ola

February 11, 2025

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

```
[ ]: !gdown  https://drive.google.com/uc?id=1UxMh76d4Tk2UybxxOR8nbc66mTBoWeqO
```

```
Downloading…
From: https://drive.google.com/uc?id=1UxMh76d4Tk2UybxxOR8nbc66mTBoWeqO
To: /content/ola_driver_scaler.csv
100% 1.13M/1.13M [00:00<00:00, 27.8MB/s]
```

```
[ ]: df= pd.read_csv("ola_driver_scaler.csv")
```

```
[ ]: df
```

```
/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")
```

```
[ ]:        Unnamed: 0    MMM-YY  Driver_ID   Age  Gender City  Education_Level  \
     0               0  01/01/19          1  28.0     0.0  C23                2
     1               1  02/01/19          1  28.0     0.0  C23                2
     2               2  03/01/19          1  28.0     0.0  C23                2
     3               3  11/01/20          2  31.0     0.0   C7                2
     4               4  12/01/20          2  31.0     0.0   C7                2
     …             …       …          …    …      …   …                …
     19099       19099  08/01/20       2788  30.0     0.0  C27                2
     19100       19100  09/01/20       2788  30.0     0.0  C27                2
     19101       19101  10/01/20       2788  30.0     0.0  C27                2
     19102       19102  11/01/20       2788  30.0     0.0  C27                2
     19103       19103  12/01/20       2788  30.0     0.0  C27                2

            Income Dateofjoining LastWorkingDate  Joining Designation  Grade  \
     0       57387      24/12/18             NaN                    1      1
     1       57387      24/12/18             NaN                    1      1
```

```
2       57387      24/12/18        03/11/19                    1      1
3       67016      11/06/20             NaN                    2      2
4       67016      11/06/20             NaN                    2      2
…       …           …             …                  …    …
19099   70254      06/08/20             NaN                    2      2
19100   70254      06/08/20             NaN                    2      2
19101   70254      06/08/20             NaN                    2      2
19102   70254      06/08/20             NaN                    2      2
19103   70254      06/08/20             NaN                    2      2

       Total Business Value  Quarterly Rating
0                   2381060                 2
1                   -665480                 2
2                         0                 2
3                         0                 1
4                         0                 1
…                      …               …
19099                740280                 3
19100                448370                 3
19101                     0                 2
19102                200420                 2
19103                411480                 2

[19104 rows x 14 columns]
```

[ ]:
```
df.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")
```

[ ]:
```
   Unnamed: 0    MMM-YY  Driver_ID   Age  Gender City  Education_Level  \
0           0  01/01/19          1  28.0     0.0  C23                2
1           1  02/01/19          1  28.0     0.0  C23                2
2           2  03/01/19          1  28.0     0.0  C23                2
3           3  11/01/20          2  31.0     0.0   C7                2
4           4  12/01/20          2  31.0     0.0   C7                2

   Income Dateofjoining LastWorkingDate  Joining Designation  Grade  \
0   57387      24/12/18             NaN                    1      1
1   57387      24/12/18             NaN                    1      1
2   57387      24/12/18        03/11/19                    1      1
3   67016      11/06/20             NaN                    2      2
4   67016      11/06/20             NaN                    2      2
```

```
   Total Business Value  Quarterly Rating
0               2381060                 2
1               -665480                 2
2                     0                 2
3                     0                 1
4                     0                 1
```

```
[ ]: df.tail()
```

```
[ ]:        Unnamed: 0    MMM-YY  Driver_ID   Age  Gender City  Education_Level  \
     19099       19099  08/01/20       2788  30.0     0.0  C27                2
     19100       19100  09/01/20       2788  30.0     0.0  C27                2
     19101       19101  10/01/20       2788  30.0     0.0  C27                2
     19102       19102  11/01/20       2788  30.0     0.0  C27                2
     19103       19103  12/01/20       2788  30.0     0.0  C27                2

            Income Dateofjoining LastWorkingDate  Joining Designation  Grade  \
     19099   70254      06/08/20             NaN                    2      2
     19100   70254      06/08/20             NaN                    2      2
     19101   70254      06/08/20             NaN                    2      2
     19102   70254      06/08/20             NaN                    2      2
     19103   70254      06/08/20             NaN                    2      2

            Total Business Value  Quarterly Rating
     19099                740280                 3
     19100                448370                 3
     19101                     0                 2
     19102                200420                 2
     19103                411480                 2
```

```
[ ]: df.columns
```

```
[ ]: Index(['Unnamed: 0', 'MMM-YY', 'Driver_ID', 'Age', 'Gender', 'City',
            'Education_Level', 'Income', 'Dateofjoining', 'LastWorkingDate',
            'Joining Designation', 'Grade', 'Total Business Value',
            'Quarterly Rating'],
           dtype='object')
```

```
[ ]: #Since, we don't need 'Unnamed: 0', We drop the column
     df.drop(['Unnamed: 0'], axis = 1,inplace =True)
```

```
[ ]: #df.drop(['Driver_ID'], axis = 1,inplace =True)
```

```
[ ]: df.drop(['MMM-YY'], axis = 1,inplace =True)
```

```
[ ]: df
```

```
/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")
```

```
[ ]:        Driver_ID   Age  Gender City  Education_Level  Income Dateofjoining  \
     0              1  28.0     0.0  C23                2   57387      24/12/18
     1              1  28.0     0.0  C23                2   57387      24/12/18
     2              1  28.0     0.0  C23                2   57387      24/12/18
     3              2  31.0     0.0   C7                2   67016      11/06/20
     4              2  31.0     0.0   C7                2   67016      11/06/20
     …            …    …       …   …               …      …             …
     19099       2788  30.0     0.0  C27                2   70254      06/08/20
     19100       2788  30.0     0.0  C27                2   70254      06/08/20
     19101       2788  30.0     0.0  C27                2   70254      06/08/20
     19102       2788  30.0     0.0  C27                2   70254      06/08/20
     19103       2788  30.0     0.0  C27                2   70254      06/08/20

          LastWorkingDate  Joining Designation  Grade  Total Business Value  \
     0                NaN                    1      1               2381060
     1                NaN                    1      1               -665480
     2           03/11/19                    1      1                     0
     3                NaN                    2      2                     0
     4                NaN                    2      2                     0
     …                …                    …     …                    …
     19099            NaN                    2      2                740280
     19100            NaN                    2      2                448370
     19101            NaN                    2      2                     0
     19102            NaN                    2      2                200420
     19103            NaN                    2      2                411480

            Quarterly Rating
     0                     2
     1                     2
     2                     2
     3                     1
     4                     1
     …                   …
     19099                 3
     19100                 3
     19101                 2
     19102                 2
     19103                 2

     [19104 rows x 12 columns]
```

[ ]:
```
df.info
```

[ ]:
```
<bound method DataFrame.info of        Driver_ID   Age  Gender City
Education_Level  Income Dateofjoining  \
0              1  28.0     0.0  C23                2   57387     24/12/18
1              1  28.0     0.0  C23                2   57387     24/12/18
2              1  28.0     0.0  C23                2   57387     24/12/18
3              2  31.0     0.0   C7                2   67016     11/06/20
4              2  31.0     0.0   C7                2   67016     11/06/20
…          …   …     …  …              …     …           …
19099       2788  30.0     0.0  C27                2   70254     06/08/20
19100       2788  30.0     0.0  C27                2   70254     06/08/20
19101       2788  30.0     0.0  C27                2   70254     06/08/20
19102       2788  30.0     0.0  C27                2   70254     06/08/20
19103       2788  30.0     0.0  C27                2   70254     06/08/20

      LastWorkingDate  Joining Designation  Grade  Total Business Value  \
0                 NaN                    1      1               2381060
1                 NaN                    1      1               -665480
2            03/11/19                    1      1                     0
3                 NaN                    2      2                     0
4                 NaN                    2      2                     0
…                …                  …    …                   …
19099             NaN                    2      2                740280
19100             NaN                    2      2                448370
19101             NaN                    2      2                     0
19102             NaN                    2      2                200420
19103             NaN                    2      2                411480

       Quarterly Rating
0                     2
1                     2
2                     2
3                     1
4                     1
…                 …
19099                 3
19100                 3
19101                 2
19102                 2
19103                 2

[19104 rows x 12 columns]>
```

[ ]:
```
df.dtypes
```

```
[ ]: 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
```

```
[ ]: df.isnull().sum()
```

```
[ ]: 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
```

```
[ ]: df.isna().sum()
```

```
[ ]: 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
```

```
[ ]: df.describe()
```

[ ]:

```
          Driver_ID           Age        Gender  Education_Level  \
count  19104.000000  19043.000000  19052.000000     19104.000000
mean    1415.591133     34.668435      0.418749         1.021671
std      810.705321      6.257912      0.493367         0.800167
min        1.000000     21.000000      0.000000         0.000000
25%      710.000000     30.000000      0.000000         0.000000
50%     1417.000000     34.000000      0.000000         1.000000
75%     2137.000000     39.000000      1.000000         2.000000
max     2788.000000     58.000000      1.000000         2.000000

              Income  Joining Designation         Grade  Total Business Value  \
count   19104.000000         19104.000000  19104.000000          1.910400e+04
mean    65652.025126             1.690536      2.252670          5.716621e+05
std     30914.515344             0.836984      1.026512          1.128312e+06
min     10747.000000             1.000000      1.000000         -6.000000e+06
25%     42383.000000             1.000000      1.000000          0.000000e+00
50%     60087.000000             1.000000      2.000000          2.500000e+05
75%     83969.000000             2.000000      3.000000          6.997000e+05
max    188418.000000             5.000000      5.000000          3.374772e+07

       Quarterly Rating
count      19104.000000
mean           2.008899
std            1.009832
min            1.000000
25%            1.000000
50%            2.000000
75%            3.000000
max            4.000000
```

[ ]:

```
df['Gender'].unique()
```

[ ]:

```
array([ 0.,  1., nan])
```

[ ]:

```
df['Education_Level'].unique()
```

[ ]:

```
array([2, 0, 1])
```

[ ]:

```
df['Quarterly Rating'].unique()
```

[ ]:

```
array([2, 1, 4, 3])
```

[ ]:

```
df['Grade'].unique()
```

[ ]:

```
array([1, 2, 3, 4, 5])
```

[ ]:

```
df['Joining Designation'].unique()
```

```
[ ]: array([1, 2, 3, 4, 5])
```

```
[ ]: df["Total Business Value"]
```

```
[ ]: 0        2381060
     1        -665480
     2              0
     3              0
     4              0
                 …
     19099     740280
     19100     448370
     19101          0
     19102     200420
     19103     411480
     Name: Total Business Value, Length: 19104, dtype: int64
```

```
[ ]: df.columns
```

```
[ ]: Index(['Driver_ID', 'Age', 'Gender', 'City', 'Education_Level', 'Income',
            'Dateofjoining', 'LastWorkingDate', 'Joining Designation', 'Grade',
            'Total Business Value', 'Quarterly Rating'],
           dtype='object')
```

```
[ ]: df["Dateofjoining"]
```

```
[ ]: 0        24/12/18
     1        24/12/18
     2        24/12/18
     3        11/06/20
     4        11/06/20
                …
     19099    06/08/20
     19100    06/08/20
     19101    06/08/20
     19102    06/08/20
     19103    06/08/20
     Name: Dateofjoining, Length: 19104, dtype: object
```

```
[ ]: df["Dateofjoining"]=pd.to_datetime(df["Dateofjoining"])
     df['Joining_day'] = df['Dateofjoining'].dt.day
     df['Joining_month'] = df['Dateofjoining'].dt.month
     df['Joining_year'] = df['Dateofjoining'].dt.year

     df["LastWorkingDate"]=pd.to_datetime(df["LastWorkingDate"])
     df['LastWorking_day'] = df['LastWorkingDate'].dt.day
     df['LastWorking_month'] = df['LastWorkingDate'].dt.month
```

```
df['LastWorking_year'] = df['LastWorkingDate'].dt.year
```

```
<ipython-input-135-ad2e9b8f7986>:1: 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.
  df["Dateofjoining"]=pd.to_datetime(df["Dateofjoining"])
<ipython-input-135-ad2e9b8f7986>:6: 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.
  df["LastWorkingDate"]=pd.to_datetime(df["LastWorkingDate"])
```

```
[ ]: df
```

```
[ ]:        Driver_ID   Age  Gender City  Education_Level  Income Dateofjoining  \
     0              1  28.0     0.0  C23                2   57387    2018-12-24
     1              1  28.0     0.0  C23                2   57387    2018-12-24
     2              1  28.0     0.0  C23                2   57387    2018-12-24
     3              2  31.0     0.0   C7                2   67016    2020-11-06
     4              2  31.0     0.0   C7                2   67016    2020-11-06
     …            …    …      …   …              …      …           …
     19099       2788  30.0     0.0  C27                2   70254    2020-06-08
     19100       2788  30.0     0.0  C27                2   70254    2020-06-08
     19101       2788  30.0     0.0  C27                2   70254    2020-06-08
     19102       2788  30.0     0.0  C27                2   70254    2020-06-08
     19103       2788  30.0     0.0  C27                2   70254    2020-06-08

           LastWorkingDate  Joining Designation  Grade  Total Business Value  \
     0                 NaT                    1      1               2381060
     1                 NaT                    1      1               -665480
     2          2019-03-11                    1      1                     0
     3                 NaT                    2      2                     0
     4                 NaT                    2      2                     0
     …               …                    …    …                   …
     19099             NaT                    2      2                740280
     19100             NaT                    2      2                448370
     19101             NaT                    2      2                     0
     19102             NaT                    2      2                200420
     19103             NaT                    2      2                411480

            Quarterly Rating  Joining_day  Joining_month  Joining_year  \
     0                     2           24             12          2018
     1                     2           24             12          2018
     2                     2           24             12          2018
     3                     1            6             11          2020
     4                     1            6             11          2020
     …                   …          …            …           …
     19099                 3            8              6          2020
```

```
19100                 3            8              6          2020
19101                 2            8              6          2020
19102                 2            8              6          2020
19103                 2            8              6          2020

       LastWorking_day  LastWorking_month  LastWorking_year
0                  NaN                NaN               NaN
1                  NaN                NaN               NaN
2                 11.0                3.0            2019.0
3                  NaN                NaN               NaN
4                  NaN                NaN               NaN
…                …                …               …
19099              NaN                NaN               NaN
19100              NaN                NaN               NaN
19101              NaN                NaN               NaN
19102              NaN                NaN               NaN
19103              NaN                NaN               NaN

[19104 rows x 18 columns]
```

```
[ ]: df.drop('LastWorkingDate' ,axis =1, inplace=True)
```

```
[ ]: df
```

```
[ ]:       Driver_ID   Age  Gender City  Education_Level  Income Dateofjoining  \
0             1  28.0     0.0  C23                2   57387    2018-12-24
1             1  28.0     0.0  C23                2   57387    2018-12-24
2             1  28.0     0.0  C23                2   57387    2018-12-24
3             2  31.0     0.0   C7                2   67016    2020-11-06
4             2  31.0     0.0   C7                2   67016    2020-11-06
…          …   …     …  …              …     …           …
19099      2788  30.0     0.0  C27                2   70254    2020-06-08
19100      2788  30.0     0.0  C27                2   70254    2020-06-08
19101      2788  30.0     0.0  C27                2   70254    2020-06-08
19102      2788  30.0     0.0  C27                2   70254    2020-06-08
19103      2788  30.0     0.0  C27                2   70254    2020-06-08

       Joining Designation  Grade  Total Business Value  Quarterly Rating  \
0                        1      1               2381060                 2
1                        1      1               -665480                 2
2                        1      1                     0                 2
3                        2      2                     0                 1
4                        2      2                     0                 1
…                    …    …                   …                …
19099                    2      2                740280                 3
19100                    2      2                448370                 3
19101                    2      2                     0                 2
```

```
19102                        2      2               200420                     2
19103                        2      2               411480                     2

       Joining_day  Joining_month  Joining_year  LastWorking_day  \
0               24             12          2018              NaN
1               24             12          2018              NaN
2               24             12          2018             11.0
3                6             11          2020              NaN
4                6             11          2020              NaN
…            …            …           …              …
19099            8              6          2020              NaN
19100            8              6          2020              NaN
19101            8              6          2020              NaN
19102            8              6          2020              NaN
19103            8              6          2020              NaN

       LastWorking_month  LastWorking_year
0                    NaN               NaN
1                    NaN               NaN
2                    3.0            2019.0
3                    NaN               NaN
4                    NaN               NaN
…                  …               …
19099                NaN               NaN
19100                NaN               NaN
19101                NaN               NaN
19102                NaN               NaN
19103                NaN               NaN

[19104 rows x 17 columns]
```

```
[ ]: df
```

```
[ ]:       Driver_ID   Age  Gender City  Education_Level  Income Dateofjoining  \
     0              1  28.0     0.0  C23                2   57387    2018-12-24
     1              1  28.0     0.0  C23                2   57387    2018-12-24
     2              1  28.0     0.0  C23                2   57387    2018-12-24
     3              2  31.0     0.0   C7                2   67016    2020-11-06
     4              2  31.0     0.0   C7                2   67016    2020-11-06
     …          …   …     …  …               …     …           …
     19099       2788  30.0     0.0  C27                2   70254    2020-06-08
     19100       2788  30.0     0.0  C27                2   70254    2020-06-08
     19101       2788  30.0     0.0  C27                2   70254    2020-06-08
     19102       2788  30.0     0.0  C27                2   70254    2020-06-08
     19103       2788  30.0     0.0  C27                2   70254    2020-06-08

            Joining Designation  Grade  Total Business Value  Quarterly Rating  \
     0                        1      1                2381060                 2
     1                        1      1                -665480                 2
     2                        1      1                      0                 2
     3                        2      2                      0                 1
     4                        2      2                      0                 1
     …                     …    …                   …               …
     19099                    2      2                 740280                 3
     19100                    2      2                 448370                 3
     19101                    2      2                      0                 2
     19102                    2      2                 200420                 2
     19103                    2      2                 411480                 2

            Joining_day  Joining_month  Joining_year  LastWorking_day  \
     0               24             12          2018              NaN
     1               24             12          2018              NaN
     2               24             12          2018             11.0
     3                6             11          2020              NaN
     4                6             11          2020              NaN
     …             …            …           …              …
     19099            8              6          2020              NaN
     19100            8              6          2020              NaN
     19101            8              6          2020              NaN
     19102            8              6          2020              NaN
     19103            8              6          2020              NaN

            LastWorking_month  LastWorking_year
     0                    NaN               NaN
     1                    NaN               NaN
     2                    3.0            2019.0
     3                    NaN               NaN
     4                    NaN               NaN
     …                   …               …
     19099                NaN               NaN
     19100                NaN               NaN
     19101                NaN               NaN
     19102                NaN               NaN
     19103                NaN               NaN

     [19104 rows x 17 columns]
```

```
[ ]: df["LastWorking_day"]
```

```
[ ]: 0          NaN
     1          NaN
     2         11.0
     3          NaN
     4          NaN
```

```
          …
19099     NaN
19100     NaN
19101     NaN
19102     NaN
19103     NaN
Name: LastWorking_day, Length: 19104, dtype: float64
```

```
[ ]: df
```

```
[ ]:
       Driver_ID   Age  Gender City  Education_Level  Income Dateofjoining  \
0              1  28.0     0.0  C23                2   57387    2018-12-24
1              1  28.0     0.0  C23                2   57387    2018-12-24
2              1  28.0     0.0  C23                2   57387    2018-12-24
3              2  31.0     0.0   C7                2   67016    2020-11-06
4              2  31.0     0.0   C7                2   67016    2020-11-06
…          …   …     …  …              …     …           …
19099       2788  30.0     0.0  C27                2   70254    2020-06-08
19100       2788  30.0     0.0  C27                2   70254    2020-06-08
19101       2788  30.0     0.0  C27                2   70254    2020-06-08
19102       2788  30.0     0.0  C27                2   70254    2020-06-08
19103       2788  30.0     0.0  C27                2   70254    2020-06-08

       Joining Designation  Grade  Total Business Value  Quarterly Rating  \
0                        1      1               2381060                 2
1                        1      1               -665480                 2
2                        1      1                     0                 2
3                        2      2                     0                 1
4                        2      2                     0                 1
…                    …    …                   …               …
19099                    2      2                740280                 3
19100                    2      2                448370                 3
19101                    2      2                     0                 2
19102                    2      2                200420                 2
19103                    2      2                411480                 2

       Joining_day  Joining_month  Joining_year  LastWorking_day  \
0               24             12          2018              NaN
1               24             12          2018              NaN
2               24             12          2018             11.0
3                6             11          2020              NaN
4                6             11          2020              NaN
…            …            …           …              …
19099            8              6          2020              NaN
19100            8              6          2020              NaN
19101            8              6          2020              NaN
19102            8              6          2020              NaN
```

```
19103            8              6          2020                NaN

       LastWorking_month  LastWorking_year
0                    NaN               NaN
1                    NaN               NaN
2                    3.0            2019.0
3                    NaN               NaN
4                    NaN               NaN
…                  …                …
19099                NaN               NaN
19100                NaN               NaN
19101                NaN               NaN
19102                NaN               NaN
19103                NaN               NaN

[19104 rows x 17 columns]
```

[ ]:
```
df.fillna(0, inplace=True)
```

[ ]:
```
df
```

[ ]:
```
       Driver_ID   Age  Gender City  Education_Level  Income Dateofjoining  \
0              1  28.0     0.0  C23                2   57387    2018-12-24
1              1  28.0     0.0  C23                2   57387    2018-12-24
2              1  28.0     0.0  C23                2   57387    2018-12-24
3              2  31.0     0.0   C7                2   67016    2020-11-06
4              2  31.0     0.0   C7                2   67016    2020-11-06
…          …   …     …  …              …     …            …
19099       2788  30.0     0.0  C27                2   70254    2020-06-08
19100       2788  30.0     0.0  C27                2   70254    2020-06-08
19101       2788  30.0     0.0  C27                2   70254    2020-06-08
19102       2788  30.0     0.0  C27                2   70254    2020-06-08
19103       2788  30.0     0.0  C27                2   70254    2020-06-08

       Joining Designation  Grade  Total Business Value  Quarterly Rating  \
0                        1      1               2381060                 2
1                        1      1               -665480                 2
2                        1      1                     0                 2
3                        2      2                     0                 1
4                        2      2                     0                 1
…                    …    …                   …                …
19099                    2      2                740280                 3
19100                    2      2                448370                 3
19101                    2      2                     0                 2
19102                    2      2                200420                 2
19103                    2      2                411480                 2
```

```
       Joining_day  Joining_month  Joining_year  LastWorking_day  \
0               24             12          2018              0.0
1               24             12          2018              0.0
2               24             12          2018             11.0
3                6             11          2020              0.0
4                6             11          2020              0.0
…            …             …           …               …
19099            8              6          2020              0.0
19100            8              6          2020              0.0
19101            8              6          2020              0.0
19102            8              6          2020              0.0
19103            8              6          2020              0.0

       LastWorking_month  LastWorking_year
0                    0.0               0.0
1                    0.0               0.0
2                    3.0            2019.0
3                    0.0               0.0
4                    0.0               0.0
…                 …                …
19099                0.0               0.0
19100                0.0               0.0
19101                0.0               0.0
19102                0.0               0.0
19103                0.0               0.0

[19104 rows x 17 columns]
```

```
[ ]: df.drop("Dateofjoining",axis=1,inplace=True)
```

```
[ ]: df
```

```
[ ]:        Driver_ID   Age  Gender City  Education_Level  Income  \
     0              1  28.0     0.0  C23                2   57387
     1              1  28.0     0.0  C23                2   57387
     2              1  28.0     0.0  C23                2   57387
     3              2  31.0     0.0   C7                2   67016
     4              2  31.0     0.0   C7                2   67016
     …           …   …     …  …              …     …
     19099       2788  30.0     0.0  C27                2   70254
     19100       2788  30.0     0.0  C27                2   70254
     19101       2788  30.0     0.0  C27                2   70254
     19102       2788  30.0     0.0  C27                2   70254
     19103       2788  30.0     0.0  C27                2   70254

            Joining Designation  Grade  Total Business Value  Quarterly Rating  \
     0                        1      1               2381060                 2
```

```
1                        1      1                -665480                 2
2                        1      1                      0                 2
3                        2      2                      0                 1
4                        2      2                      0                 1
…                    …    …                    …               …
19099                    2      2                 740280                 3
19100                    2      2                 448370                 3
19101                    2      2                      0                 2
19102                    2      2                 200420                 2
19103                    2      2                 411480                 2

       Joining_day  Joining_month  Joining_year  LastWorking_day  \
0               24             12          2018              0.0
1               24             12          2018              0.0
2               24             12          2018             11.0
3                6             11          2020              0.0
4                6             11          2020              0.0
…            …            …           …               …
19099            8              6          2020              0.0
19100            8              6          2020              0.0
19101            8              6          2020              0.0
19102            8              6          2020              0.0
19103            8              6          2020              0.0

       LastWorking_month  LastWorking_year
0                    0.0               0.0
1                    0.0               0.0
2                    3.0            2019.0
3                    0.0               0.0
4                    0.0               0.0
…                  …               …
19099                0.0               0.0
19100                0.0               0.0
19101                0.0               0.0
19102                0.0               0.0
19103                0.0               0.0

[19104 rows x 16 columns]
```

[ ]:

```
```

[ ]:

```
df["Good_performers"] = df["Quarterly Rating"].apply (lambda  x: 1 if x >= 3 else 0)
```

