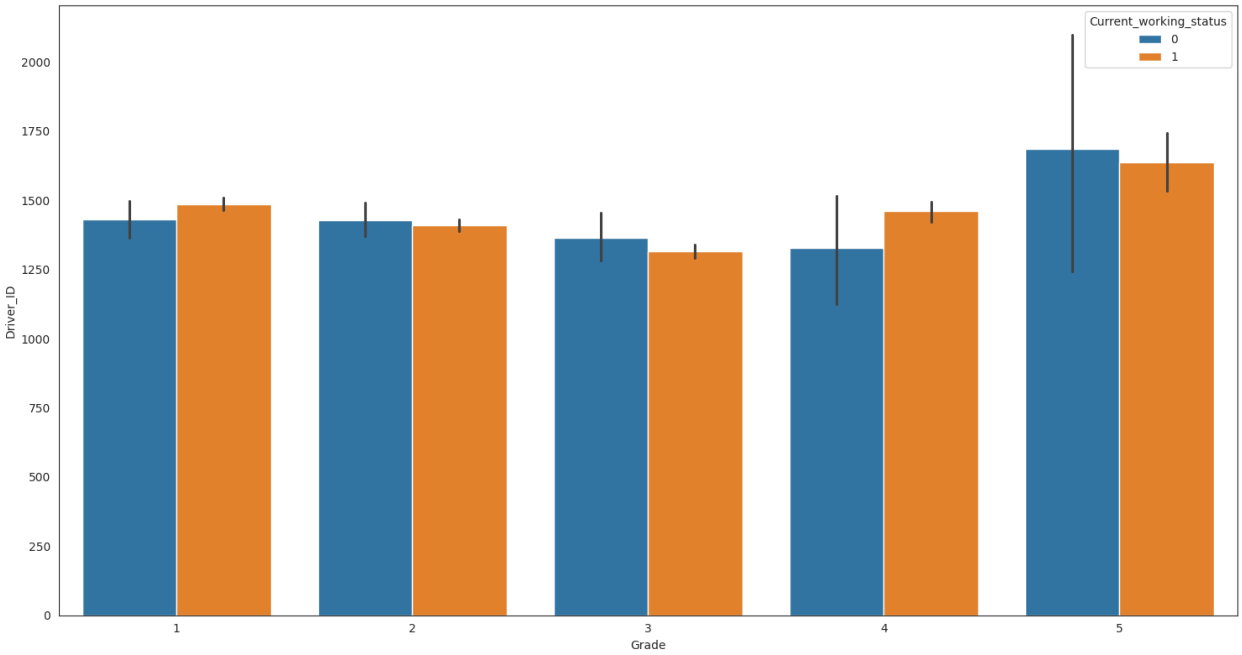
```
sns.set_style('white')
plt.figure(figsize=(15,8))
sns.lineplot(x=data['City'],y=data['Driver_ID'],hue=data['Current_work
ing_status'],markers='o')
```

```
plt.tight_layout()
plt.show()
```



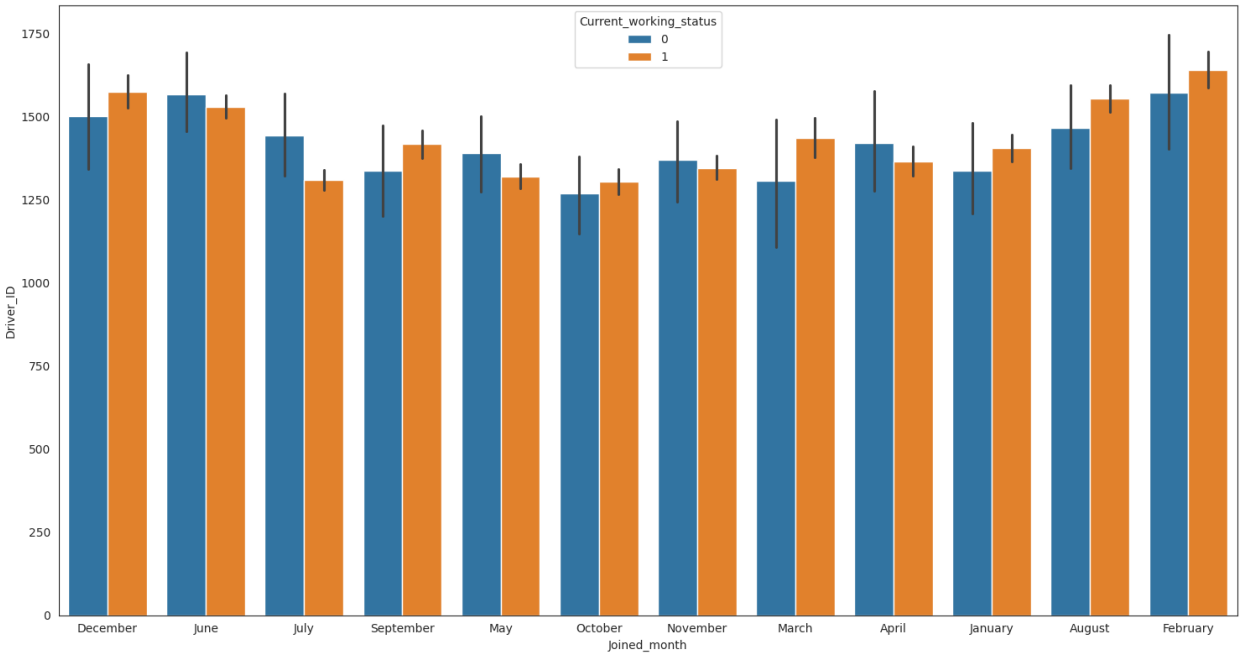
Driver leave based on their Grade

```
sns.set_style('white')
plt.figure(figsize=(15,8))
sns.barplot(x=data['Grade'],y=data['Driver_ID'],hue=data['Current_work
ing_status'])
plt.tight_layout()
plt.show()
```