[ ]:

```
df["Good_performers"]
```

```
[ ]: 0        0
     1        0
     2        0
     3        0
     4        0
             ..
     19099    1
     19100    1
     19101    0
     19102    0
     19103    0
     Name: Good_performers, Length: 19104, dtype: int64
```

```
[ ]: df
```

```
[ ]:        Driver_ID   Age  Gender City  Education_Level  Income  \
     0              1  28.0     0.0  C23                2   57387
     1              1  28.0     0.0  C23                2   57387
     2              1  28.0     0.0  C23                2   57387
     3              2  31.0     0.0   C7                2   67016
     4              2  31.0     0.0   C7                2   67016
     …            …    …       …   …                  …      …
     19099       2788  30.0     0.0  C27                2   70254
     19100       2788  30.0     0.0  C27                2   70254
     19101       2788  30.0     0.0  C27                2   70254
     19102       2788  30.0     0.0  C27                2   70254
     19103       2788  30.0     0.0  C27                2   70254

            Joining Designation  Grade  Total Business Value  Quarterly Rating  \
     0                        1      1               2381060                 2
     1                        1      1               -665480                 2
     2                        1      1                     0                 2
     3                        2      2                     0                 1
     4                        2      2                     0                 1
     …                      …    …                     …                 …
     19099                    2      2                740280                 3
     19100                    2      2                448370                 3
     19101                    2      2                     0                 2
     19102                    2      2                200420                 2
     19103                    2      2                411480                 2

            Joining_day  Joining_month  Joining_year  LastWorking_day  \
     0               24             12          2018              0.0
     1               24             12          2018              0.0
     2               24             12          2018             11.0
     3                6             11          2020              0.0
     4                6             11          2020              0.0
```

```
…            …            …          …               …
19099            8              6          2020              0.0
19100            8              6          2020              0.0
19101            8              6          2020              0.0
19102            8              6          2020              0.0
19103            8              6          2020              0.0

       LastWorking_month  LastWorking_year  Good_performers
0                    0.0               0.0                0
1                    0.0               0.0                0
2                    3.0            2019.0                0
3                    0.0               0.0                0
4                    0.0               0.0                0
…                  …               …              …
19099                0.0               0.0                1
19100                0.0               0.0                1
19101                0.0               0.0                0
19102                0.0               0.0                0
19103                0.0               0.0                0

[19104 rows x 17 columns]
```

[ ]:
```
df["Churn"] = df["LastWorking_day"].apply(lambda x:1 if x==1 else 0)
```

[ ]:
```
df["Churn"]
```

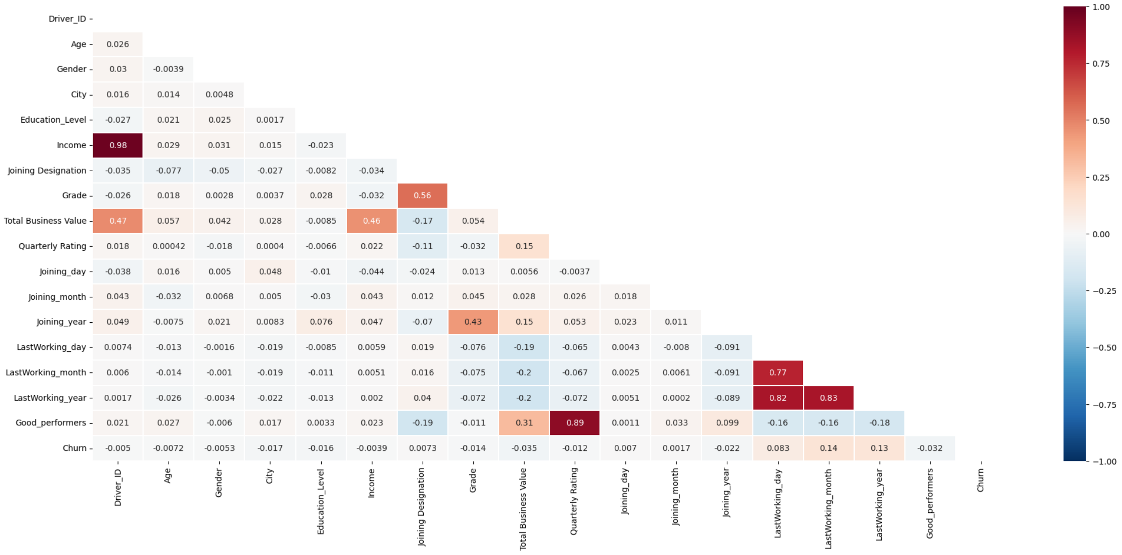
[ ]:
```
0        0
1        0
2        0
3        0
4        0
        ..
19099    0
19100    0
19101    0
19102    0
19103    0
Name: Churn, Length: 19104, dtype: int64
```

[ ]:
```
df["Churn"].unique()
```

[ ]:
```
array([0, 1])
```

[ ]:
```
plt.figure(figsize=(25, 10))

corr = df.apply(lambda x: pd.factorize(x)[0]).corr()
```

```python
mask = np.triu(np.ones_like(corr, dtype=bool))

ax = sns.heatmap(corr, mask=mask, xticklabels=corr.columns, yticklabels=corr.
 ↪columns, annot=True, linewidths=.2, cmap='RdBu_r', vmin=-1, vmax=1)
```



[ ]:
```python
# THERFORE GOOD PERFORMERS AND LAST WORKING DAY ARE MORE CORRELATED
```

[ ]:
```python
df.drop("City",axis=1,inplace=True)
```

[ ]:
```python
df
```

[ ]:
```
       Driver_ID   Age  Gender  Education_Level  Income  Joining Designation  \
0              1  28.0     0.0                2   57387                    1
1              1  28.0     0.0                2   57387                    1
2              1  28.0     0.0                2   57387                    1
3              2  31.0     0.0                2   67016                    2
4              2  31.0     0.0                2   67016                    2
…            …     …     …                …     …                    …
19099       2788  30.0     0.0                2   70254                    2
19100       2788  30.0     0.0                2   70254                    2
19101       2788  30.0     0.0                2   70254                    2
19102       2788  30.0     0.0                2   70254                    2
19103       2788  30.0     0.0                2   70254                    2

       Grade  Total Business Value  Quarterly Rating  Joining_day  \
0          1               2381060                 2           24
1          1               -665480                 2           24
2          1                     0                 2           24
```

```
3          2                     0                  1            6
4          2                     0                  1            6
…      …                   …               …          …
19099      2                740280                  3            8
19100      2                448370                  3            8
19101      2                     0                  2            8
19102      2                200420                  2            8
19103      2                411480                  2            8

       Joining_month  Joining_year  LastWorking_day  LastWorking_month  \
0                 12          2018              0.0                0.0
1                 12          2018              0.0                0.0
2                 12          2018             11.0                3.0
3                 11          2020              0.0                0.0
4                 11          2020              0.0                0.0
…              …           …              …                …
19099              6          2020              0.0                0.0
19100              6          2020              0.0                0.0
19101              6          2020              0.0                0.0
19102              6          2020              0.0                0.0
19103              6          2020              0.0                0.0

       LastWorking_year  Good_performers  Churn
0                   0.0                0      0
1                   0.0                0      0
2                2019.0                0      0
3                   0.0                0      0
4                   0.0                0      0
…                 …              …    …
19099               0.0                1      0
19100               0.0                1      0
19101               0.0                0      0
19102               0.0                0      0
19103               0.0                0      0

[19104 rows x 17 columns]
```

```
[ ]: df["Income"].min()
```

```
[ ]: 10747
```

```
[ ]: df["Income"].max()
```

```
[ ]: 188418
```

```
[ ]: df.sort_values(by=['Driver_ID', 'LastWorking_year'])
[ ]:        Driver_ID   Age  Gender  Education_Level  Income  Joining Designation  \
     0              1  28.0     0.0                2   57387                    1
     1              1  28.0     0.0                2   57387                    1
     2              1  28.0     0.0                2   57387                    1
     3              2  31.0     0.0                2   67016                    2
     4              2  31.0     0.0                2   67016                    2
     …            …   …     …                …     …                  …
     19099       2788  30.0     0.0                2   70254                    2
     19100       2788  30.0     0.0                2   70254                    2
     19101       2788  30.0     0.0                2   70254                    2
     19102       2788  30.0     0.0                2   70254                    2
     19103       2788  30.0     0.0                2   70254                    2

            Grade  Total Business Value  Quarterly Rating  Joining_day  \
     0          1               2381060                 2           24
     1          1               -665480                 2           24
     2          1                     0                 2           24
     3          2                     0                 1            6
     4          2                     0                 1            6
     …      …                   …                …           …
     19099      2                740280                 3            8
     19100      2                448370                 3            8
     19101      2                     0                 2            8
     19102      2                200420                 2            8
     19103      2                411480                 2            8

            Joining_month  Joining_year  LastWorking_day  LastWorking_month  \
     0                 12          2018              0.0                0.0
     1                 12          2018              0.0                0.0
     2                 12          2018             11.0                3.0
     3                 11          2020              0.0                0.0
     4                 11          2020              0.0                0.0
     …                …             …               …                 …
     19099              6          2020              0.0                0.0
     19100              6          2020              0.0                0.0
     19101              6          2020              0.0                0.0
     19102              6          2020              0.0                0.0
     19103              6          2020              0.0                0.0

            LastWorking_year  Good_performers  Churn
     0                   0.0                0      0
     1                   0.0                0      0
     2                2019.0                0      0
     3                   0.0                0      0
     4                   0.0                0      0
     …                 …              …    …
     19099               0.0                1      0
```

```
19100               0.0                1      0
19101               0.0                0      0
19102               0.0                0      0
19103               0.0                0      0

[19104 rows x 17 columns]
```

```
[ ]: df['income_change'] = df.groupby('Driver_ID')['Income'].diff()
```

```
[ ]: df['income_increased'] = df['income_change'] > 10000
```

```
[ ]: df['income_increased']
```

```
[ ]: 0        False
     1        False
     2        False
     3        False
     4        False
              …
     19099    False
     19100    False
     19101    False
     19102    False
     19103    False
     Name: income_increased, Length: 19104, dtype: bool
```

```
[ ]: # Sort by driver_id and period to ensure correct order
     df = df.sort_values(by=['Driver_ID', 'LastWorking_year'])

     # Calculate income difference for each driver
     df['income_change'] = df.groupby('Driver_ID')['Income'].diff()

     # Determine if income has increased
     df['income_increased'] = df['income_change'] > 0

     print(df)
```

```
       Driver_ID   Age  Gender  Education_Level  Income  Joining Designation  \
0              1  28.0     0.0                2   57387                    1
1              1  28.0     0.0                2   57387                    1
2              1  28.0     0.0                2   57387                    1
3              2  31.0     0.0                2   67016                    2
4              2  31.0     0.0                2   67016                    2
…          …   …     …              …     …                  …
19099       2788  30.0     0.0                2   70254                    2
19100       2788  30.0     0.0                2   70254                    2
19101       2788  30.0     0.0                2   70254                    2
```

```
19102       2788  30.0     0.0                2   70254                    2
19103       2788  30.0     0.0                2   70254                    2

       Grade  Total Business Value  Quarterly Rating  Joining_day  \
0          1               2381060                 2           24
1          1               -665480                 2           24
2          1                     0                 2           24
3          2                     0                 1            6
4          2                     0                 1            6
…      …                   …               …          …
19099      2                740280                 3            8
19100      2                448370                 3            8
19101      2                     0                 2            8
19102      2                200420                 2            8
19103      2                411480                 2            8

       Joining_month  Joining_year  LastWorking_day  LastWorking_month  \
0                 12          2018              0.0                0.0
1                 12          2018              0.0                0.0
2                 12          2018             11.0                3.0
3                 11          2020              0.0                0.0
4                 11          2020              0.0                0.0
…              …           …              …                …
19099              6          2020              0.0                0.0
19100              6          2020              0.0                0.0
19101              6          2020              0.0                0.0
19102              6          2020              0.0                0.0
19103              6          2020              0.0                0.0

       LastWorking_year  Good_performers  Churn  income_change  \
0                   0.0                0      0            NaN
1                   0.0                0      0            0.0
2                2019.0                0      0            0.0
3                   0.0                0      0            NaN
4                   0.0                0      0            0.0
…                 …              …    …            …
19099               0.0                1      0            0.0
19100               0.0                1      0            0.0
19101               0.0                0      0            0.0
19102               0.0                0      0            0.0
19103               0.0                0      0            0.0

       income_increased
0                 False
1                 False
2                 False
3                 False
4                 False
```

```
…                 …
19099             False
19100             False
19101             False
19102             False
19103             False

[19104 rows x 19 columns]
```

[ ]:

```
df['income_increased'] = df['income_increased'].astype(int)
```

[ ]:

```
df['income_increased'] = df['income_increased'].apply(lambda x: 1 if x == True␣
 ↪else 0)
```

[ ]:

```
df['income_increased'].unique()
```

[ ]:

```
array([0, 1])
```

[ ]:

```
df
```

[ ]:

```
       Driver_ID   Age  Gender  Education_Level  Income  Joining Designation  \
0              1  28.0     0.0                2   57387                    1
1              1  28.0     0.0                2   57387                    1
2              1  28.0     0.0                2   57387                    1
3              2  31.0     0.0                2   67016                    2
4              2  31.0     0.0                2   67016                    2
…          …   …     …              …     …                  …
19099       2788  30.0     0.0                2   70254                    2
19100       2788  30.0     0.0                2   70254                    2
19101       2788  30.0     0.0                2   70254                    2
19102       2788  30.0     0.0                2   70254                    2
19103       2788  30.0     0.0                2   70254                    2

       Grade  Total Business Value  Quarterly Rating  Joining_day  \
0          1               2381060                 2           24
1          1               -665480                 2           24
2          1                     0                 2           24
3          2                     0                 1            6
4          2                     0                 1            6
…      …                   …               …          …
19099      2                740280                 3            8
19100      2                448370                 3            8
19101      2                     0                 2            8
19102      2                200420                 2            8
19103      2                411480                 2            8

       Joining_month  Joining_year  LastWorking_day  LastWorking_month  \
0                 12          2018               0.0                0.0
1                 12          2018               0.0                0.0
2                 12          2018              11.0                3.0
3                 11          2020               0.0                0.0
4                 11          2020               0.0                0.0
…              …           …               …                …
19099              6          2020               0.0                0.0
19100              6          2020               0.0                0.0
19101              6          2020               0.0                0.0
19102              6          2020               0.0                0.0
19103              6          2020               0.0                0.0

       LastWorking_year  Good_performers  Churn  income_change  \
0                   0.0                0      0            NaN
1                   0.0                0      0            0.0
2                2019.0                0      0            0.0
3                   0.0                0      0            NaN
4                   0.0                0      0            0.0
…                  …              …    …            …
19099               0.0                1      0            0.0
19100               0.0                1      0            0.0
19101               0.0                0      0            0.0
19102               0.0                0      0            0.0
19103               0.0                0      0            0.0

       income_increased
0                     0
1                     0
2                     0
3                     0
4                     0
…                  …
19099                 0
19100                 0
19101                 0
19102                 0
19103                 0

[19104 rows x 19 columns]
```

[ ]:

```
Drivers_whose_income_increaded_counts = df['income_increased'].value_counts()

print(Drivers_whose_income_increaded_counts)
```

```
income_increased
0    19060
1       44
```

```
Name: count, dtype: int64
```

[ ]:
```
# THERFORE ONLY 44 DRIVERS INCOME HAS INCRESED
```

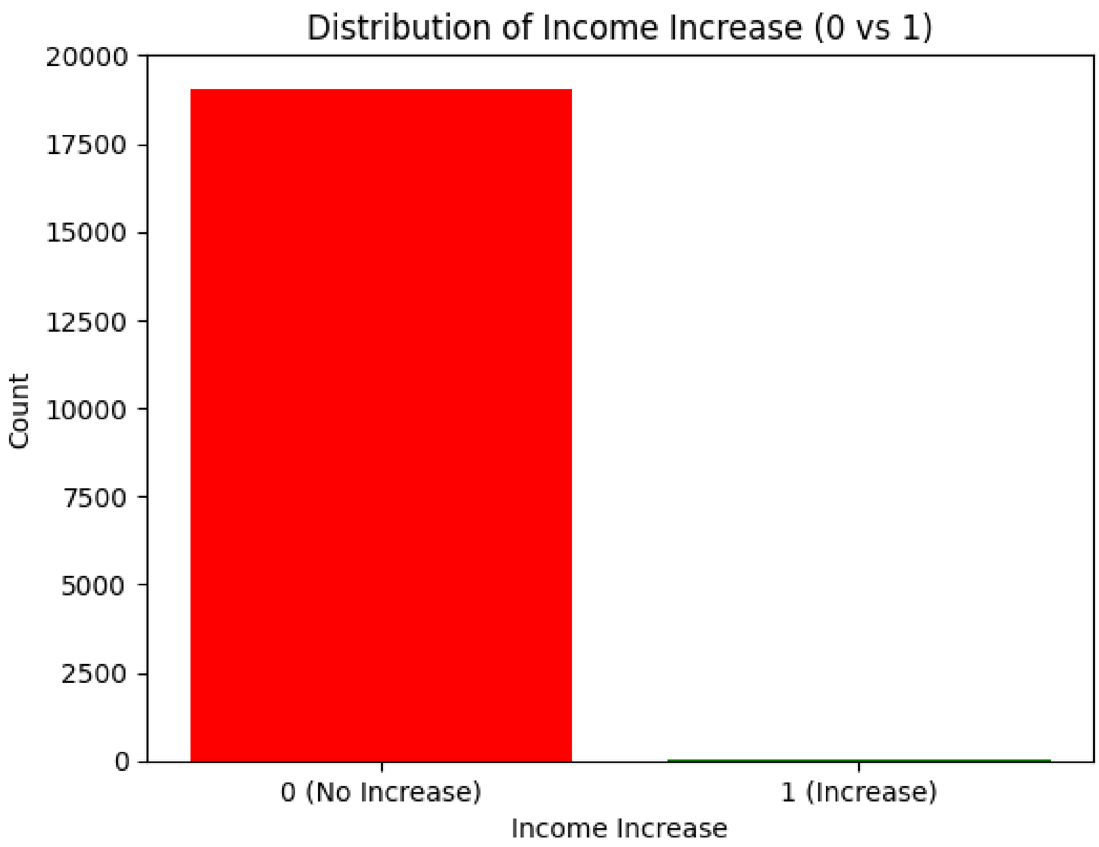
[ ]:
```

```

[ ]:
```
df
```

[ ]:
```
       Driver_ID   Age  Gender  Education_Level  Income  Joining Designation  \
0              1  28.0     0.0                2   57387                    1
1              1  28.0     0.0                2   57387                    1
2              1  28.0     0.0                2   57387                    1
3              2  31.0     0.0                2   67016                    2
4              2  31.0     0.0                2   67016                    2
…          …   …     …              …     …                  …
19099       2788  30.0     0.0                2   70254                    2
19100       2788  30.0     0.0                2   70254                    2
19101       2788  30.0     0.0                2   70254                    2
19102       2788  30.0     0.0                2   70254                    2
19103       2788  30.0     0.0                2   70254                    2

       Grade  Total Business Value  Quarterly Rating  Joining_day  \
0          1               2381060                 2           24
1          1               -665480                 2           24
2          1                     0                 2           24
3          2                     0                 1            6
4          2                     0                 1            6
…      …                   …               …          …
19099      2                740280                 3            8
19100      2                448370                 3            8
19101      2                     0                 2            8
19102      2                200420                 2            8
19103      2                411480                 2            8

       Joining_month  Joining_year  LastWorking_day  LastWorking_month  \
0                 12          2018              0.0                0.0
1                 12          2018              0.0                0.0
2                 12          2018             11.0                3.0
3                 11          2020              0.0                0.0
4                 11          2020              0.0                0.0
…              …           …              …                …
19099              6          2020              0.0                0.0
19100              6          2020              0.0                0.0
19101              6          2020              0.0                0.0
19102              6          2020              0.0                0.0
19103              6          2020              0.0                0.0
```

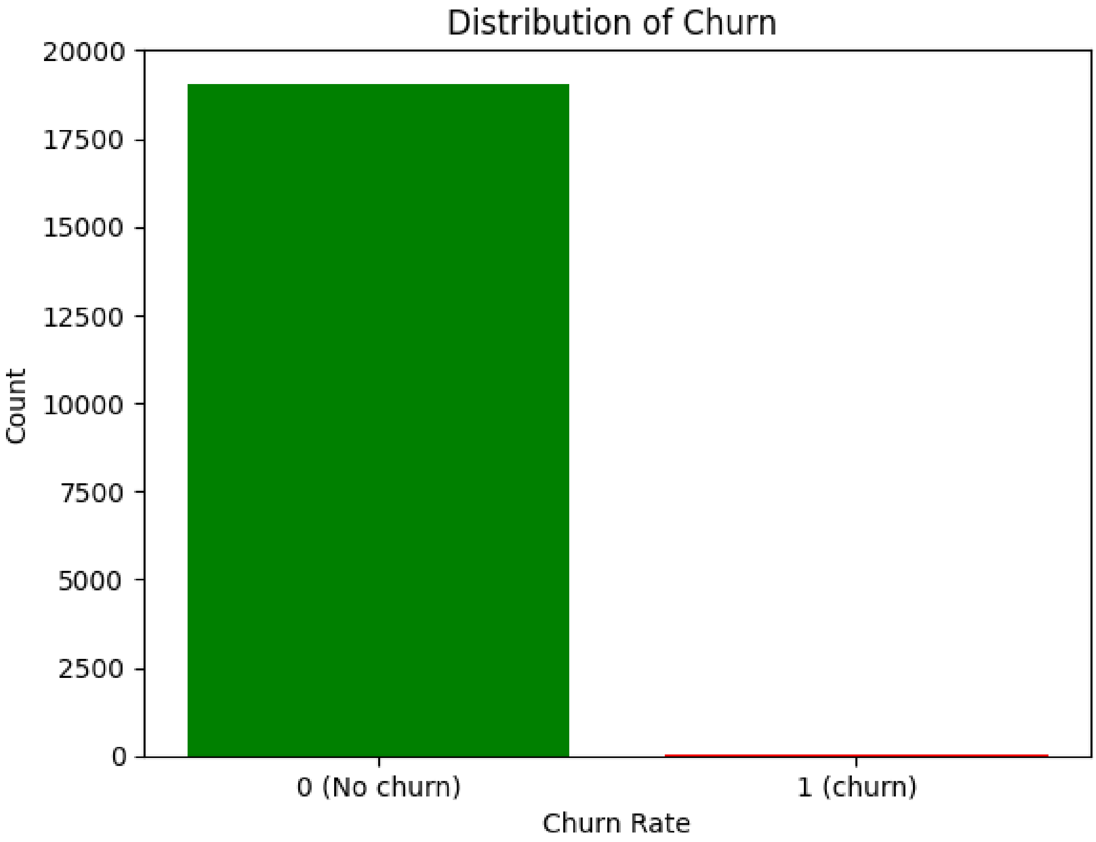
```
       LastWorking_year  Good_performers  Churn  income_change  \
0                   0.0                0      0            NaN
1                   0.0                0      0            0.0
2                2019.0                0      0            0.0
3                   0.0                0      0            NaN
4                   0.0                0      0            0.0
…                 …              …    …            …
19099               0.0                1      0            0.0
19100               0.0                1      0            0.0
19101               0.0                0      0            0.0
19102               0.0                0      0            0.0
19103               0.0                0      0            0.0

       income_increased
0                     0
1                     0
2                     0
3                     0
4                     0
…                 …
19099                 0
19100                 0
19101                 0
19102                 0
19103                 0

[19104 rows x 19 columns]
```

```
counts = Drivers_whose_income_increaded_counts
plt.bar(counts.index, counts.values, color=['red', 'green'])
plt.xticks([0, 1], ['0 (No Increase)', '1 (Increase)'])
plt.xlabel('Income Increase')
plt.ylabel('Count')
plt.title('Distribution of Income Increase (0 vs 1)')
plt.show()
```



[ ]:

```
plt.bar(counts.index, counts.values, color=['green', 'red'])
plt.xticks([0, 1], ['0 (No churn)', '1 (churn)'])
plt.xlabel('Churn Rate')
plt.ylabel('Count')
plt.title('Distribution of Churn ')
plt.show()
```



```
[ ]: df.columns
```

```
[ ]: Index(['Driver_ID', 'Age', 'Gender', 'Education_Level', 'Income',
            'Joining Designation', 'Grade', 'Total Business Value',
            'Quarterly Rating', 'Joining_day', 'Joining_month', 'Joining_year',
            'LastWorking_day', 'LastWorking_month', 'LastWorking_year',
            'Good_performers', 'Churn', 'income_change', 'income_increased'],
           dtype='object')
```

```
[ ]: df["Churn"].value_counts()
```

```
[ ]: Churn
     0    19060
     1       44
     Name: count, dtype: int64
```

```
[ ]: df["Total Business Value"]
```

```
[ ]: 0         2381060
     1         -665480
```

```
2              0
3              0
4              0
              …
19099     740280
19100     448370
19101          0
19102     200420
19103     411480
Name: Total Business Value, Length: 19104, dtype: int64
```

[ ]:

```
df["Drivers_Business_Value_acquired"]= df["Total Business Value"].apply(lambda x:1 if x>1 else 0)
```

[ ]:

```
df["Drivers_Business_Value_acquired"]
```

[ ]:

```
0        1
1        0
2        0
3        0
4        0
        ..
19099    1
19100    1
19101    0
19102    1
19103    1
Name: Drivers_Business_Value_acquired, Length: 19104, dtype: int64
```

[ ]:

```
df["Drivers_Business_Value_acquired"].value_counts()
```

[ ]:

```
Drivers_Business_Value_acquired
1    12456
0     6648
Name: count, dtype: int64
```

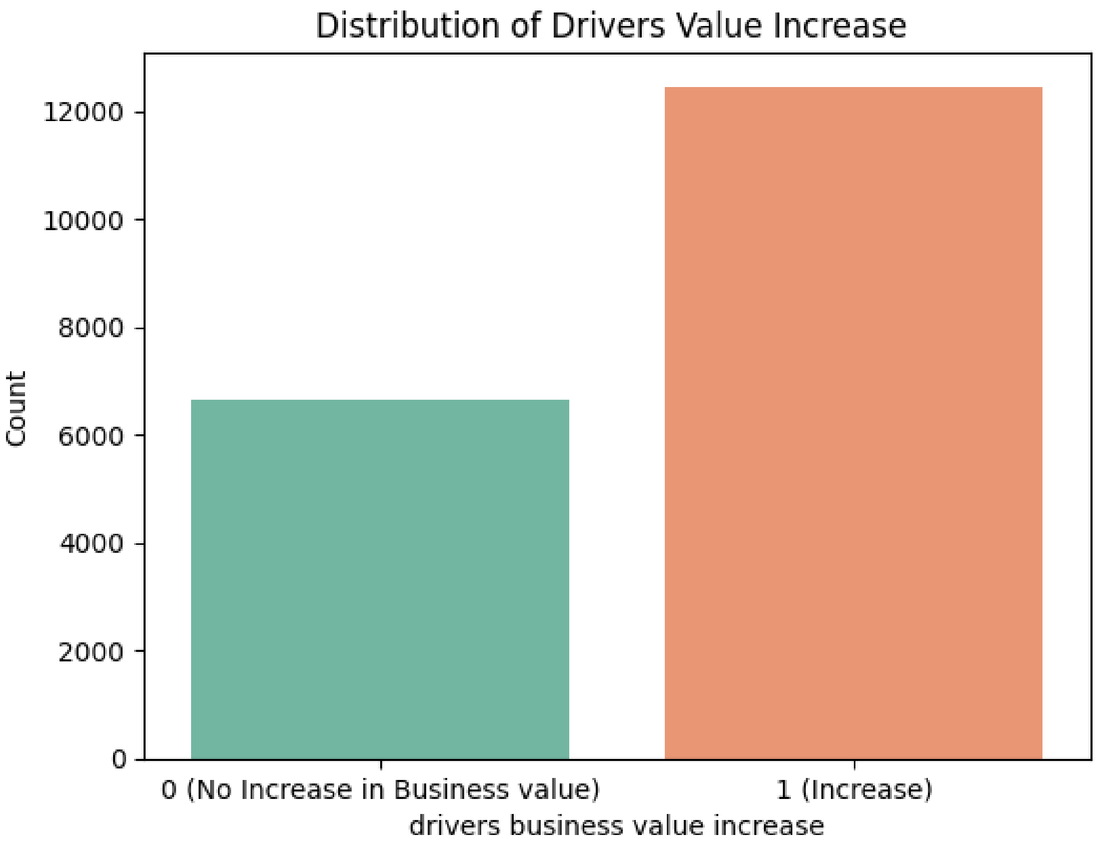
[ ]:

```
sns.countplot(x='Drivers_Business_Value_acquired', data=df, palette='Set2')
plt.xticks([0, 1], ['0 (No Increase in Business value)', '1 (Increase)'])
plt.xlabel('drivers business value increase')
plt.ylabel('Count')
plt.title('Distribution of Drivers Value Increase ')
plt.show()
```

```
<ipython-input-178-a2adf5a0a416>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
```

```
effect.
```

```
sns.countplot(x='Drivers_Business_Value_acquired', data=df, palette='Set2')
```



```
[ ]: df
```

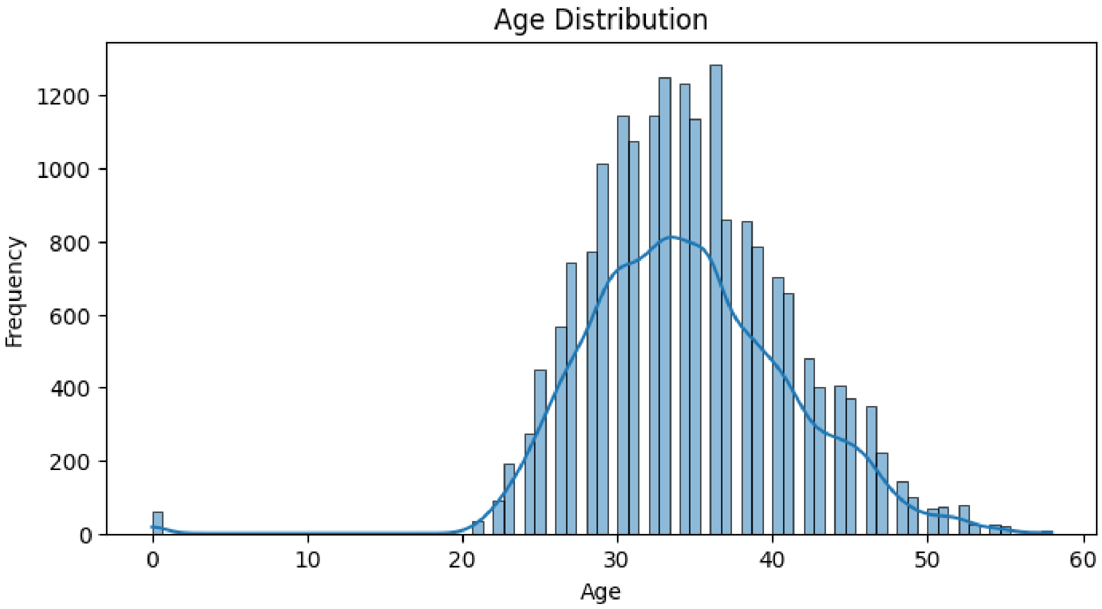
```
[ ]:       Driver_ID   Age  Gender  Education_Level  Income  Joining Designation  \
     0             1  28.0     0.0                2   57387                    1
     1             1  28.0     0.0                2   57387                    1
     2             1  28.0     0.0                2   57387                    1
     3             2  31.0     0.0                2   67016                    2
     4             2  31.0     0.0                2   67016                    2
     …          …   …     …              …     …                  …
     19099      2788  30.0     0.0                2   70254                    2
     19100      2788  30.0     0.0                2   70254                    2
     19101      2788  30.0     0.0                2   70254                    2
     19102      2788  30.0     0.0                2   70254                    2
     19103      2788  30.0     0.0                2   70254                    2

              Grade  Total Business Value  Quarterly Rating  Joining_day  \
     0          1               2381060                 2            24
     1          1               -665480                 2            24
     2          1                     0                 2            24
     3          2                     0                 1             6
     4          2                     0                 1             6
     …      …                   …               …          …
     19099      2                740280                 3             8
     19100      2                448370                 3             8
     19101      2                     0                 2             8
     19102      2                200420                 2             8
     19103      2                411480                 2             8

            Joining_month  Joining_year  LastWorking_day  LastWorking_month  \
     0                 12          2018              0.0                0.0
     1                 12          2018              0.0                0.0
     2                 12          2018             11.0                3.0
     3                 11          2020              0.0                0.0
     4                 11          2020              0.0                0.0
     …              …           …              …                …
     19099              6          2020              0.0                0.0
     19100              6          2020              0.0                0.0
     19101              6          2020              0.0                0.0
     19102              6          2020              0.0                0.0
     19103              6          2020              0.0                0.0

            LastWorking_year  Good_performers  Churn  income_change  \
     0                   0.0                0      0            NaN
     1                   0.0                0      0            0.0
     2                2019.0                0      0            0.0
     3                   0.0                0      0            NaN
     4                   0.0                0      0            0.0
     …                 …              …    …            …
     19099               0.0                1      0            0.0
     19100               0.0                1      0            0.0
     19101               0.0                0      0            0.0
     19102               0.0                0      0            0.0
     19103               0.0                0      0            0.0

            income_increased  Drivers_Business_Value_acquired
     0                     0                                1
     1                     0                                0
     2                     0                                0
     3                     0                                0
     4                     0                                0
     …                 …                              …
     19099                 0                                1
     19100                 0                                1
```

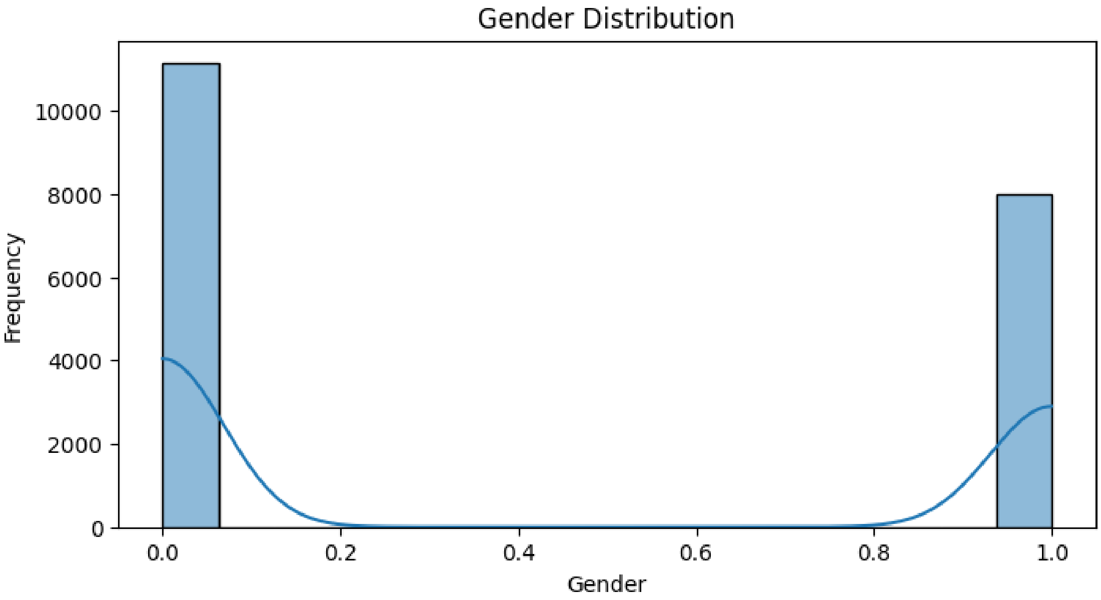
```
19101                     0                                 0
19102                     0                                 1
19103                     0                                 1

[19104 rows x 20 columns]
```

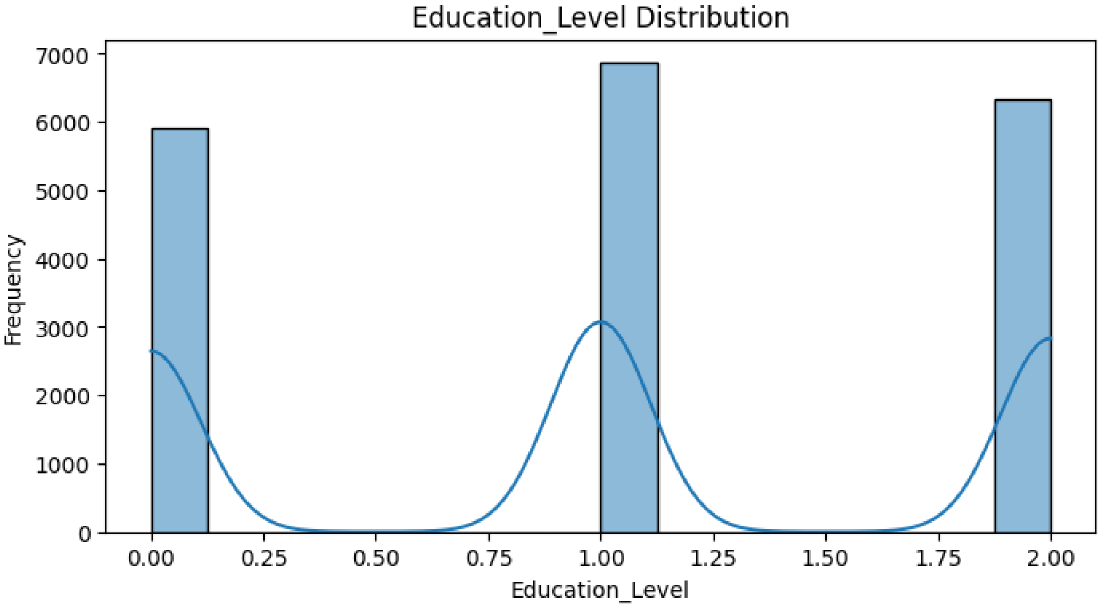
```
[ ]: df.drop("Driver_ID",axis=1,inplace=True)
```

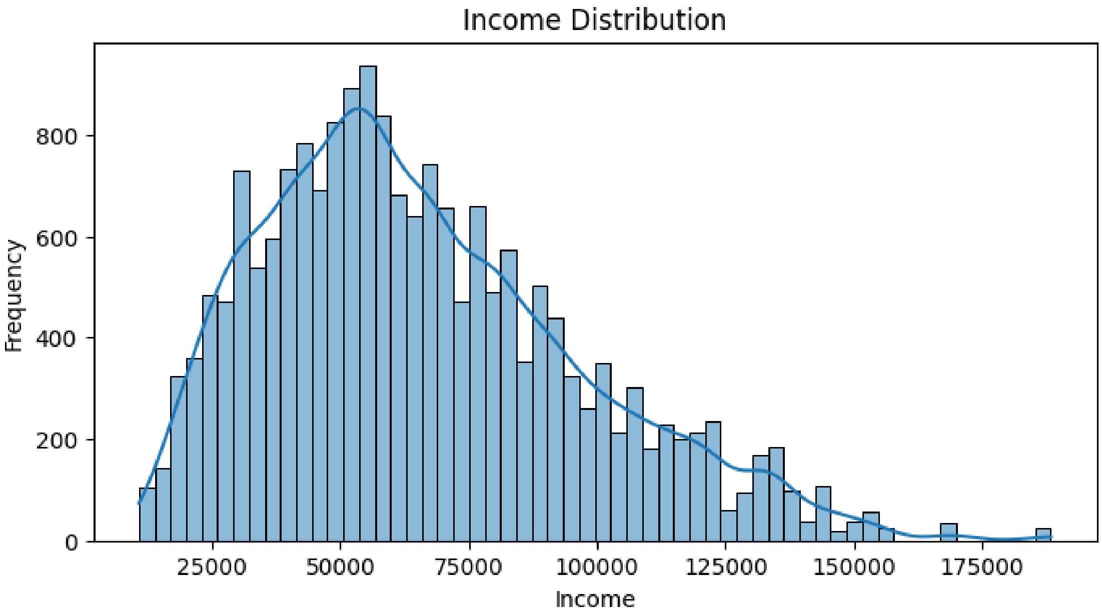
```
[ ]: # UNIVARITAE ANALYSIS
```

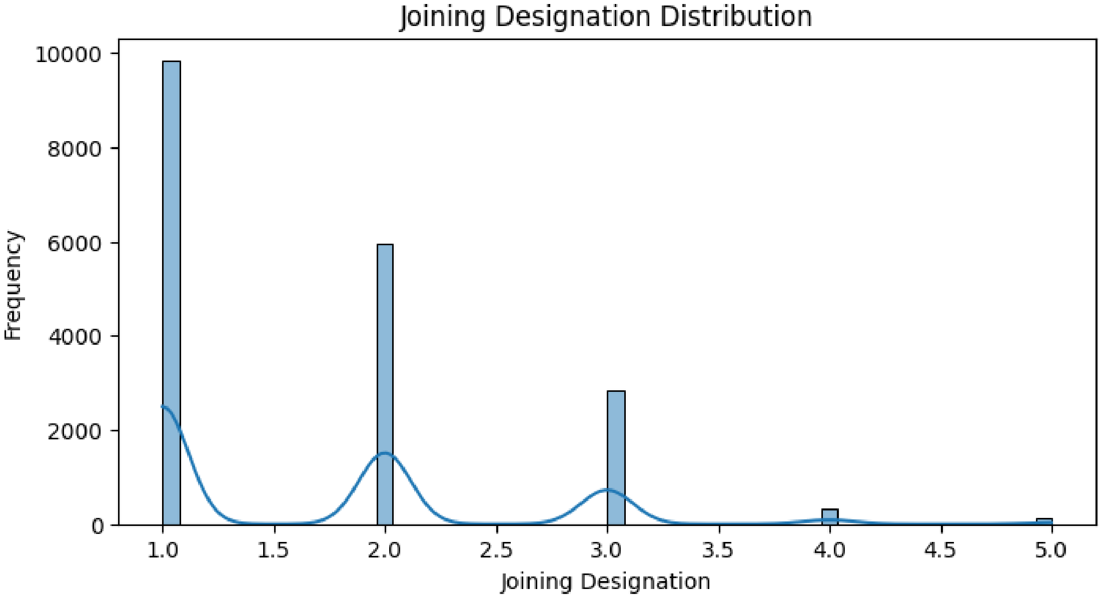
```
[ ]: for i in df.columns:
         plt.figure(figsize=(8, 4))

         # Plot the histogram for the actual data, not just the column name
         sns.histplot(df[i], kde=True)  # kde=True adds the Kernel Density Estimate

         # Set dynamic title, xlabel, and ylabel based on the column being plotted
         plt.title(f'{i} Distribution')
         plt.xlabel(i)
         plt.ylabel('Frequency')

         # Show the plot
         plt.show()
```

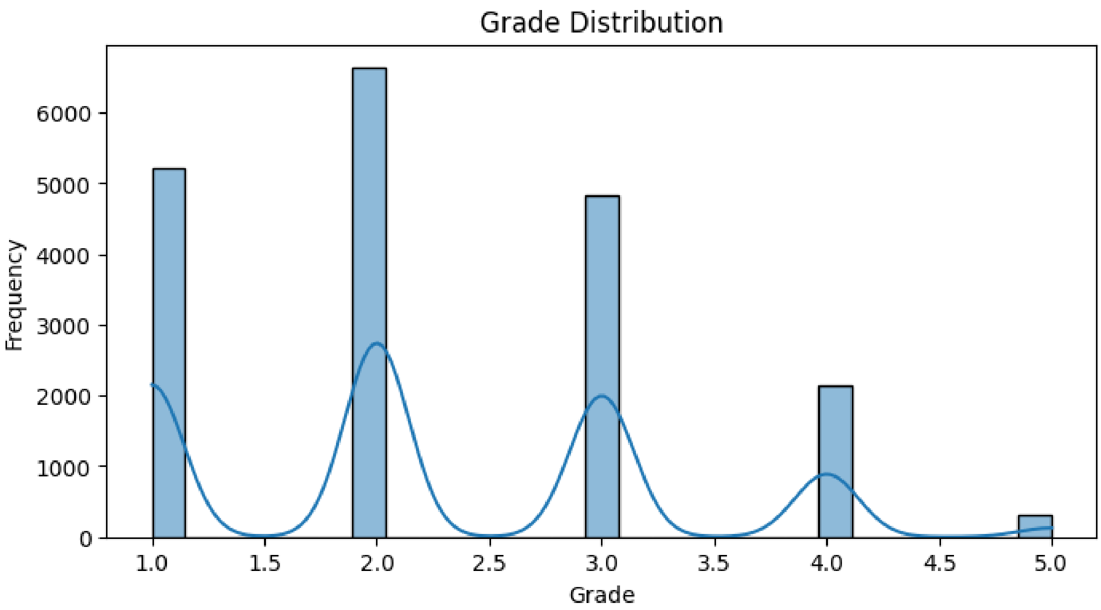


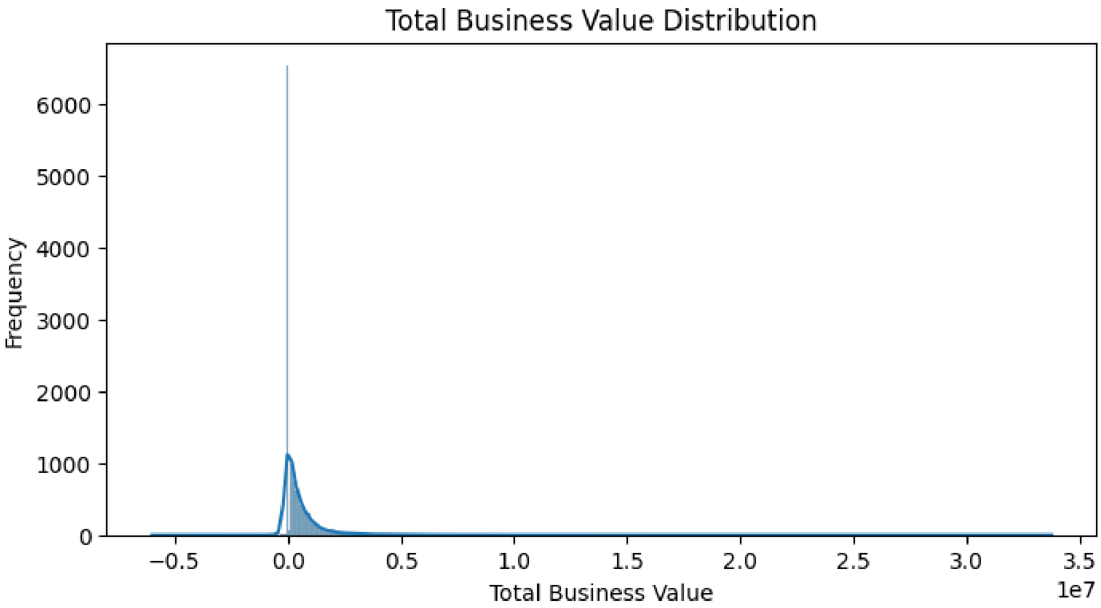

Gender Distribution
Frequency
0
2000
4000
6000
8000
10000
0.0
0.2
0.4
0.6
0.8
1.0
Gender

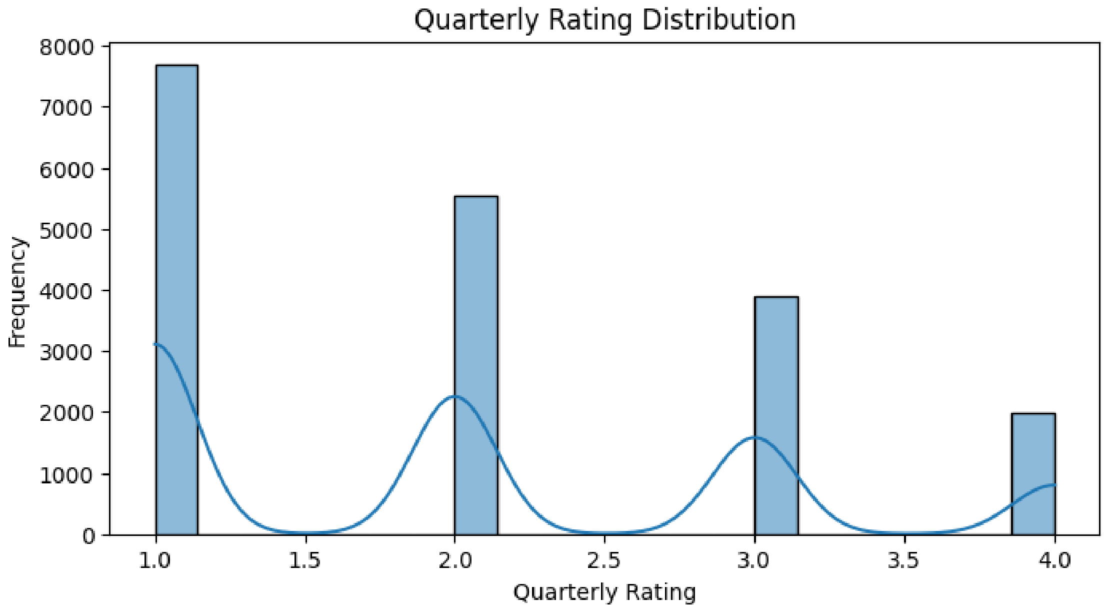

Education_Level Distribution
Frequency
0
1000
2000
3000
4000
5000
6000
7000
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Education_Level

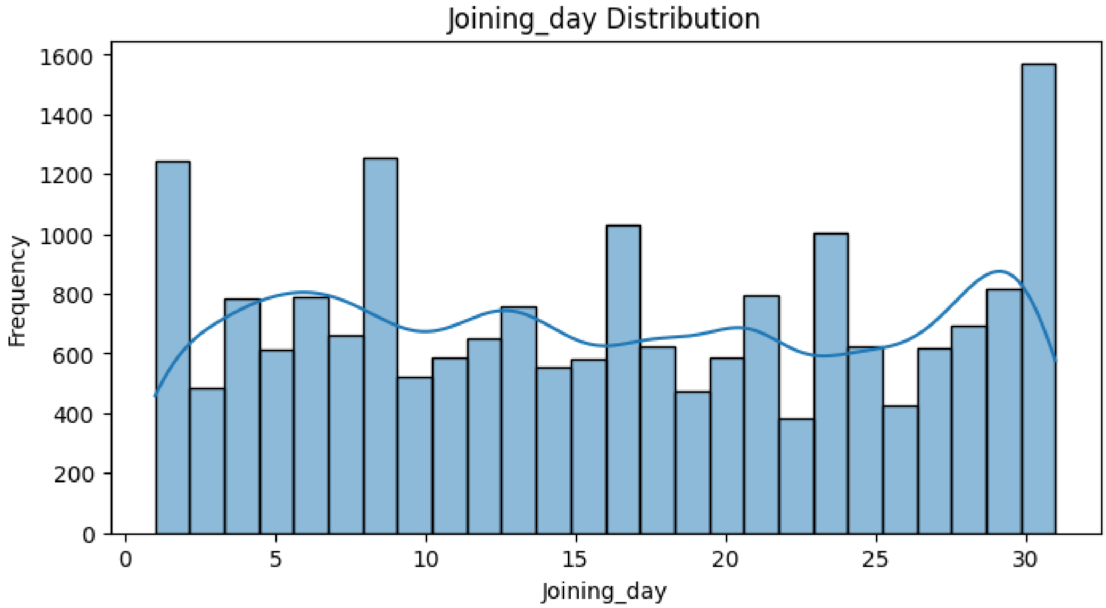

Income Distribution
Frequency
0
200
400
600
800
25000
50000
75000
100000
125000
150000
175000
Income

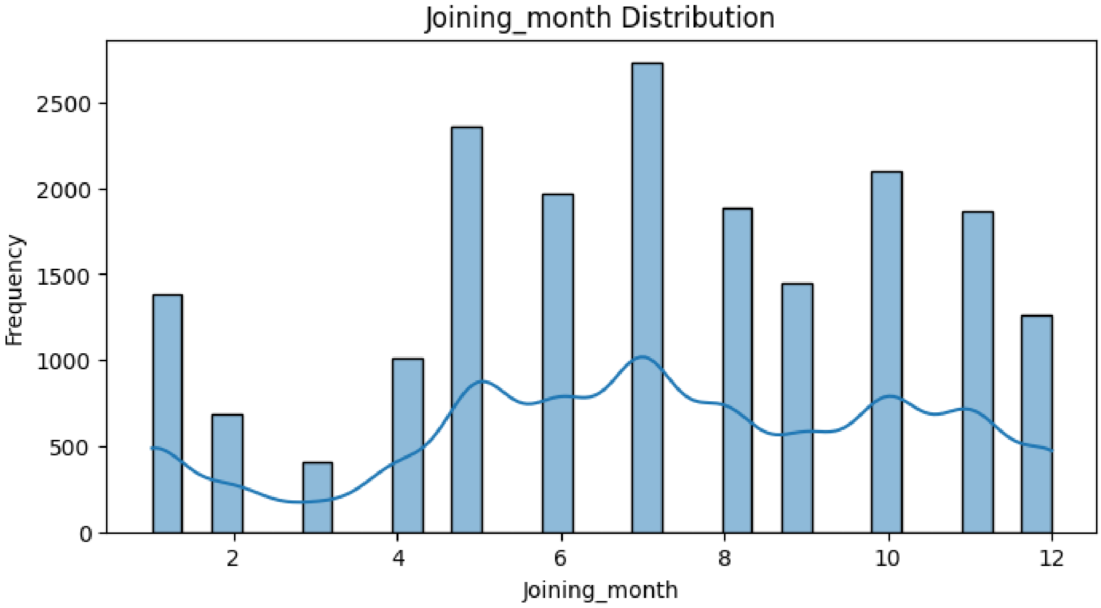

Joining Designation Distribution
Frequency
0
2000
4000
6000
8000
10000
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Joining Designation