Driver leave based on Year Month

```
sns.set_style('white')
plt.figure(figsize=(15,8))
sns.barplot(x=data['Joined_month'],y=data['Driver_ID'],hue=data['Curre
nt_working_status'])
plt.tight_layout()
plt.show()
```



#Actionable Insights

• Ola has a total of 2,381 unique drivers.

• Driver attrition rate: o Monthly driver departures: January: 134, February: 137, March: 117, April: 126, May: 147 June: 139, July: 183, August: 80, September: 150, October: 117 November: 149, December: 137

• Average tenure: Drivers work for approximately 370 days before leaving.

Driver Demographics and Income • Highest earning age group: Drivers aged 30-55 have the highest income. • High attrition age groups: Drivers aged 30-40 and 55+ exhibit the highest attrition rates.

• Income range: Almost all drivers fall within an income range of ₹50,000-₹1,25,000.

Business Value Insights • Total business value range: Most drivers generate a total business value between ₹0-₹0.5.

• Top cities by business value: Cities C13, C29, C5, C16, and C12 have the highest business value.
• Lowest business value cities: Cities C9, C17, C25, and C3 have the lowest business value.

Correlations and Education • Age vs. Income correlation: A correlation coefficient of 0.19 suggests a very weak positive relationship between age and income.

• Impact of education: Education level does not significantly affect total business value: o High School+, Intermediate+, and Graduate+ education levels all have a business value range of approximately ₹50,000-₹5,50,000.

Driver Ratings • Driver rating and work duration: o Drivers with a rating of 1 have the lowest participation in Ola. o Drivers with a rating of 3 are the most active. o Ratings of 2 and 4 show similar levels of participation.

#Recommendations

Recommendations to Improve Ola Driver Churn Rate

Focus on drivers aged 30-40 and 55+ since they have the highest leaving rates. Conduct surveys or exit interviews to understand their challenges and provide tailored solutions, such as: Flexible working hours for older drivers. Targeted incentives for drivers aged 30-40 to reduce attrition. Enhance Driver Earnings Potential

Introduce performance-based incentives for drivers in the ₹50,000-₹1,25,000 income range to increase their business value.

Offer bonuses or commission boosts for drivers operating in low business value cities (C9, C17, C25, and C3) to motivate them.

Improve Retention Through Tenure-Based Rewards Implement loyalty programs where drivers receive additional perks or incentives based on their tenure milestones (e.g., after 6 months, 1 year, etc.).

Support Drivers in High Attrition Months Analyze operational factors leading to high attrition months like July, September, and November and introduce retention strategies during these periods: Seasonal bonuses or reduced commission rates during these months. Address work-life

balance concerns that may contribute to higher departures. Leverage Data Insights for Proactive Intervention

Use the correlation between age and income (weak but positive) to identify drivers at risk of attrition due to low income and offer targeted support programs.

Monitor drivers with low ratings (1) and provide them with training and support to improve performance and engagement.

Introduce Flexible Work Schedules Provide part-time or flexible working options for drivers who may not be able to commit to full-time driving, especially those in the 55+ age group. Enhance Driver Support and Engagement

Improve Driver Ratings Offer workshops or training to help drivers improve their customer interaction skills and receive higher ratings, especially targeting drivers with ratings below 3. Incentivize drivers with consistent high ratings to foster long-term commitment. Focus on Low Business Value Cities

Develop city-specific strategies for C9, C17, C25, and C3 to boost demand and driver profitability in these areas. This can include: Collaborating with local businesses to increase ride volumes. Running city-specific marketing campaigns to attract riders. Strengthen Onboarding and Early Career Support

Set up regular feedback mechanisms like surveys or driver town halls to understand their concerns and build a sense of belonging. Provide better support during peak demand times, ensuring drivers do not feel overworked.