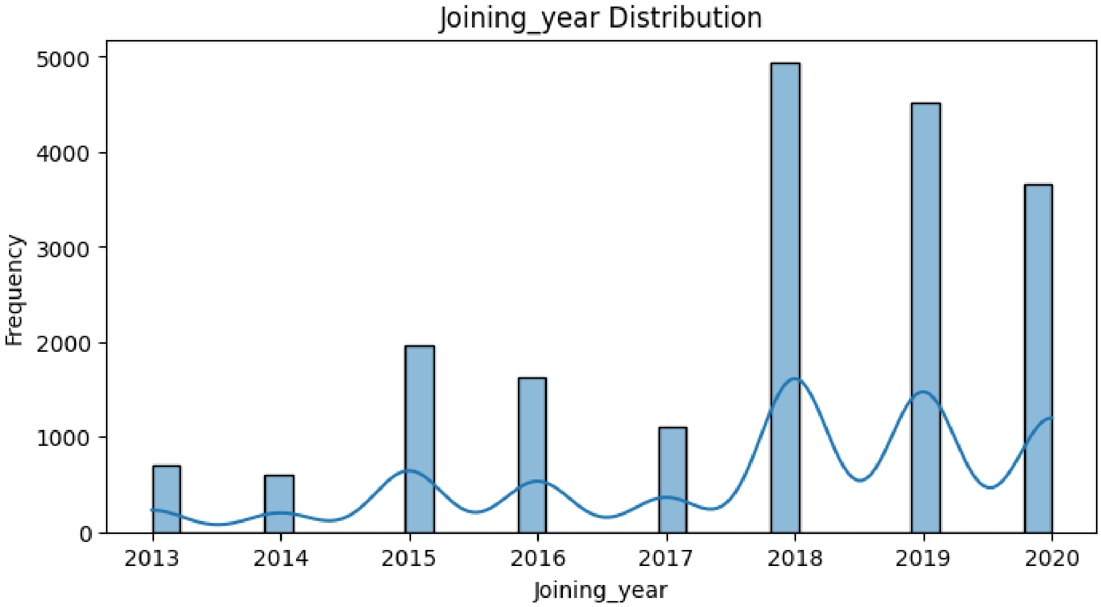

Grade Distribution
Frequency
0
1000
2000
3000
4000
5000
6000
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Grade

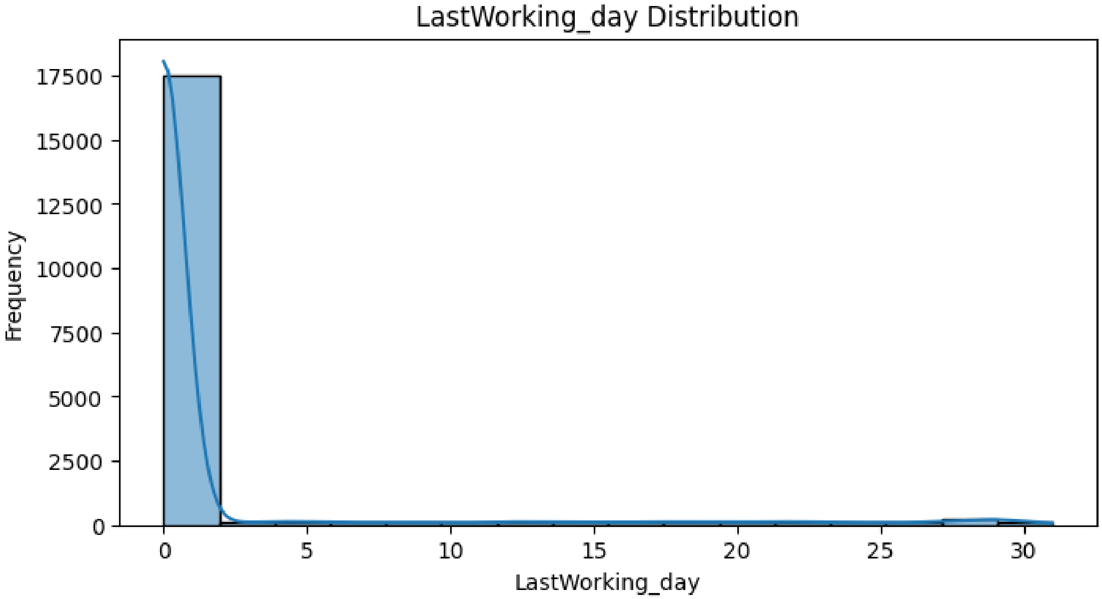

Total Business Value Distribution
Frequency
0
1000
2000
3000
4000
5000
6000
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1e7
Total Business Value

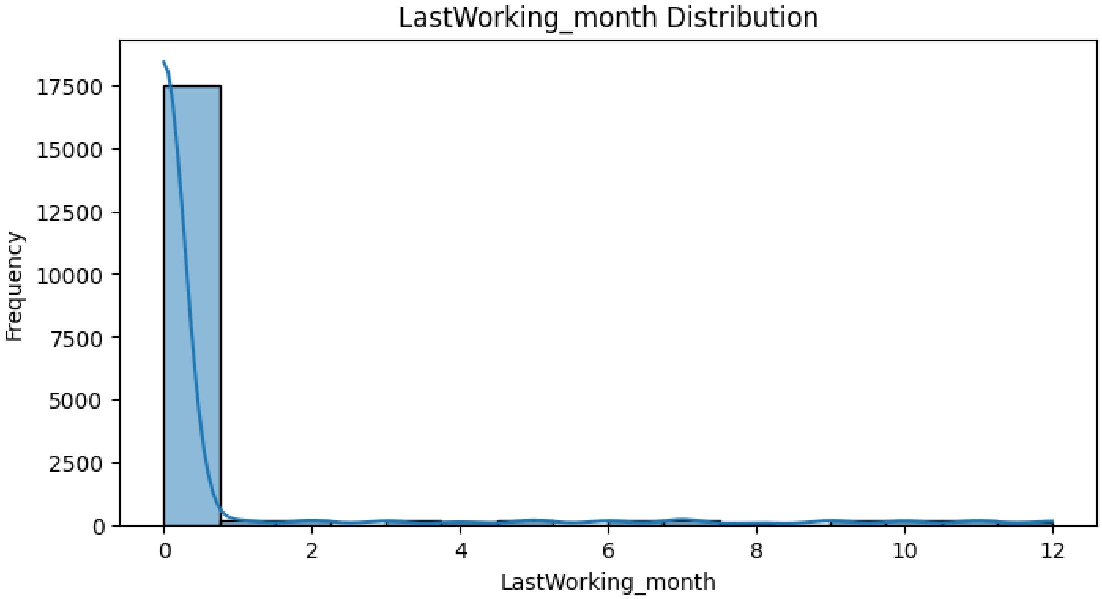

Quarterly Rating Distribution
Frequency
0
1000
2000
3000
4000
5000
6000
7000
8000
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Quarterly Rating

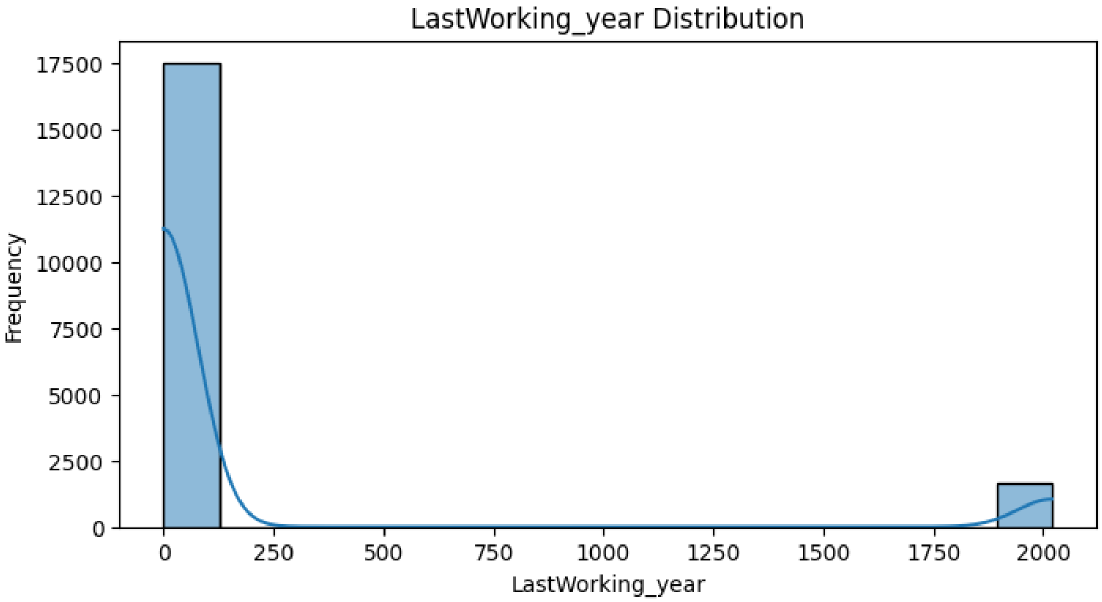

Joining_day Distribution
Frequency
0
200
400
600
800
1000
1200
1400
1600
0
5
10
15
20
25
30
Joining_day

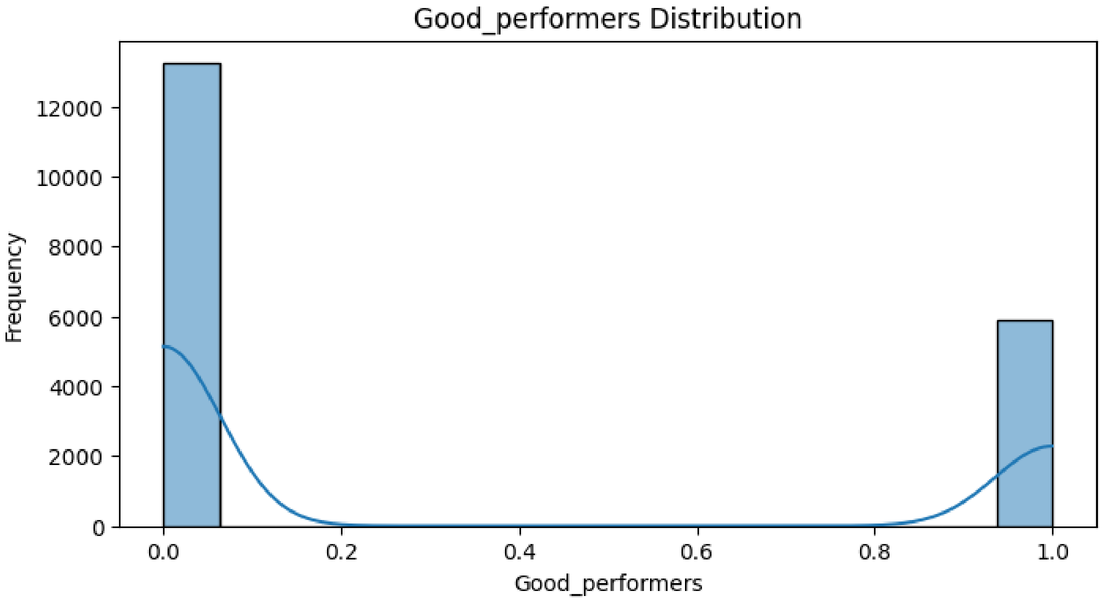

Joining_month Distribution
Frequency
0
500
1000
1500
2000
2500
2
4
6
8
10
12
Joining_month

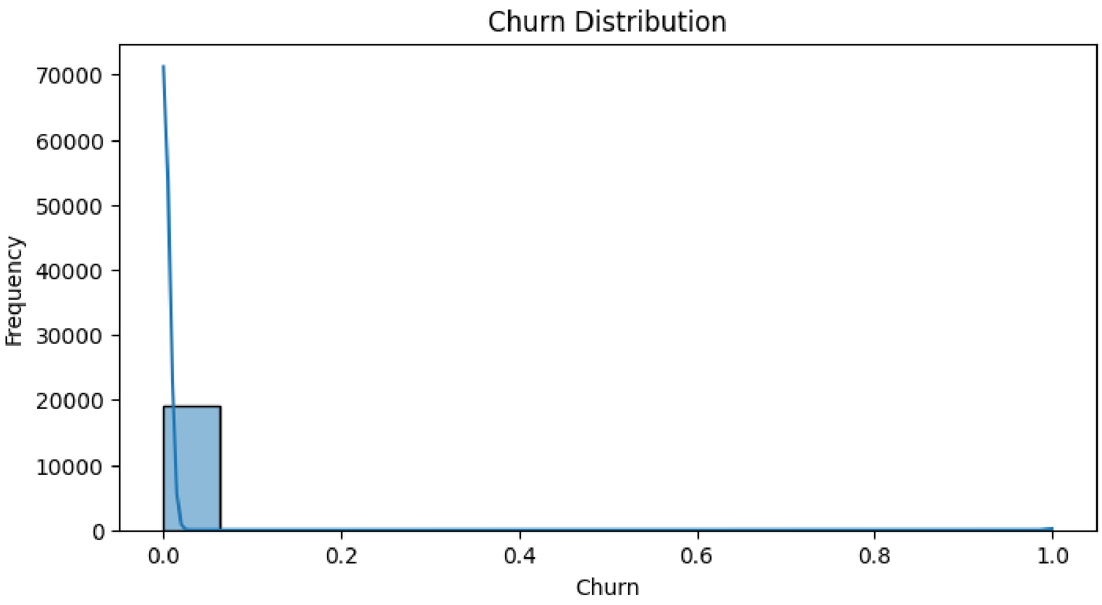

Joining_year Distribution
Frequency
0
1000
2000
3000
4000
5000
2013
2014
2015
2016
2017
2018
2019
2020
Joining_year

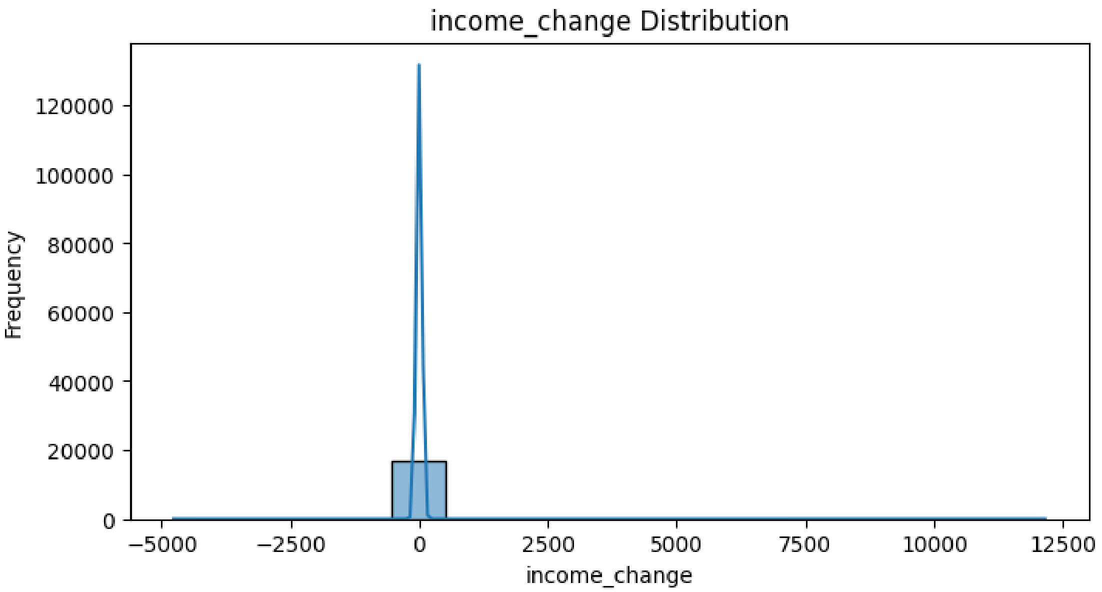

LastWorking_day Distribution
Frequency
17500
15000
12500
10000
7500
5000
2500
0
0
5
10
15
20
25
30
LastWorking_day

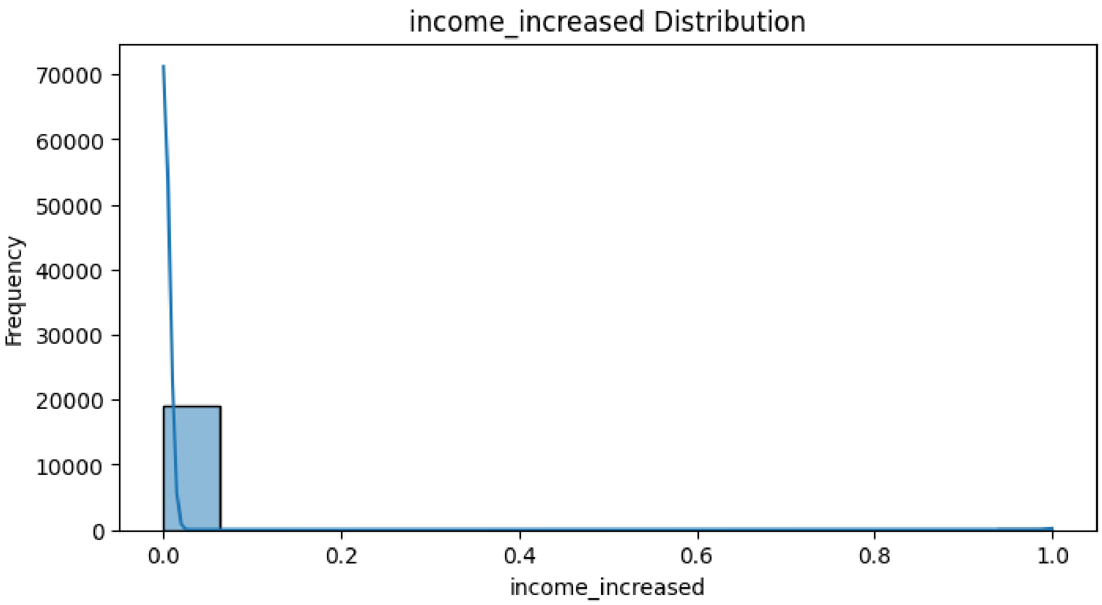

LastWorking_month Distribution
Frequency
17500
15000
12500
10000
7500
5000
2500
0
0
2
4
6
8
10
12
LastWorking_month

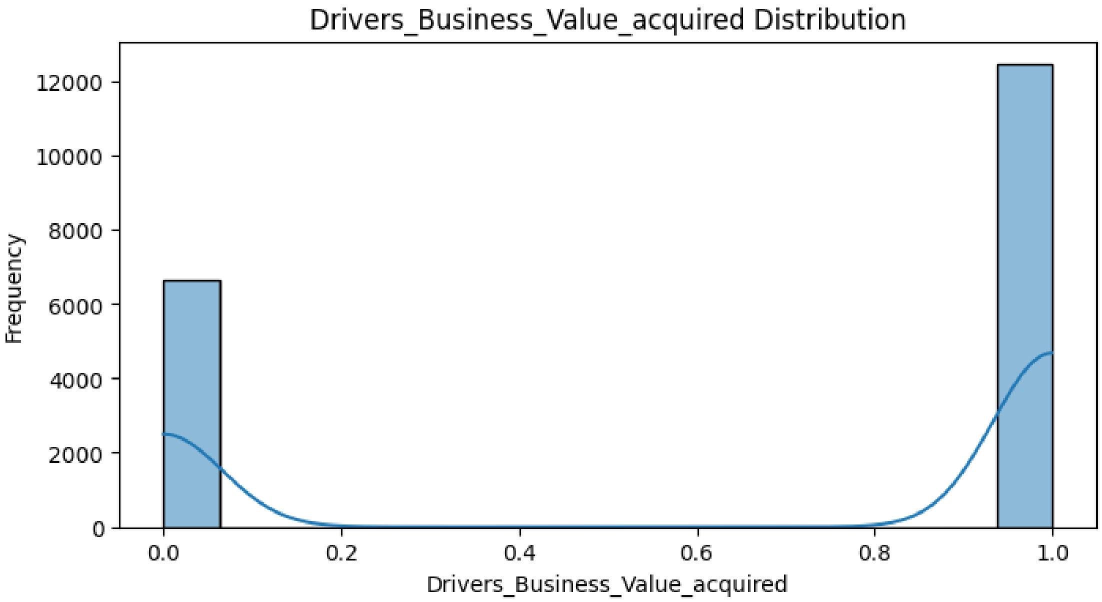

LastWorking_year Distribution
Frequency
17500
15000
12500
10000
7500
5000
2500
0
0
250
500
750
1000
1250
1500
1750
2000
LastWorking_year


Good_performers Distribution
Frequency
12000
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
Good_performers


Churn Distribution
Frequency
70000
60000
50000
40000
30000
20000
10000
0
0.0
0.2
0.4
0.6
0.8
1.0
Churn


income_change Distribution
Frequency
120000
100000
80000
60000
40000
20000
0
−5000
−2500
0
2500
5000
7500
10000
12500
income_change


income_increased Distribution
Frequency
70000
60000
50000
40000
30000
20000
10000
0
0.0
0.2
0.4
0.6
0.8
1.0
income_increased

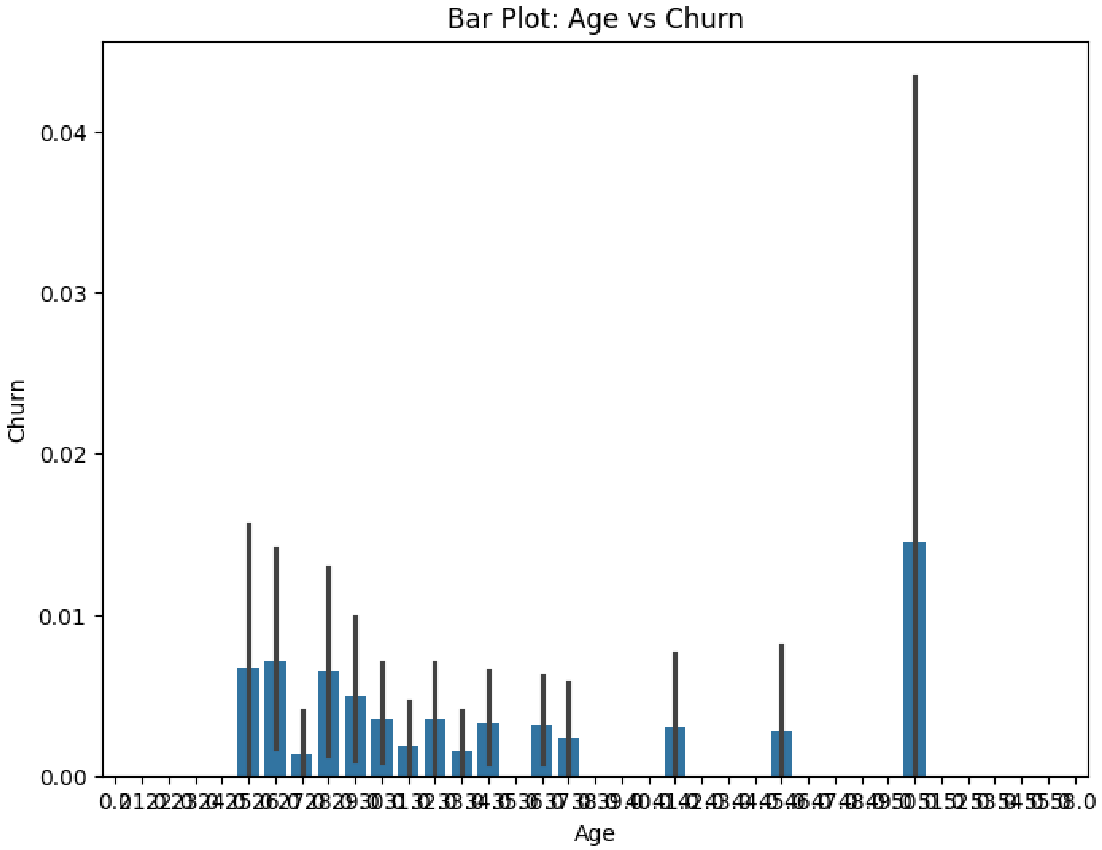

Drivers_Business_Value_acquired Distribution
Frequency
12000
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
Drivers_Business_Value_acquired

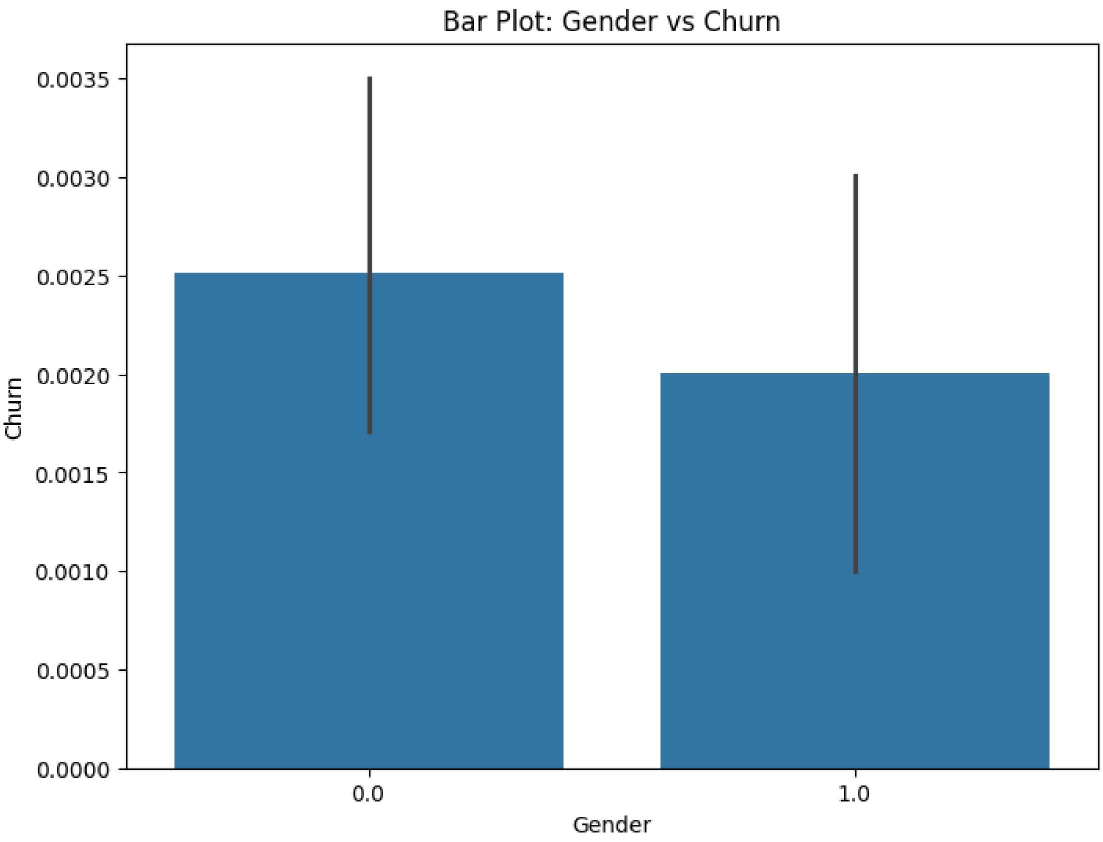
[ ]:

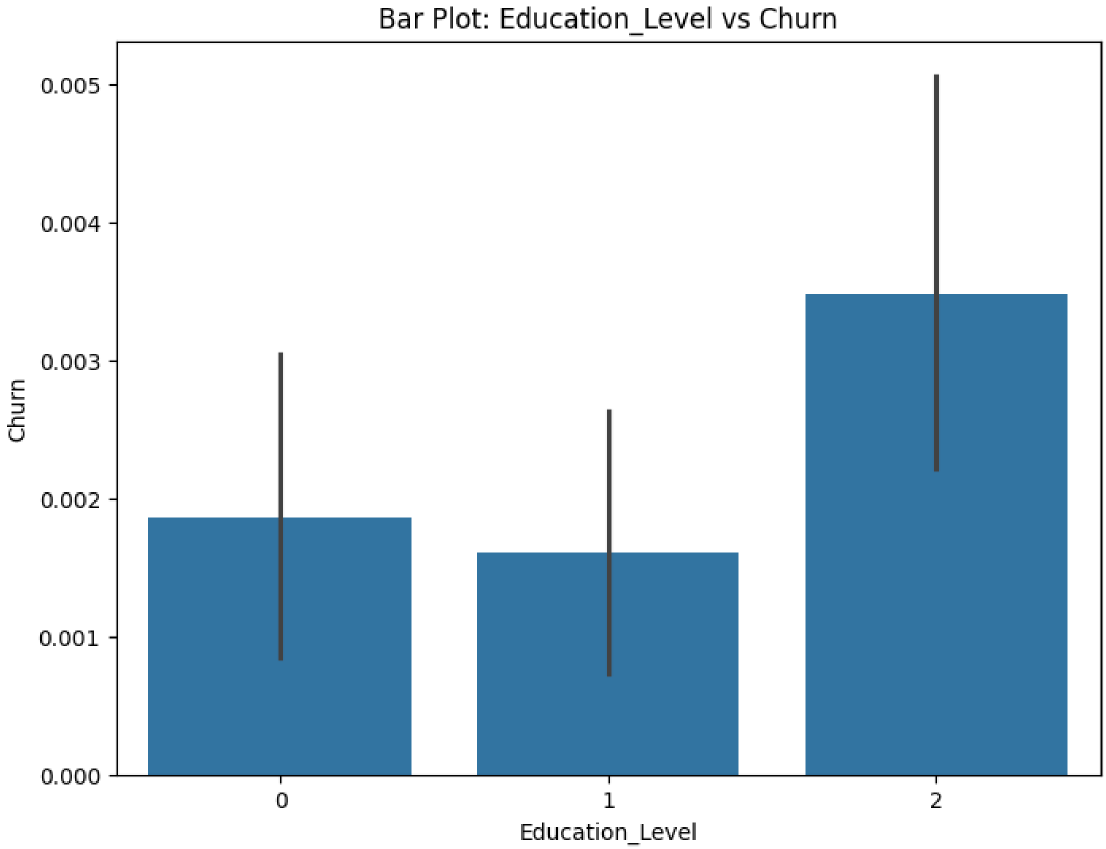
```

```

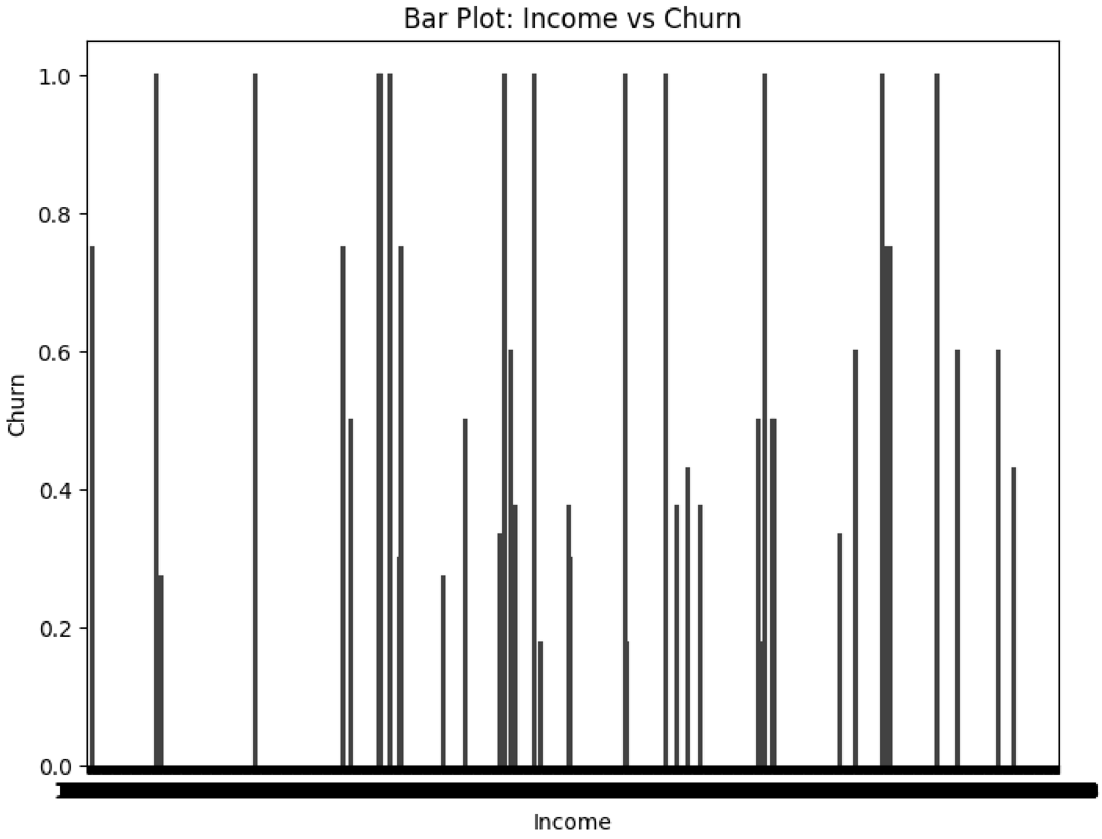
1 Bivariate Analysis

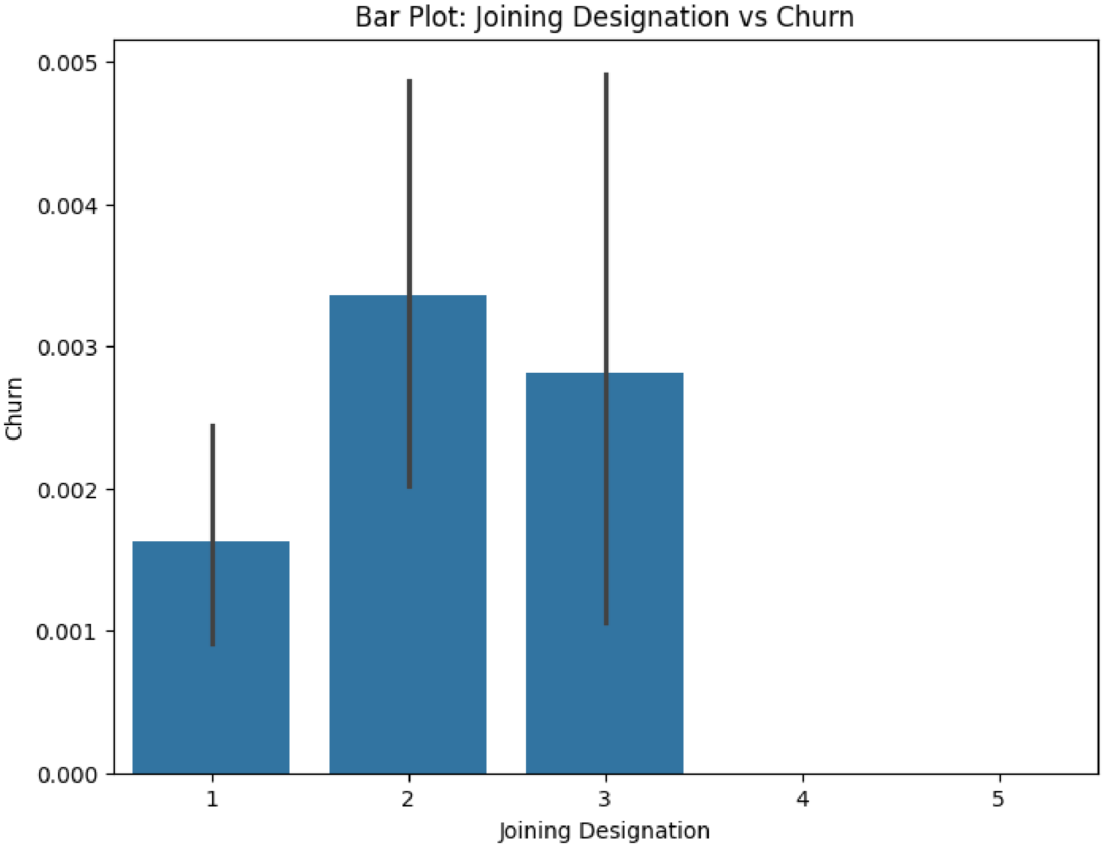
```
[ ]: # Define your target column (which will always be on one axis)
     target_column = "Churn"  # Changed to the column name

     # Loop through numerical columns, skipping the target column
     for i in df:
         if i != target_column:  # Avoid plotting the target column against itself
             plt.figure(figsize=(8, 6))

             # Bar plot comparing other columns to the target column
             sns.barplot(x=df[i], y=df[target_column])
             plt.title(f'Bar Plot: {i} vs {target_column}')
             plt.xlabel(i)
             plt.ylabel(target_column)

             # Display the plot
             plt.show()
```

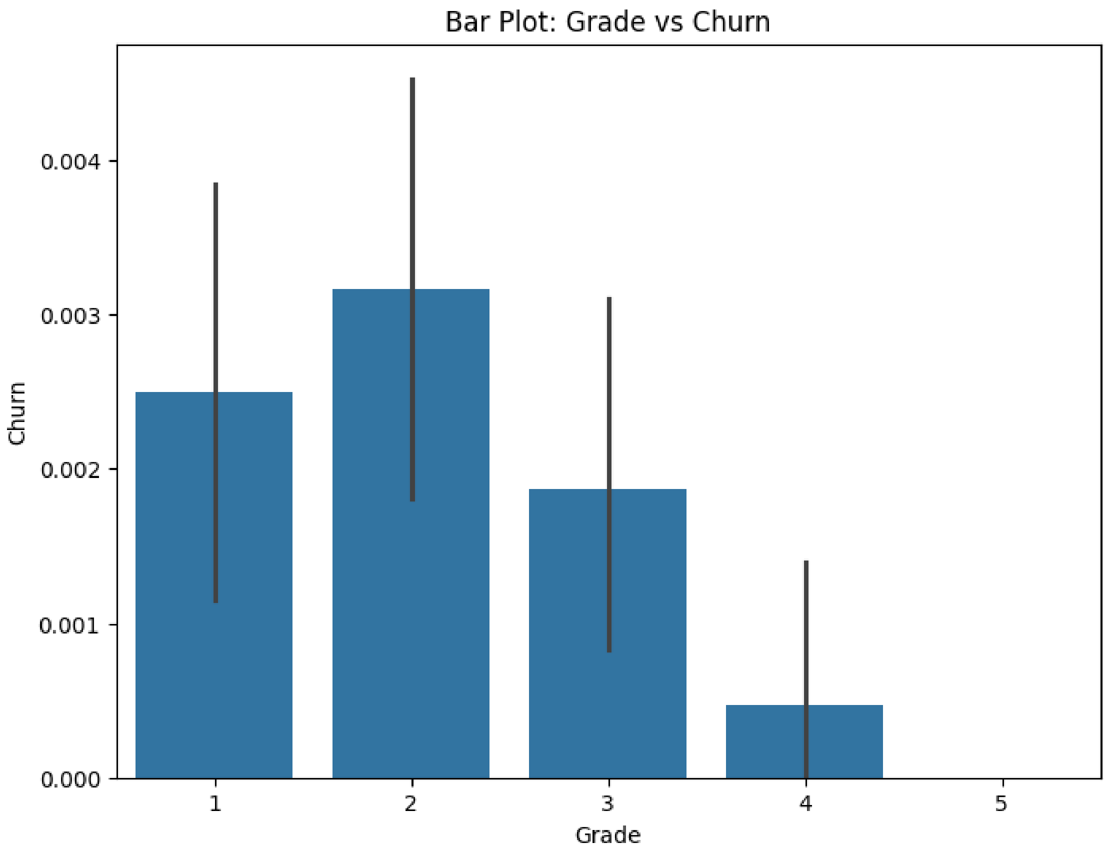


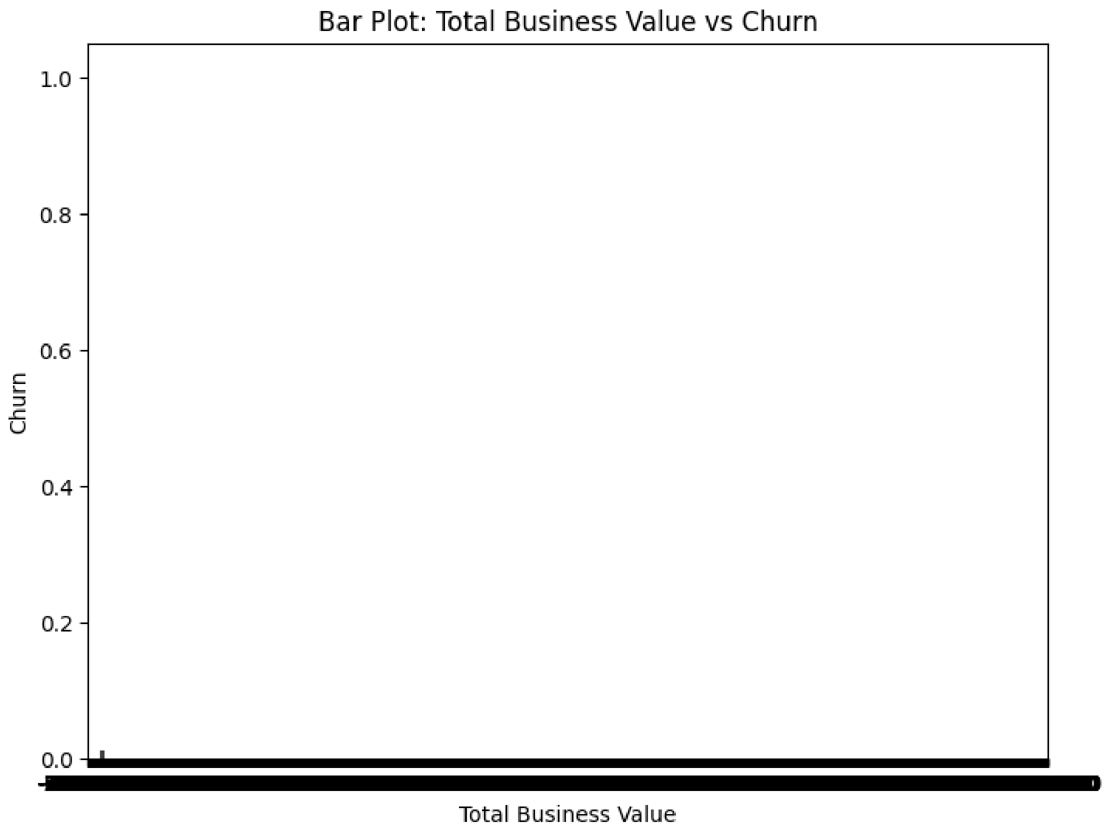

Bar Plot: Gender vs Churn
Churn
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0.0
1.0
Gender

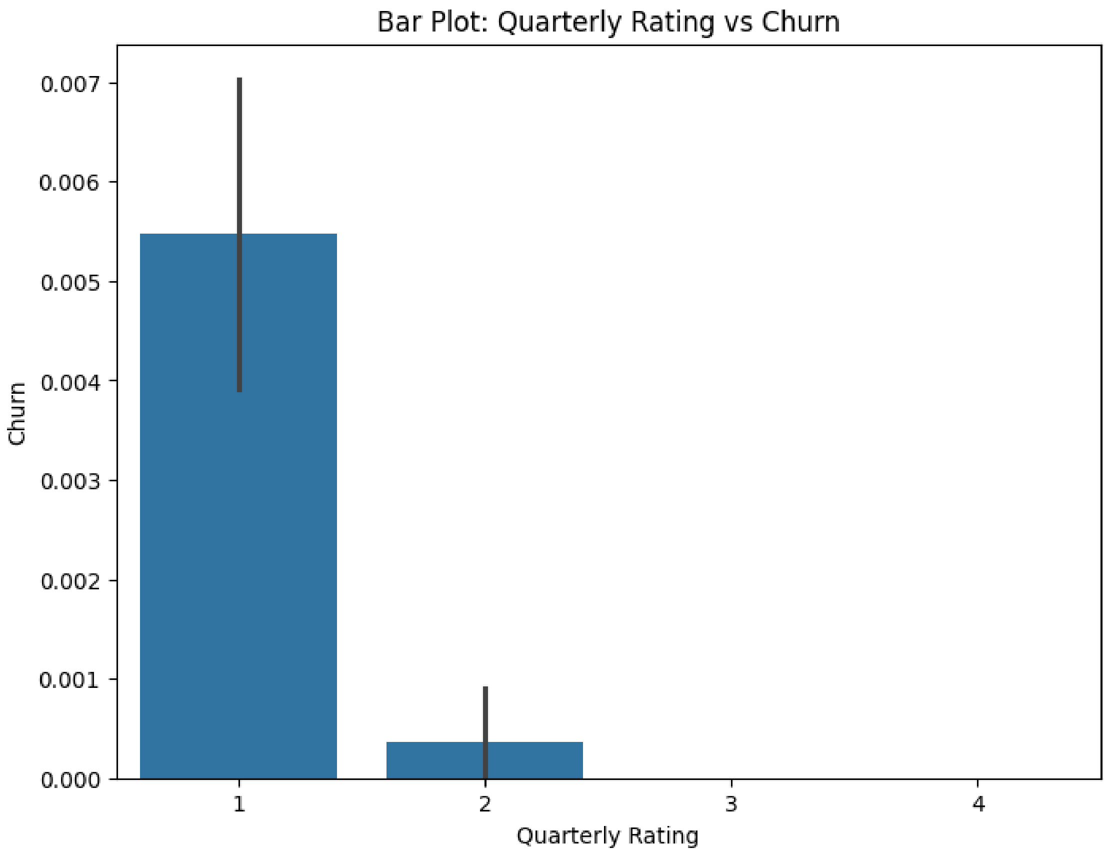

Bar Plot: Education_Level vs Churn
Churn
0.000
0.001
0.002
0.003
0.004
0.005
0
1
2
Education_Level

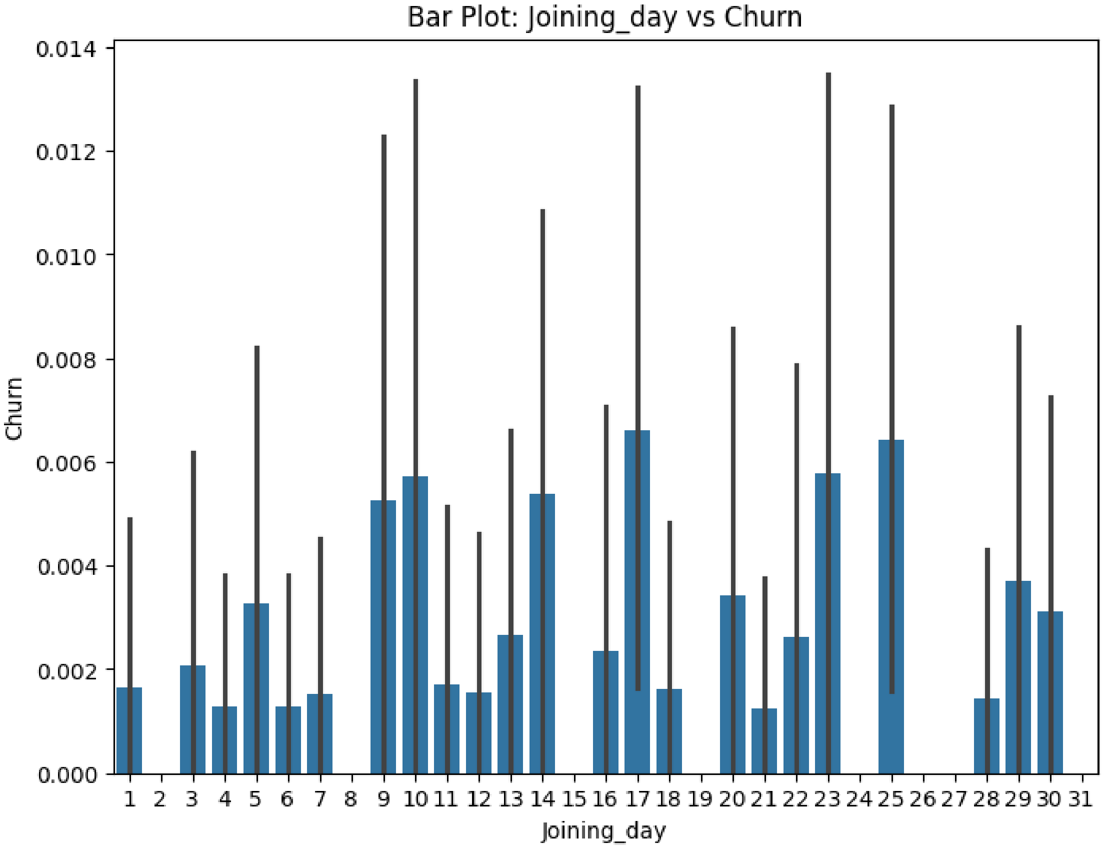

Bar Plot: Income vs Churn
Churn
1.0
0.8
0.6
0.4
0.2
0.0
Income

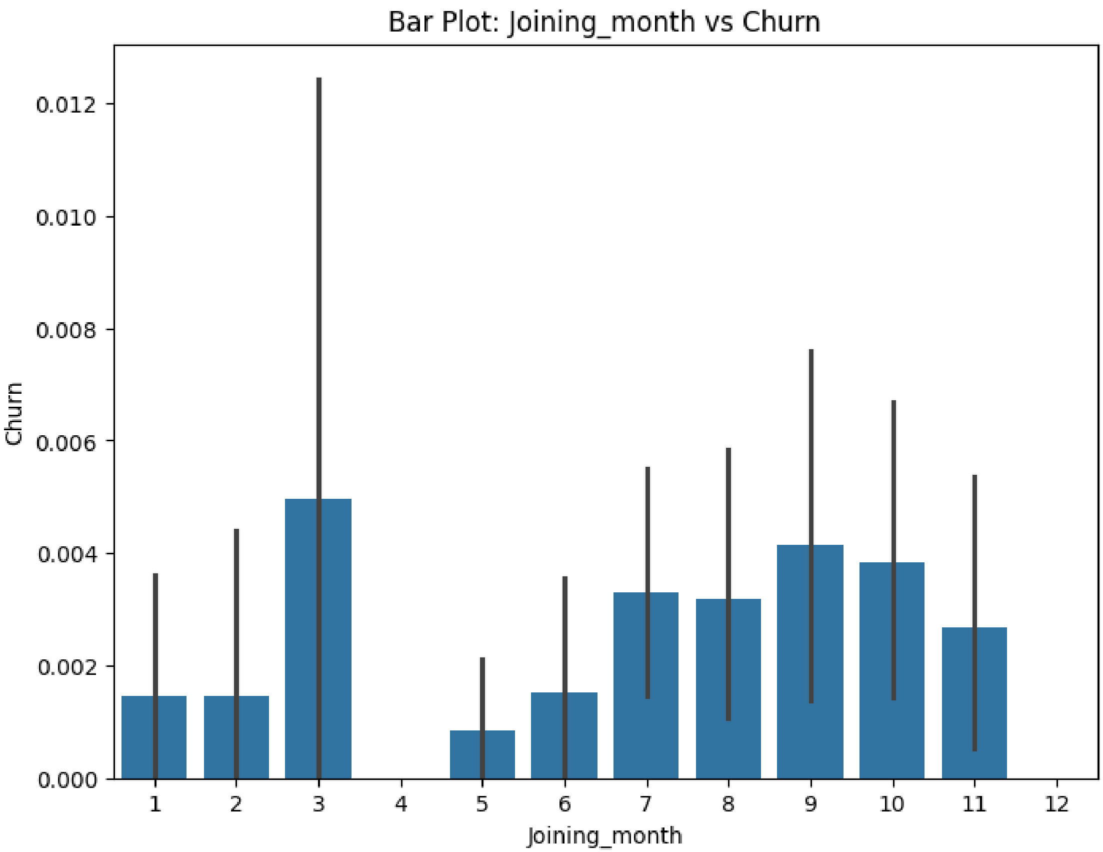

Bar Plot: Joining Designation vs Churn
Churn
0.000
0.001
0.002
0.003
0.004
0.005
Joining Designation
1
2
3
4
5

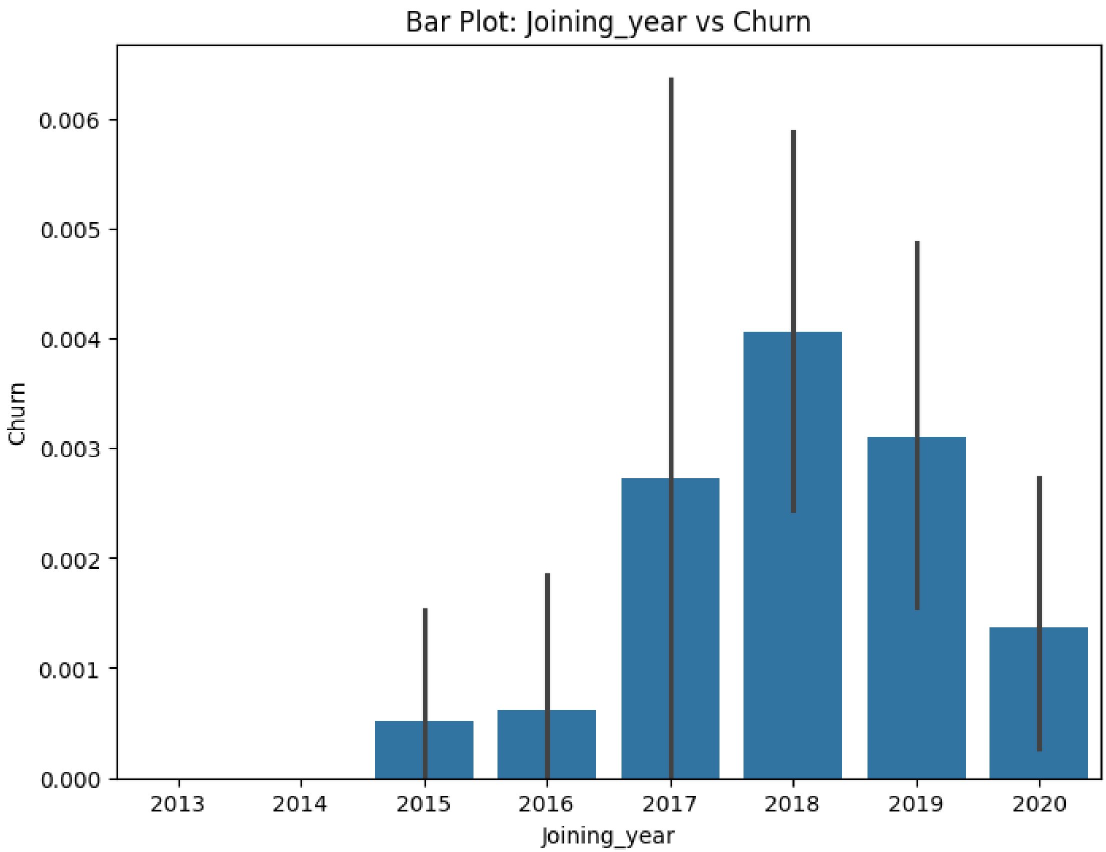

Bar Plot: Grade vs Churn
Churn
0.004
0.003
0.002
0.001
0.000
1
2
3
4
5
Grade

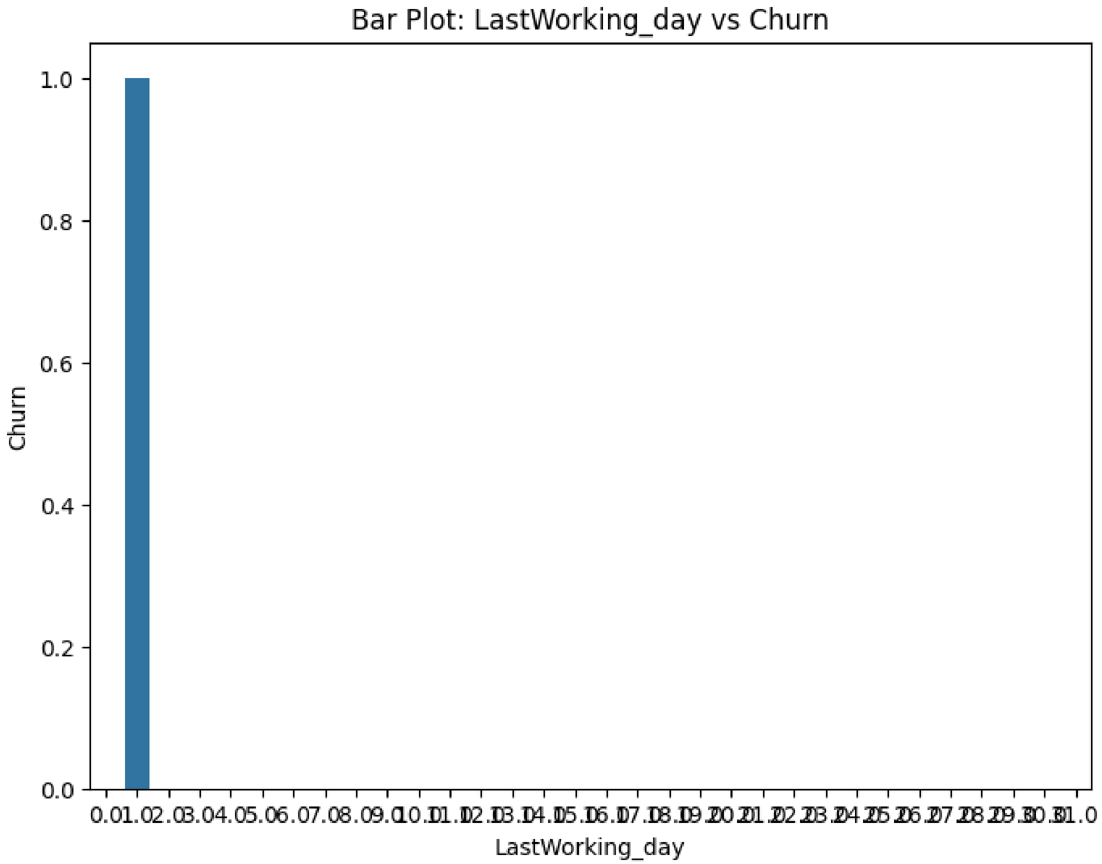

Bar Plot: Total Business Value vs Churn
Churn
1.0
0.8
0.6
0.4
0.2
0.0
Total Business Value

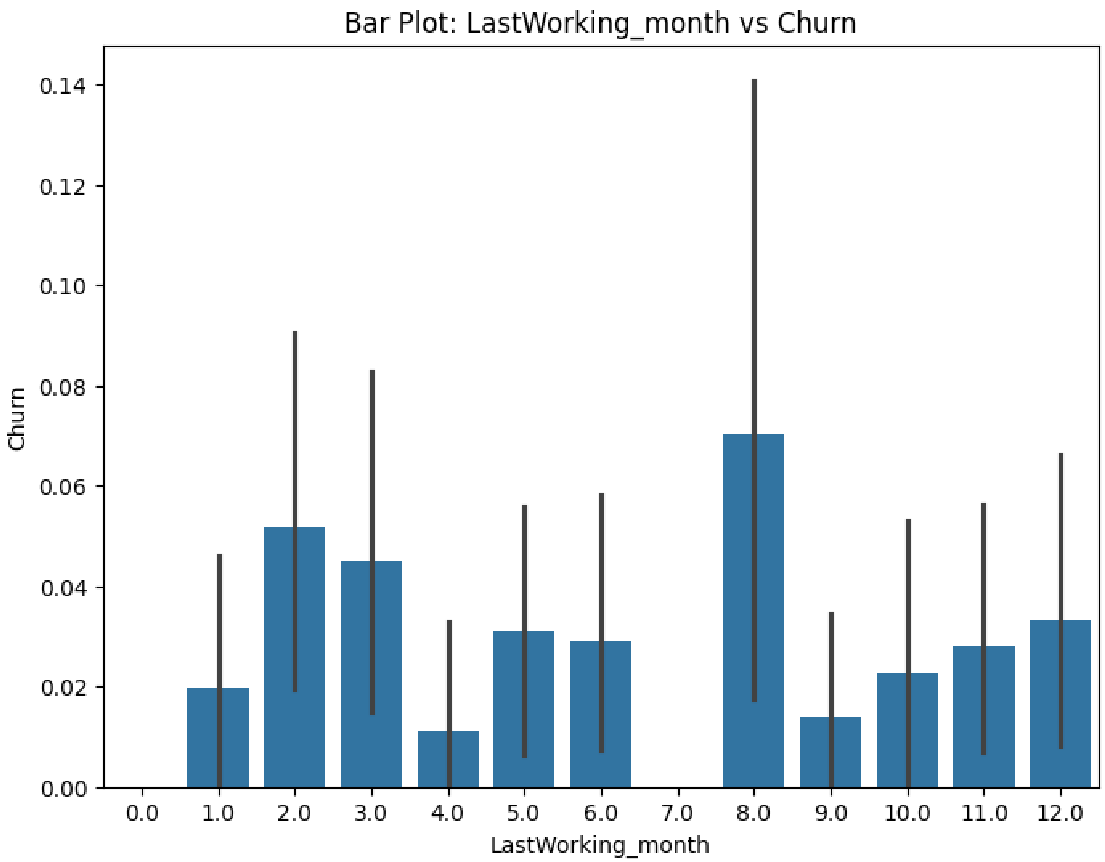

Bar Plot: Quarterly Rating vs Churn
Churn
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
1
2
3
4
Quarterly Rating

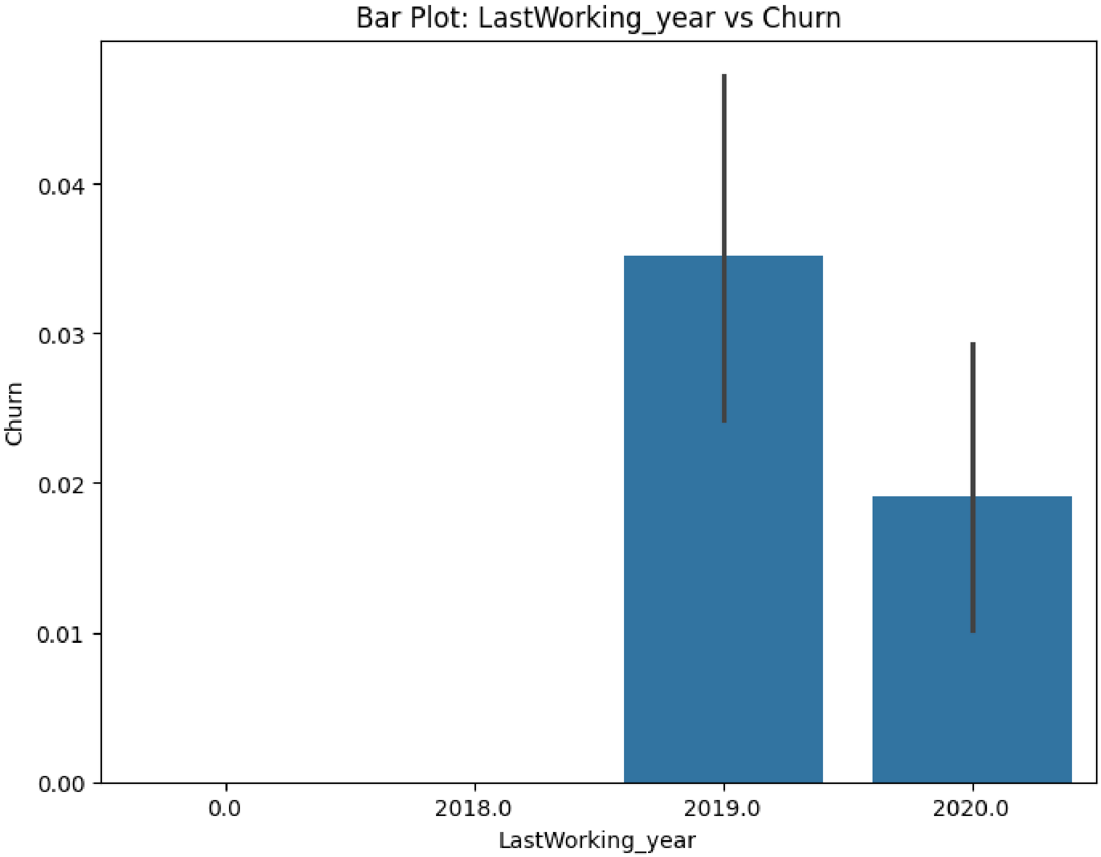

Bar Plot: Joining_day vs Churn
Churn
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
Joining_day
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

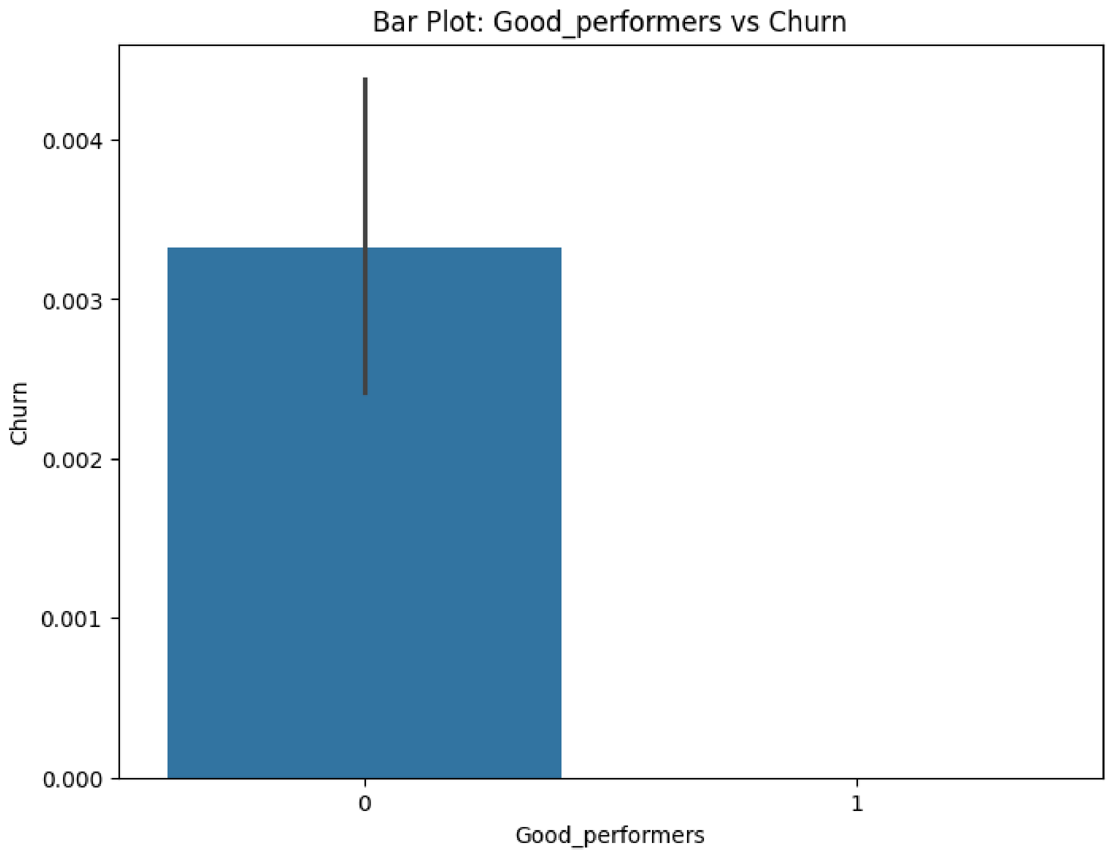

Bar Plot: Joining_month vs Churn
Churn
0.012
0.010
0.008
0.006
0.004
0.002
0.000
1
2
3
4
5
6
7
8
9
10
11
12
Joining_month

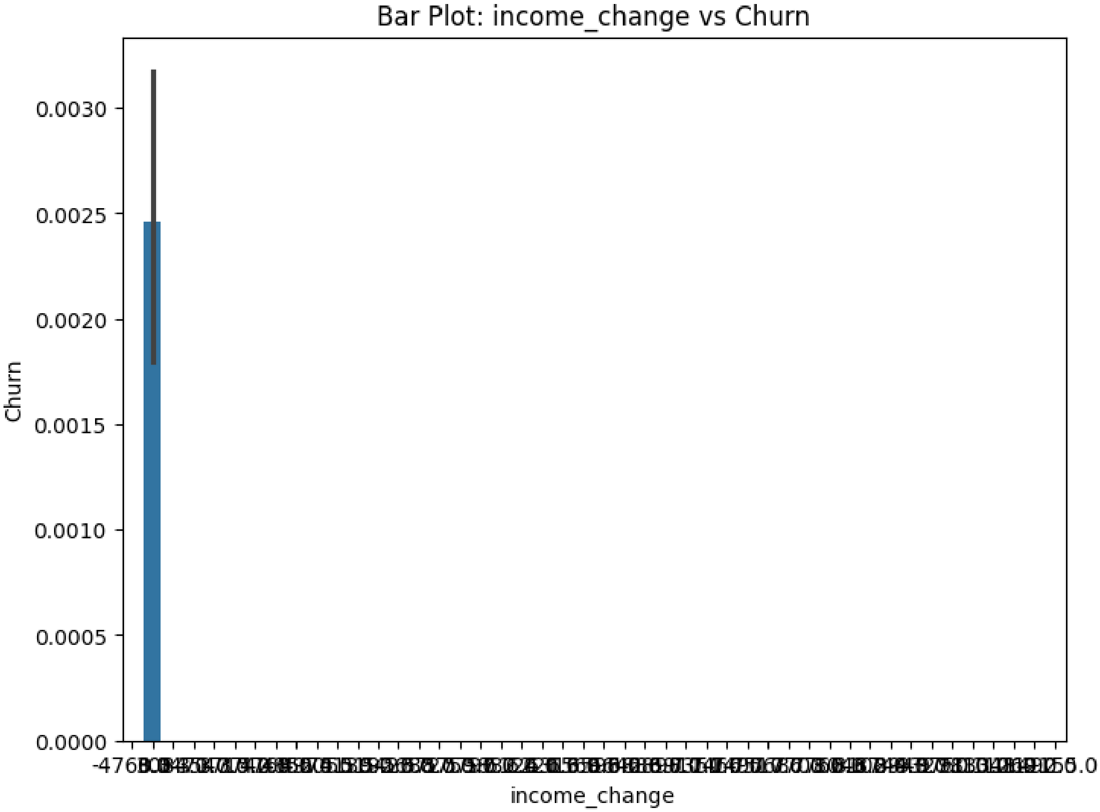

Bar Plot: Joining_year vs Churn
Churn
0.000
0.001
0.002
0.003
0.004
0.005
0.006
2013
2014
2015
2016
2017
2018
2019
2020
Joining_year

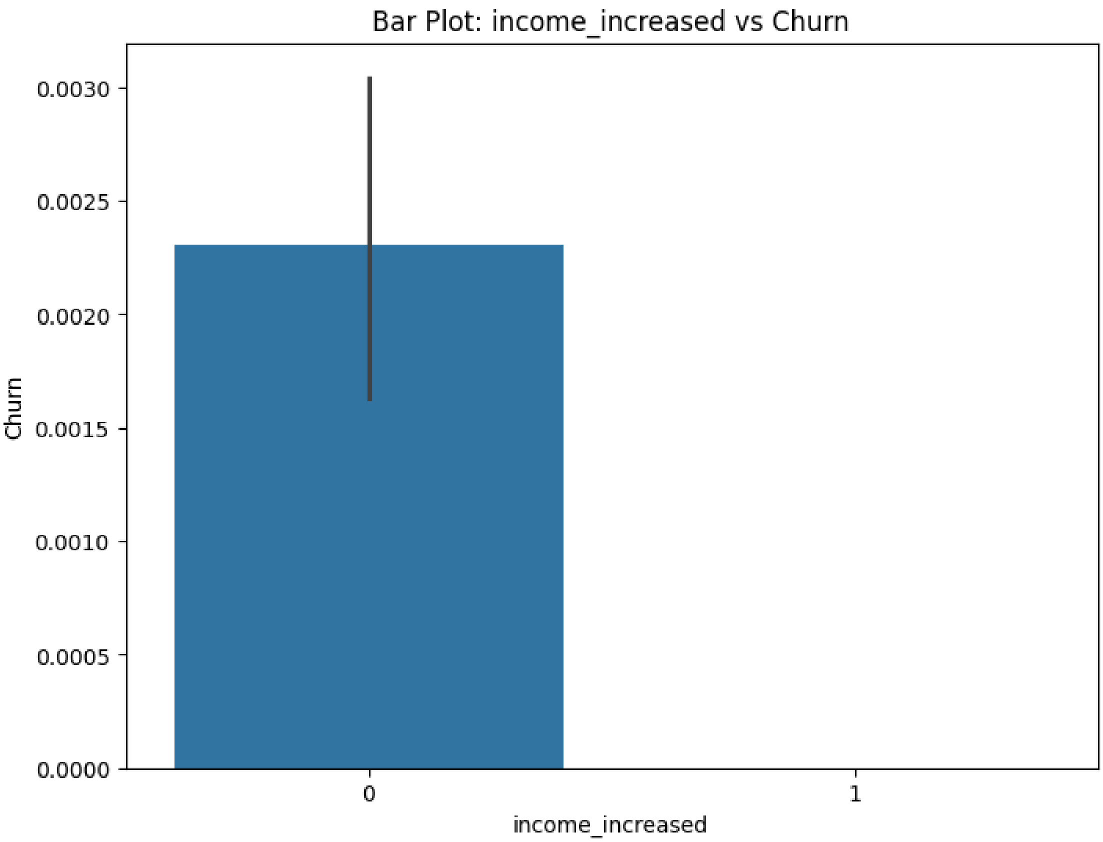

Bar Plot: LastWorking_day vs Churn
Churn
1.0
0.8
0.6
0.4
0.2
0.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
15.0
16.0
17.0
18.0
19.0
20.0
21.0
22.0
23.0
24.0
25.0
26.0
27.0
28.0
29.0
30.0
31.0
LastWorking_day

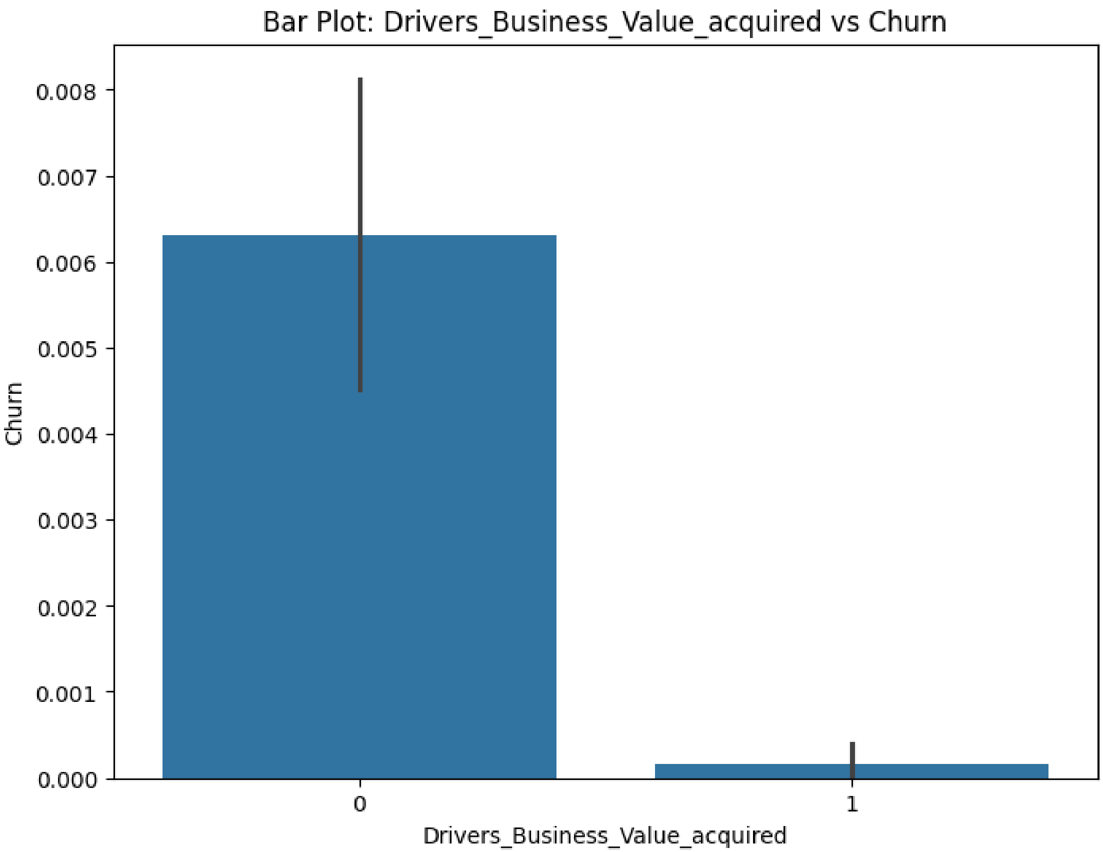

Bar Plot: LastWorking_month vs Churn
Churn
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
LastWorking_month
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0


Bar Plot: LastWorking_year vs Churn
Churn
0.00
0.01
0.02
0.03
0.04
0.0
2018.0
2019.0
2020.0
LastWorking_year


Bar Plot: Good_performers vs Churn
Churn
0.000
0.001
0.002
0.003
0.004
0
1
Good_performers


Bar Plot: income_change vs Churn
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
Churn
income_change


Bar Plot: income_increased vs Churn
Churn
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
1
income_increased



```
[ ]:
```

2 EDA Summary

2.0.1 * Ola Drivers who ride the vehicle are mostly Men whoes age lie between 25-30 on an average

2.0.2 * on an average a Ola driver makes an income of 50000

2.0.3 * The rating given to the drivers are maximum 1 and 2 and very few Drivers have 4 ratings

2.0.4 * This also states that the good performers are less

2.0.5 * Churn rate of the ola drivers is very less

2.0.6 * Income level that the driver makes through ola services has incresed

```
[ ]:
```

```
[ ]:
```

##Data Preprocessing

```
[ ]: df
```

```
[ ]:        Age  Gender  Education_Level  Income  Joining Designation  Grade  \
0      28.0     0.0                2   57387                    1      1
1      28.0     0.0                2   57387                    1      1
2      28.0     0.0                2   57387                    1      1
3      31.0     0.0                2   67016                    2      2
4      31.0     0.0                2   67016                    2      2
…     …     …              …     …                  …    …
19099  30.0     0.0                2   70254                    2      2
19100  30.0     0.0                2   70254                    2      2
19101  30.0     0.0                2   70254                    2      2
19102  30.0     0.0                2   70254                    2      2
19103  30.0     0.0                2   70254                    2      2

       Total Business Value  Quarterly Rating  Joining_day  Joining_month  \
0                   2381060                 2           24             12
1                   -665480                 2           24             12
2                         0                 2           24             12
3                         0                 1            6             11
4                         0                 1            6             11
…                     …               …         …           …
19099                740280                 3            8              6
19100                448370                 3            8              6
19101                     0                 2            8              6
19102                200420                 2            8              6
19103                411480                 2            8              6

       Joining_year  LastWorking_day  LastWorking_month  LastWorking_year  \
0              2018              0.0                0.0               0.0
1              2018              0.0                0.0               0.0
2              2018             11.0                3.0            2019.0
3              2020              0.0                0.0               0.0
4              2020              0.0                0.0               0.0
…             …              …                …               …
19099          2020              0.0                0.0               0.0
19100          2020              0.0                0.0               0.0
19101          2020              0.0                0.0               0.0
19102          2020              0.0                0.0               0.0
19103          2020              0.0                0.0               0.0

       Good_performers  Churn  income_change  income_increased  \
0                    0      0            NaN                 0
1                    0      0            0.0                 0
2                    0      0            0.0                 0
```

```
3                    0      0            NaN                 0
4                    0      0            0.0                 0
…                …    …            …               …
19099                1      0            0.0                 0
19100                1      0            0.0                 0
19101                0      0            0.0                 0
19102                0      0            0.0                 0
19103                0      0            0.0                 0

       Drivers_Business_Value_acquired
0                                    1
1                                    0
2                                    0
3                                    0
4                                    0
…                                 …
19099                                1
19100                                1
19101                                0
19102                                1
19103                                1

[19104 rows x 19 columns]
```

2.1 Data needs to be label encoded before applying machine learning models.

[ ]:

```
# Defining a Function to Convert Objects to Int
def object_to_int(dataframe_series):
    if dataframe_series.dtype=='object':
        dataframe_series = LabelEncoder().fit_transform(dataframe_series)
    return dataframe_series
```

[ ]:

```
df2 = df
df = df.apply(lambda x: object_to_int(x))
X = df.drop(columns = ['Churn'])
y = df['Churn'].values
df.head()
```

[ ]:

```
    Age  Gender  Education_Level  Income  Joining Designation  Grade  \
0  28.0     0.0                2   57387                    1      1
1  28.0     0.0                2   57387                    1      1
2  28.0     0.0                2   57387                    1      1
3  31.0     0.0                2   67016                    2      2
4  31.0     0.0                2   67016                    2      2

   Total Business Value  Quarterly Rating  Joining_day  Joining_month  \
0               2381060                 2           24             12
```

```
1               -665480                 2           24             12
2                     0                 2           24             12
3                     0                 1            6             11
4                     0                 1            6             11

   Joining_year  LastWorking_day  LastWorking_month  LastWorking_year  \
0          2018              0.0                0.0               0.0
1          2018              0.0                0.0               0.0
2          2018             11.0                3.0            2019.0
3          2020              0.0                0.0               0.0
4          2020              0.0                0.0               0.0

   Good_performers  Churn  income_change  income_increased  \
0                0      0            NaN                 0
1                0      0            0.0                 0
2                0      0            0.0                 0
3                0      0            NaN                 0
4                0      0            0.0                 0

   Drivers_Business_Value_acquired
0                                1
1                                0
2                                0
3                                0
4                                0
```

[ ]:

```
df.fillna(0, inplace=True)
```

[ ]:

```
df.head()
```

[ ]:

```
    Age  Gender  Education_Level  Income  Joining Designation  Grade  \
0  28.0     0.0                2   57387                    1      1
1  28.0     0.0                2   57387                    1      1
2  28.0     0.0                2   57387                    1      1
3  31.0     0.0                2   67016                    2      2
4  31.0     0.0                2   67016                    2      2

   Total Business Value  Quarterly Rating  Joining_day  Joining_month  \
0               2381060                 2           24             12
1               -665480                 2           24             12
2                     0                 2           24             12
3                     0                 1            6             11
4                     0                 1            6             11

   Joining_year  LastWorking_day  LastWorking_month  LastWorking_year  \
0          2018              0.0                0.0               0.0
1          2018              0.0                0.0               0.0
```

```
2          2018             11.0                 3.0            2019.0
3          2020              0.0                 0.0               0.0
4          2020              0.0                 0.0               0.0

   Good_performers  Churn  income_change  income_increased  \
0                0      0            0.0                 0
1                0      0            0.0                 0
2                0      0            0.0                 0
3                0      0            0.0                 0
4                0      0            0.0                 0

   Drivers_Business_Value_acquired
0                                1
1                                0
2                                0
3                                0
4                                0
```

[ ]:

3 Splitting the data into train and test sets

[ ]:

```python
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder

from sklearn import metrics
from sklearn.metrics import roc_curve
from sklearn.metrics import recall_score, confusion_matrix, precision_score

from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score, accuracy_score, classification_report
from xgboost import XGBClassifier
```

[ ]:

4 SCALING DATA

[ ]:
```python
from sklearn.preprocessing import StandardScaler

# Initialize the scaler
scaling = StandardScaler()

# Fit and transform the data
standardized_df = pd.DataFrame(scaling.fit_transform(df), columns=df.columns)
print(standardized_df)
```

```
            Age    Gender  Education_Level    Income  Joining Designation  \
0     -1.001679 -0.846793         1.222688 -0.267358            -0.825051
1     -1.001679 -0.846793         1.222688 -0.267358            -0.825051
2     -1.001679 -0.846793         1.222688 -0.267358            -0.825051
3     -0.543436 -0.846793         1.222688  0.044122             0.369747
4     -0.543436 -0.846793         1.222688  0.044122             0.369747
…             …         …                …         …                    …
19099 -0.696184 -0.846793         1.222688  0.148865             0.369747
19100 -0.696184 -0.846793         1.222688  0.148865             0.369747
19101 -0.696184 -0.846793         1.222688  0.148865             0.369747
19102 -0.696184 -0.846793         1.222688  0.148865             0.369747
19103 -0.696184 -0.846793         1.222688  0.148865             0.369747

          Grade  Total Business Value  Quarterly Rating  Joining_day  \
0     -1.220348              1.603674         -0.008812     0.861339
1     -1.220348             -1.096482         -0.008812     0.861339
2     -1.220348             -0.506666         -0.008812     0.861339
3     -0.246150             -0.506666         -0.999102    -1.091148
4     -0.246150             -0.506666         -0.999102    -1.091148
…             …                     …                 …            …
19099 -0.246150              0.149447          0.981477    -0.874205
19100 -0.246150             -0.109274          0.981477    -0.874205
19101 -0.246150             -0.506666         -0.008812    -0.874205
19102 -0.246150             -0.329033         -0.008812    -0.874205
19103 -0.246150             -0.141970         -0.008812    -0.874205

       Joining_month  Joining_year  LastWorking_day  LastWorking_month  \
0           1.591510      0.116445        -0.264712          -0.263929
1           1.591510      0.116445        -0.264712          -0.263929
2           1.591510      0.116445         1.777715           1.208073
3           1.267132      1.157666        -0.264712          -0.263929
4           1.267132      1.157666        -0.264712          -0.263929
…                  …             …                …                  …
19099      -0.354754      1.157666        -0.264712          -0.263929
19100      -0.354754      1.157666        -0.264712          -0.263929
19101      -0.354754      1.157666        -0.264712          -0.263929
19102      -0.354754      1.157666        -0.264712          -0.263929
```

```
19103      -0.354754      1.157666        -0.264712          -0.263929

       LastWorking_year  Good_performers     Churn  income_change  \
0             -0.303984        -0.666163 -0.048047      -0.044831
1             -0.303984        -0.666163 -0.048047      -0.044831
2              3.288788        -0.666163 -0.048047      -0.044831
3             -0.303984        -0.666163 -0.048047      -0.044831
4             -0.303984        -0.666163 -0.048047      -0.044831
…                   …              …        …             …
19099         -0.303984         1.501135 -0.048047      -0.044831
19100         -0.303984         1.501135 -0.048047      -0.044831
19101         -0.303984        -0.666163 -0.048047      -0.044831
19102         -0.303984        -0.666163 -0.048047      -0.044831
19103         -0.303984        -0.666163 -0.048047      -0.044831

       income_increased  Drivers_Business_Value_acquired
0             -0.048047                         0.730561
1             -0.048047                        -1.368812
2             -0.048047                        -1.368812
3             -0.048047                        -1.368812
4             -0.048047                        -1.368812
…                   …                               …
19099         -0.048047                         0.730561
19100         -0.048047                         0.730561
19101         -0.048047                        -1.368812
19102         -0.048047                         0.730561
19103         -0.048047                         0.730561

[19104 rows x 19 columns]
```

```
[ ]: standardized_df
```

```
[ ]:            Age    Gender  Education_Level    Income  Joining Designation  \
      0     -1.001679 -0.846793         1.222688 -0.267358            -0.825051
      1     -1.001679 -0.846793         1.222688 -0.267358            -0.825051
      2     -1.001679 -0.846793         1.222688 -0.267358            -0.825051
      3     -0.543436 -0.846793         1.222688  0.044122             0.369747
      4     -0.543436 -0.846793         1.222688  0.044122             0.369747
      …         …        …              …        …                  …
      19099 -0.696184 -0.846793         1.222688  0.148865             0.369747
      19100 -0.696184 -0.846793         1.222688  0.148865             0.369747
      19101 -0.696184 -0.846793         1.222688  0.148865             0.369747
      19102 -0.696184 -0.846793         1.222688  0.148865             0.369747
      19103 -0.696184 -0.846793         1.222688  0.148865             0.369747

               Grade  Total Business Value  Quarterly Rating  Joining_day  \
      0     -1.220348              1.603674         -0.008812     0.861339
```

```
1     -1.220348            -1.096482        -0.008812     0.861339
2     -1.220348            -0.506666        -0.008812     0.861339
3     -0.246150            -0.506666        -0.999102    -1.091148
4     -0.246150            -0.506666        -0.999102    -1.091148
…         …                  …               …          …
19099 -0.246150             0.149447         0.981477    -0.874205
19100 -0.246150            -0.109274         0.981477    -0.874205
19101 -0.246150            -0.506666        -0.008812    -0.874205
19102 -0.246150            -0.329033        -0.008812    -0.874205
19103 -0.246150            -0.141970        -0.008812    -0.874205

       Joining_month  Joining_year  LastWorking_day  LastWorking_month  \
0           1.591510      0.116445        -0.264712          -0.263929
1           1.591510      0.116445        -0.264712          -0.263929
2           1.591510      0.116445         1.777715           1.208073
3           1.267132      1.157666        -0.264712          -0.263929
4           1.267132      1.157666        -0.264712          -0.263929
…                …           …             …                  …
19099      -0.354754      1.157666        -0.264712          -0.263929
19100      -0.354754      1.157666        -0.264712          -0.263929
19101      -0.354754      1.157666        -0.264712          -0.263929
19102      -0.354754      1.157666        -0.264712          -0.263929
19103      -0.354754      1.157666        -0.264712          -0.263929

       LastWorking_year  Good_performers     Churn  income_change  \
0             -0.303984        -0.666163 -0.048047      -0.044831
1             -0.303984        -0.666163 -0.048047      -0.044831
2              3.288788        -0.666163 -0.048047      -0.044831
3             -0.303984        -0.666163 -0.048047      -0.044831
4             -0.303984        -0.666163 -0.048047      -0.044831
…                   …              …        …            …
19099         -0.303984         1.501135 -0.048047      -0.044831
19100         -0.303984         1.501135 -0.048047      -0.044831
19101         -0.303984        -0.666163 -0.048047      -0.044831
19102         -0.303984        -0.666163 -0.048047      -0.044831
19103         -0.303984        -0.666163 -0.048047      -0.044831

       income_increased  Drivers_Business_Value_acquired
0             -0.048047                         0.730561
1             -0.048047                        -1.368812
2             -0.048047                        -1.368812
3             -0.048047                        -1.368812
4             -0.048047                        -1.368812
…                   …                             …
19099         -0.048047                         0.730561
19100         -0.048047                         0.730561
19101         -0.048047                        -1.368812
```

```
19102         -0.048047                          0.730561
19103         -0.048047                          0.730561

[19104 rows x 19 columns]
```

```
[ ]: # Assuming 'Target' is your target variable
     # Standardize the data
     X = standardized_df.drop('Churn', axis=1)  # Features
     y = df['Churn']  # Target variable
```

```
[ ]: X_train,X_test,y_train,y_test=train_test_split(X, y,test_size=0.2)
```

```
[ ]: X_train.shape
```

```
[ ]: (15283, 18)
```

```
[ ]: X_test.shape
```

```
[ ]: (3821, 18)
```

```
[ ]: y_train.shape
```

```
[ ]: (15283,)
```

```
[ ]: y_test.shape
```

```
[ ]: (3821,)
```

```
[ ]:
```

#MODEL BUILDING

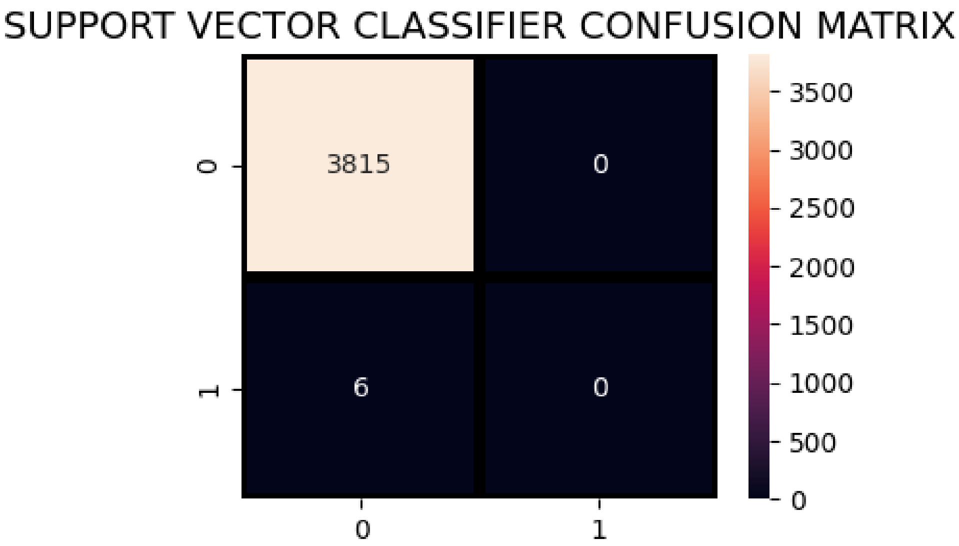
4.1 Multiple Machine Learning Model Evaluations and Testing

#SUPPORT VECTOR CLASSIFIER (SVC)

```
[ ]: svc_model = SVC( C=1.0, degree=3, gamma='scale', coef0=0.0, shrinking=True,␣
      ↪probability=False, tol=0.001, cache_size=200, class_weight=None,␣
      ↪verbose=False,
                   max_iter=-1, decision_function_shape='ovr', break_ties=False,␣
      ↪random_state=None)
     svc_model.fit(X_train,y_train)
     predict_y = svc_model.predict(X_test)
     accuracy_svc = svc_model.score(X_test,y_test)
     print("SVM accuracy is :",accuracy_svc)
     print('-'*60)
     train=svc_model.score(X_train,y_train)
     test=svc_model.score(X_test,y_test)
```

```python
print(f"Training score is : {train}\nTesting score is : {test}")
print('-'*60)
print(classification_report(y_test, predict_y))
print('-'*60)
plt.figure(figsize=(4,3))

sns.heatmap(confusion_matrix(y_test, predict_y),
                annot=True,fmt = "d",linecolor="k",linewidths=3)

plt.title("SUPPORT VECTOR CLASSIFIER CONFUSION MATRIX",fontsize=14)
plt.show()
```

```
SVM accuracy is : 0.9984297304370584
------------------------------------------------------------
Training score is : 0.9975135771772558
Testing score is : 0.9984297304370584
------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3815
           1       0.00      0.00      0.00         6

    accuracy                           1.00      3821
   macro avg       0.50      0.50      0.50      3821
weighted avg       1.00      1.00      1.00      3821

------------------------------------------------------------

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```



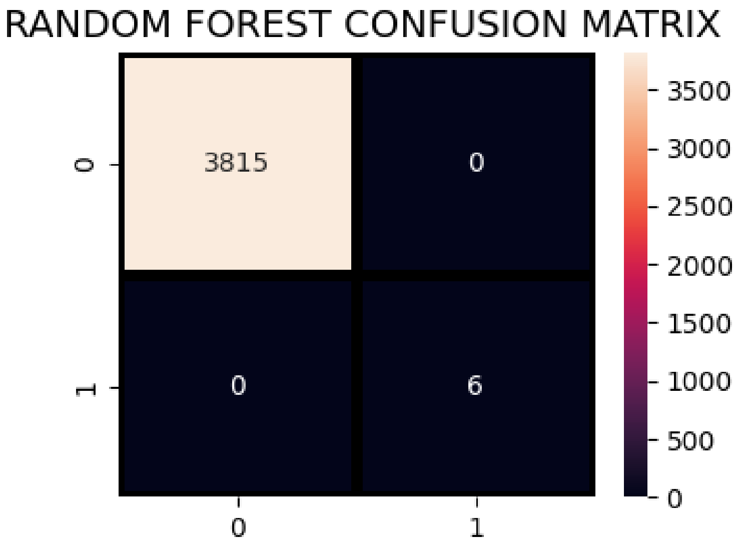
```
[ ]:
```

#RANDOM FOREST CLASSIFIER

```
[ ]: model_rf = RandomForestClassifier(n_estimators=1000 , oob_score = True, n_jobs␣
      ↪= -1,
                                       random_state =100, max_features = "sqrt",
                                       max_leaf_nodes = 35)
     model_rf.fit(X_train, y_train)

     # Make predictions
     prediction_test = model_rf.predict(X_test)
     print ("RandomForestClassifier  accuracy:",metrics.accuracy_score(y_test,␣
      ↪prediction_test))
     print('-'*60)

     train=model_rf.score(X_train,y_train)
     test=model_rf.score(X_test,y_test)
     print(f"Training score is : {train}\nTesting score is : {test}")
     print('-'*60)

     # 500, 50, 30

     print(classification_report(y_test, prediction_test))
     print('-'*60)

     plt.figure(figsize=(4,3))
```

```
sns.heatmap(confusion_matrix(y_test, prediction_test),annot=True,fmt =␣
 ↪"d",linecolor="k",linewidths=3)
plt.title(" RANDOM FOREST CONFUSION MATRIX",fontsize=14)
plt.show()
print('-'*60)
```

```
RandomForestClassifier  accuracy: 1.0
------------------------------------------------------------
Training score is : 1.0
Testing score is : 1.0
------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3815
           1       1.00      1.00      1.00         6

    accuracy                           1.00      3821
   macro avg       1.00      1.00      1.00      3821
weighted avg       1.00      1.00      1.00      3821

------------------------------------------------------------
```



```
------------------------------------------------------------
```

```
[ ]:
```

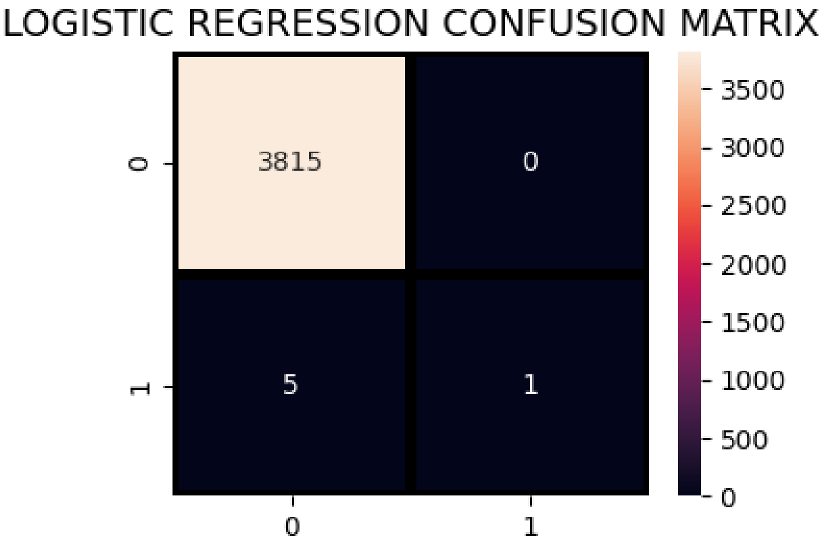
5 LOGISTIC REGRESSION

[ ]:

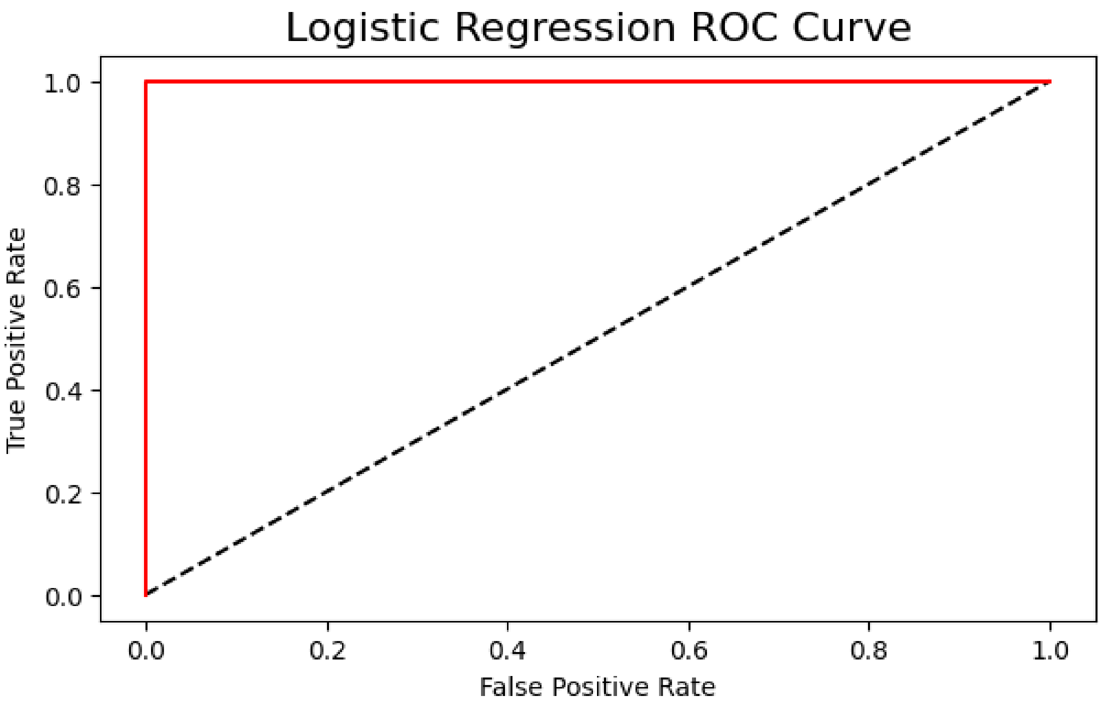
```
lr_model = LogisticRegression()
lr_model.fit(X_train,y_train)
accuracy_lr = lr_model.score(X_test,y_test)
print("Logistic Regression accuracy is :",accuracy_lr)
print('-'*60)

train=lr_model.score(X_train,y_train)
test=lr_model.score(X_test,y_test)
print(f"Training score is : {train}\nTesting score is : {test}")
print('-'*60)



lr_pred= lr_model.predict(X_test)
report = classification_report(y_test,lr_pred)
print(report)

print('-'*60)

plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, lr_pred),annot=True,fmt =␣
 ↪"d",linecolor="k",linewidths=3)
plt.title("LOGISTIC REGRESSION CONFUSION MATRIX",fontsize=14)
plt.show()

print('-'*60)


y_pred_prob = lr_model.predict_proba(X_test)[:,1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
plt.figure(figsize=(7, 4))

plt.plot([0, 1], [0, 1], 'k--' )
plt.plot(fpr, tpr, label='Logistic Regression',color = "r")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Logistic Regression ROC Curve',fontsize=16)
plt.show();
```

```
Logistic Regression accuracy is : 0.998691442030882
------------------------------------------------------------
Training score is : 0.9990185173068115
Testing score is : 0.998691442030882
------------------------------------------------------------
              precision    recall  f1-score   support
```

```
           0       1.00      1.00      1.00      3815
           1       1.00      0.17      0.29         6

    accuracy                           1.00      3821
   macro avg       1.00      0.58      0.64      3821
weighted avg       1.00      1.00      1.00      3821

------------------------------------------------------------
```



```
------------------------------------------------------------
```



[ ]:

6 CHECKING DATA IF IT IS BALANCED OR IMBALANCED

[ ]:

```
class_counts = df['Churn'].value_counts()
print(class_counts)
```

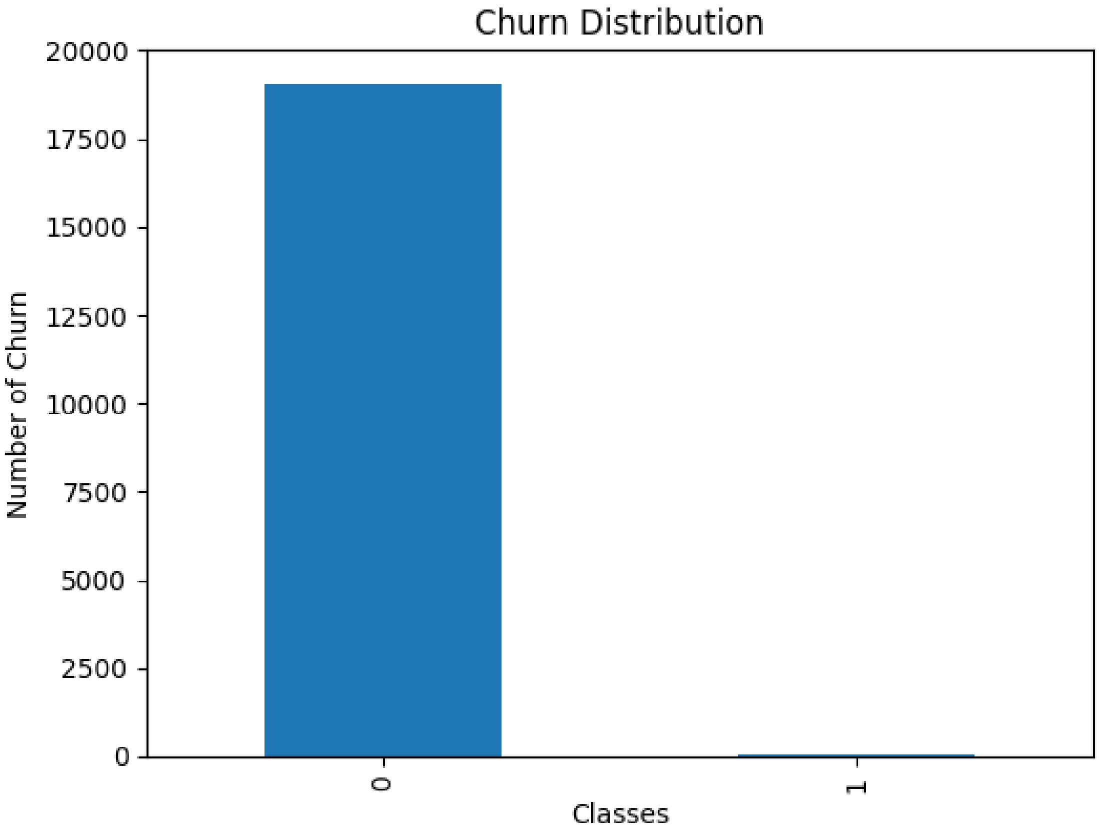
```
Churn
0    19060
1       44
Name: count, dtype: int64
```

[ ]:

```
# it is imbalanced data set throught the analysis we found out
```

[ ]:

```
import matplotlib.pyplot as plt

# Plotting class distribution
class_counts.plot(kind='bar')
plt.title('Churn Distribution')
plt.xlabel('Classes')
plt.ylabel('Number of Churn')
plt.show()
```



```
[ ]: # So applying the snote technique on imbalanced dataset
```

```
[ ]: #LOGISTIC REGRESSION USING ---- SMOTE TECHNIQUE
```

```
[ ]:
```

#Oversample Training Data (SMOTE-NC) SMOTE is an oversampling method that balances imbalanced datasets by sampling (with replacement) minority class. SMOTE-NC stands for Synthetic Minority Over-sampling TEchnique for data with Numerical-Categorical features. Note that only training data is oversampled. The testing data is untouched.

```
[ ]: from imblearn.combine import SMOTEENN
```

```
[ ]: sm = SMOTEENN()
     X_resampled, y_resampled = sm.fit_resample(X,y)
```

```
[ ]: X_resampled
```

```
[ ]:            Age    Gender  Education_Level    Income  Joining Designation  \
     0   -1.001679 -0.846793         1.222688 -0.267358            -0.825051
```

```
1     -1.001679 -0.846793         1.222688 -0.267358            -0.825051
2     -1.001679 -0.846793         1.222688 -0.267358            -0.825051
3     -0.543436 -0.846793         1.222688  0.044122             0.369747
4     -0.543436 -0.846793         1.222688  0.044122             0.369747
…         …       …              …       …                  …
37976 -1.001679  1.180926         1.222688 -0.809245             0.309372
37977 -0.496617 -0.846793         1.222688 -0.441621            -0.273887
37978 -1.100190  1.180926         1.222688 -1.263787            -0.825051
37979  0.862901 -0.846793         1.024456 -0.277839             0.369747
37980 -0.418408  1.058269        -1.276855 -0.742824             0.297473

          Grade  Total Business Value  Quarterly Rating  Joining_day  \
0     -1.220348              1.603674         -0.008812     0.861339
1     -1.220348             -1.096482         -0.008812     0.861339
2     -1.220348             -0.506666         -0.008812     0.861339
3     -0.246150             -0.506666         -0.999102    -1.091148
4     -0.246150             -0.506666         -0.999102    -1.091148
…         …                   …               …          …
37976 -0.246150             -0.506666         -0.999102    -0.944308
37977 -0.770948             -0.506666         -0.999102    -1.316486
37978 -1.220348             -0.506666         -0.999102    -0.339970
37979 -0.246150             -0.506666         -0.999102     1.340117
37980 -0.246150             -0.506666         -0.999102     0.154530

       Joining_month  Joining_year  LastWorking_day  LastWorking_month  \
0           1.591510      0.116445        -0.264712          -0.263929
1           1.591510      0.116445        -0.264712          -0.263929
2           1.591510      0.116445         1.777715           1.208073
3           1.267132      1.157666        -0.264712          -0.263929
4           1.267132      1.157666        -0.264712          -0.263929
…              …           …              …                …
37976       0.343174      1.078745        -0.079036           5.059030
37977      -0.754446      0.396897        -0.079036           0.981727
37978       0.060513      0.228364        -0.079036           0.822887
37979       1.061329      0.199022        -0.079036           3.817064
37980       0.598756      0.574072        -0.079036           5.594399

       LastWorking_year  Good_performers  income_change  income_increased  \
0             -0.303984        -0.666163      -0.044831         -0.048047
1             -0.303984        -0.666163      -0.044831         -0.048047
2              3.288788        -0.666163      -0.044831         -0.048047
3             -0.303984        -0.666163      -0.044831         -0.048047
4             -0.303984        -0.666163      -0.044831         -0.048047
…                 …              …            …               …
37976          3.290477        -0.666163      -0.044831         -0.048047
37977          3.288788        -0.666163      -0.044831         -0.048047
37978          3.288788        -0.666163      -0.044831         -0.048047
```

```
37979          3.288788        -0.666163      -0.044831          -0.048047
37980          3.288788        -0.666163      -0.044831          -0.048047

       Drivers_Business_Value_acquired
0                             0.730561
1                            -1.368812
2                            -1.368812
3                            -1.368812
4                            -1.368812
…                                    …
37976                        -1.368812
37977                        -1.368812
37978                        -1.368812
37979                        -1.368812
37980                        -1.368812

[37981 rows x 18 columns]
```

```
[ ]: y_resampled
```

```
[ ]: 0        0
     1        0
     2        0
     3        0
     4        0
             ..
     37976    1
     37977    1
     37978    1
     37979    1
     37980    1
     Name: Churn, Length: 37981, dtype: int64
```

```
[ ]: xr_train,xr_test,yr_train,yr_test=train_test_split(X_resampled,␣
      ↪y_resampled,test_size=0.2)
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

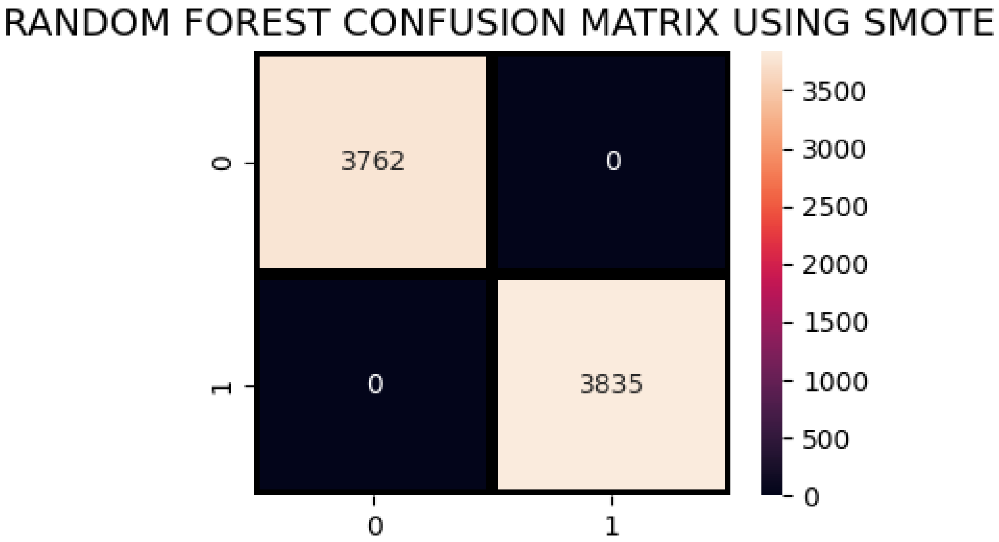
```
[ ]:
```

7 Random Forest Using SMOTE Technique

```
[ ]: model_RandomForestClassifier_smote=RandomForestClassifier(n_estimators=1000 ,␣
      ↪oob_score = True, n_jobs = -1,
                                         random_state =65, max_features = "sqrt",
                                         max_leaf_nodes = 35)
     model_RandomForestClassifier_smote.fit(xr_train, yr_train)

     # Make predictions
     prediction_test = model_RandomForestClassifier_smote.predict(xr_test)
     print ("RandomForestClassifierusingSMOTE  accuracy:",metrics.
      ↪accuracy_score(yr_test, prediction_test))
     print('-'*60)

     # 500, 50, 30

     train=model_RandomForestClassifier_smote.score(xr_train,yr_train)
     test=model_RandomForestClassifier_smote.score(xr_test,yr_test)
     print(f"Training score is : {train}\nTesting score is : {test}")
     print('-'*60)


     print(classification_report(yr_test, prediction_test))
     print('-'*60)

     plt.figure(figsize=(4,3))
     sns.heatmap(confusion_matrix(yr_test, prediction_test),annot=True,fmt =␣
      ↪"d",linecolor="k",linewidths=3)
     plt.title(" RANDOM FOREST CONFUSION MATRIX USING SMOTE",fontsize=14)
     plt.show()
     print('-'*60)


     y_rfpred_prob = model_rf.predict_proba(xr_test)[:,1]
     fpr_rf, tpr_rf, thresholds = roc_curve(yr_test, y_rfpred_prob)
     plt.figure(figsize=(8, 5))
     plt.plot([0, 1], [0, 1], 'k--' )
     plt.plot(fpr_rf, tpr_rf, label='Random Forest',color = "r")
     plt.xlabel('False Positive Rate')
     plt.ylabel('True Positive Rate')
     plt.title('Random Forest ROC Curve',fontsize=16)
     plt.show();
```

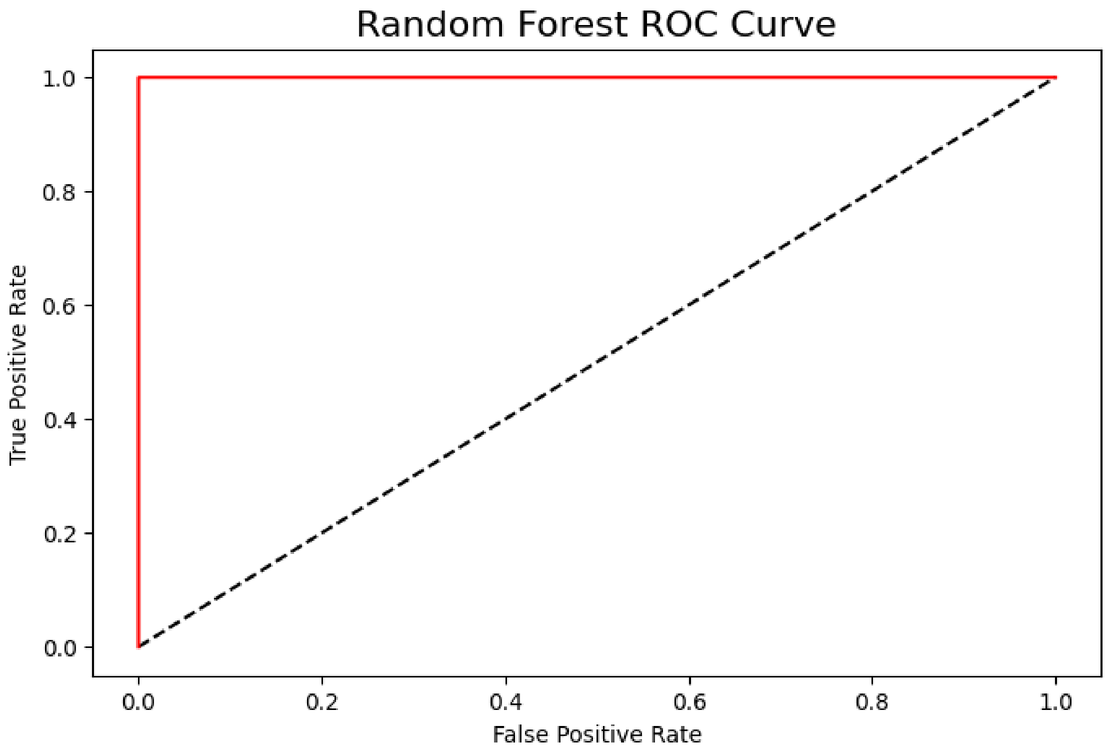
```
RandomForestClassifierusingSMOTE  accuracy: 1.0
------------------------------------------------------------
Training score is : 1.0
Testing score is : 1.0
```

```
------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3762
           1       1.00      1.00      1.00      3835

    accuracy                           1.00      7597
   macro avg       1.00      1.00      1.00      7597
weighted avg       1.00      1.00      1.00      7597

------------------------------------------------------------
```



RANDOM FOREST CONFUSION MATRIX USING SMOTE

```
------------------------------------------------------------
```



[ ]:

```
```

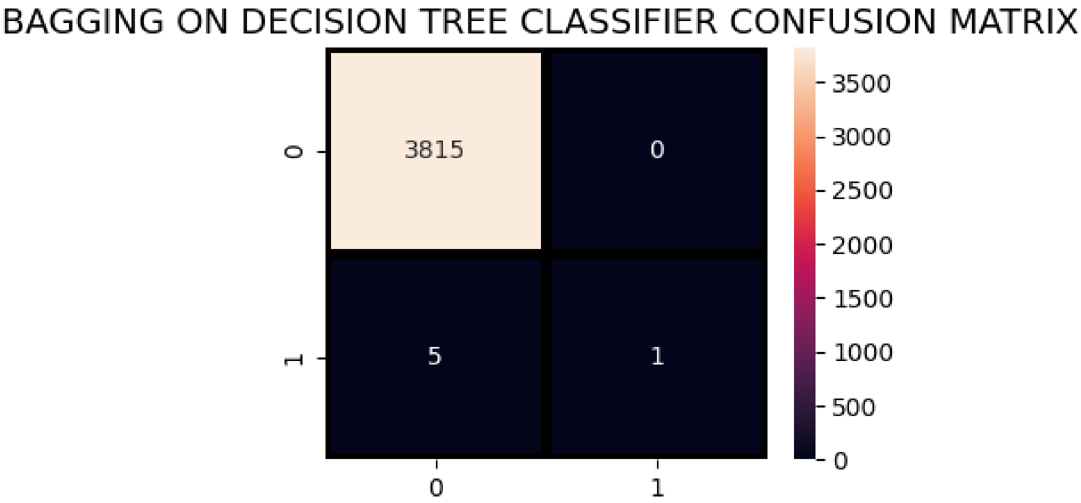
8 USING BAGGING ON DECISION TREE

[ ]:

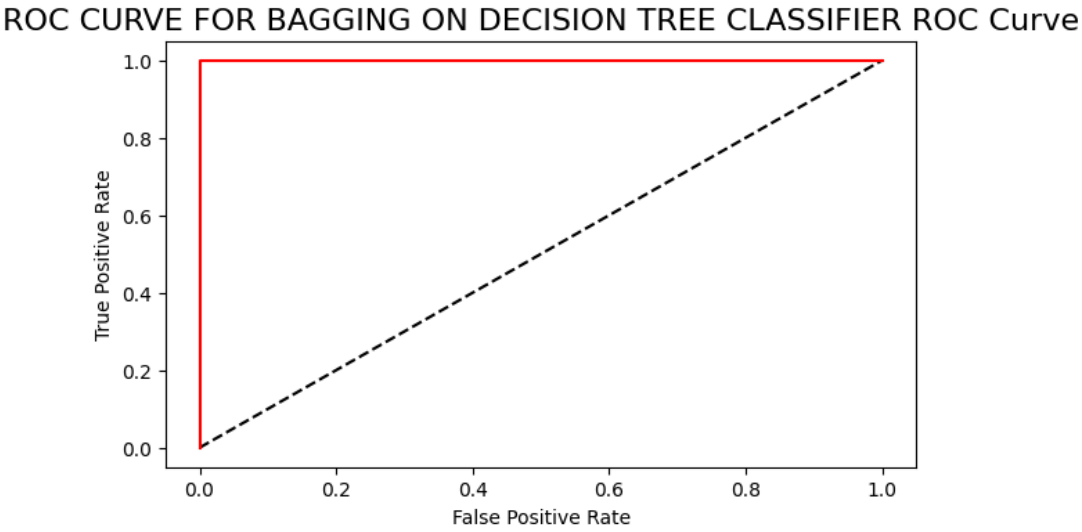
```
from sklearn.ensemble import BaggingClassifier
bg= BaggingClassifier(
                       max_samples=110, n_estimators=80,
                       max_features=15, n_jobs=-1,
                       random_state=0)



bg=BaggingClassifier(LogisticRegression())
bg.fit(X_train,y_train)
accuracy_Bagging = bg.score(X_test,y_test)
print("BAGGING ON DECISION TREE :",accuracy_Bagging)
print('-'*60)




train=bg.score(X_train,y_train)
```

```
test=bg.score(X_test,y_test)
print(f"Training score is : {train}\nTesting score is : {test}")
print('-'*60)


Bagging_pred=bg.predict(X_test)
report=classification_report(y_test,Bagging_pred)
print(report)


print('-'*60)
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, Bagging_pred),annot=True,fmt = "d",linecolor="k",linewidths=3)
plt.title("BAGGING ON DECISION TREE CLASSIFIER CONFUSION MATRIX ",fontsize=14)
plt.show()

print('-'*60)


y_pred_prob = bg.predict_proba(X_test)[:,1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
plt.figure(figsize=(7, 4))

plt.plot([0, 1], [0, 1], 'k--' )
plt.plot(fpr, tpr, label='BAGGING ON DECISION TREE CLASSIFIER',color = "r")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title(' ROC CURVE FOR BAGGING ON DECISION TREE CLASSIFIER ROC Curve ',fontsize=16)
plt.show();
```

```
BAGGING ON DECISION TREE : 0.998691442030882
------------------------------------------------------------
Training score is : 0.9986259242295361
Testing score is : 0.998691442030882
------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3815
           1       1.00      0.17      0.29         6

    accuracy                           1.00      3821
   macro avg       1.00      0.58      0.64      3821
weighted avg       1.00      1.00      1.00      3821

------------------------------------------------------------
```



```
------------------------------------------------------------
```



[ ]:

```

```

9 ADABOOST CLASSIFIER

[ ]:

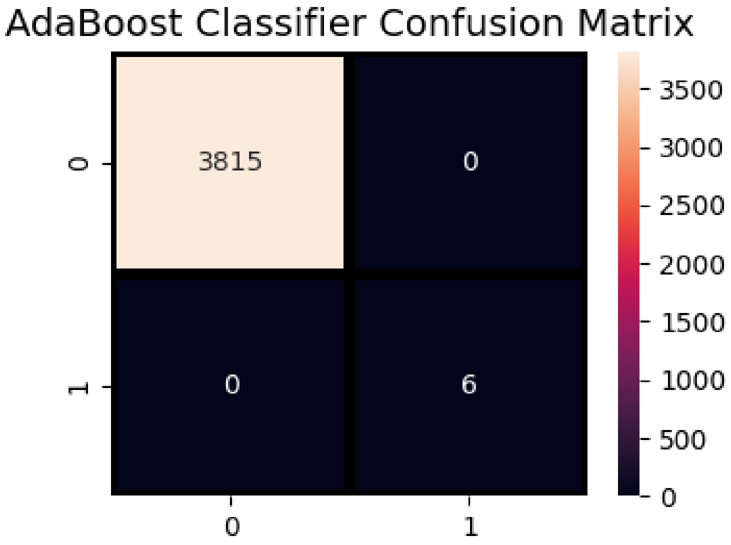
```
a_model = AdaBoostClassifier()
a_model.fit(X_train,y_train)
a_preds = a_model.predict(X_test)
print("AdaBoost Classifier accuracy")
```

```python
print(metrics.accuracy_score(y_test, a_preds))
print('-'*60)



train=a_model.score(X_train,y_train)
test=a_model.score(X_test,y_test)
print(f"Training score is : {train}\nTesting score is : {test}")
print('-'*60)


print(classification_report(y_test, a_preds))

plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, a_preds),
                annot=True,fmt = "d",linecolor="k",linewidths=3)
print('-'*60)
plt.title("AdaBoost Classifier Confusion Matrix",fontsize=14)
plt.show()
print('-'*60)

y_pred_prob = a_model.predict_proba(X_test)[:,1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
plt.figure(figsize=(7, 4))

plt.plot([0, 1], [0, 1], 'k--' )
plt.plot(fpr, tpr, label='ADABOOST CLASSIFIER',color = "r")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC CURVE FOR ADABOOST',fontsize=16)
plt.show();
```

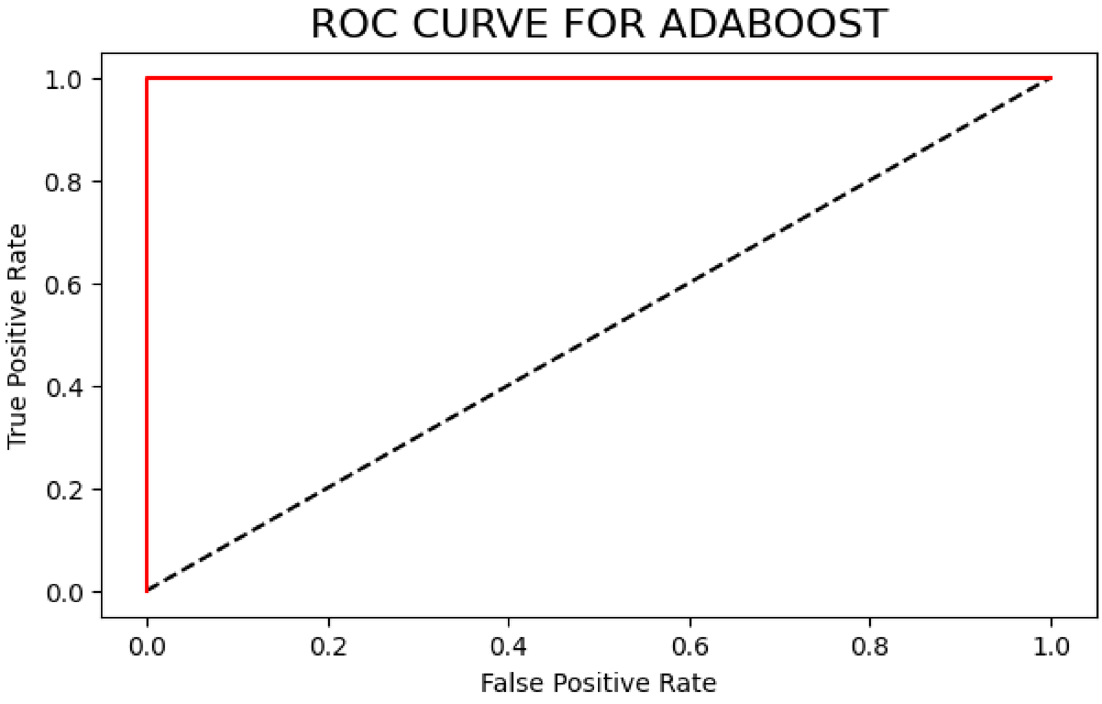
```
/usr/local/lib/python3.10/dist-
packages/sklearn/ensemble/_weight_boosting.py:527: FutureWarning: The SAMME.R
algorithm (the default) is deprecated and will be removed in 1.6. Use the SAMME
algorithm to circumvent this warning.
  warnings.warn(
```

```
AdaBoost Classifier accuracy
1.0
------------------------------------------------------------
Training score is : 1.0
Testing score is : 1.0
------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3815
           1       1.00      1.00      1.00         6
```

```
    accuracy                           1.00      3821
   macro avg       1.00      1.00      1.00      3821
weighted avg       1.00      1.00      1.00      3821

------------------------------------------------------------
```



```
------------------------------------------------------------
```



[ ]:

```

```

[ ]:

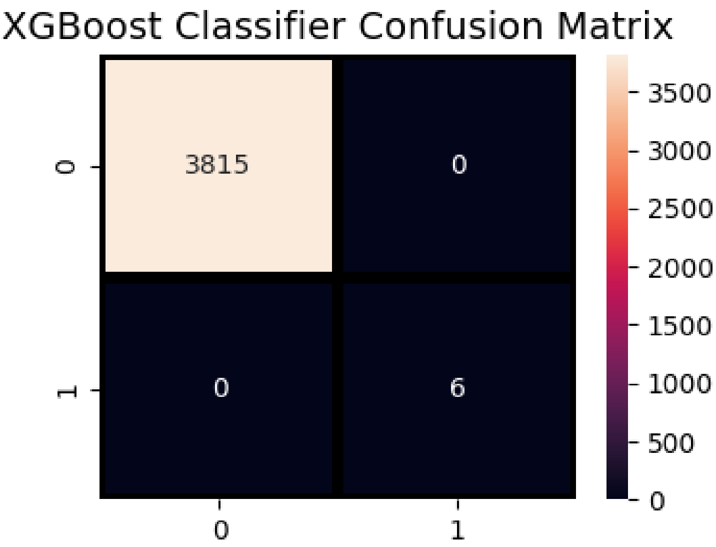
```

```

10 XGboost

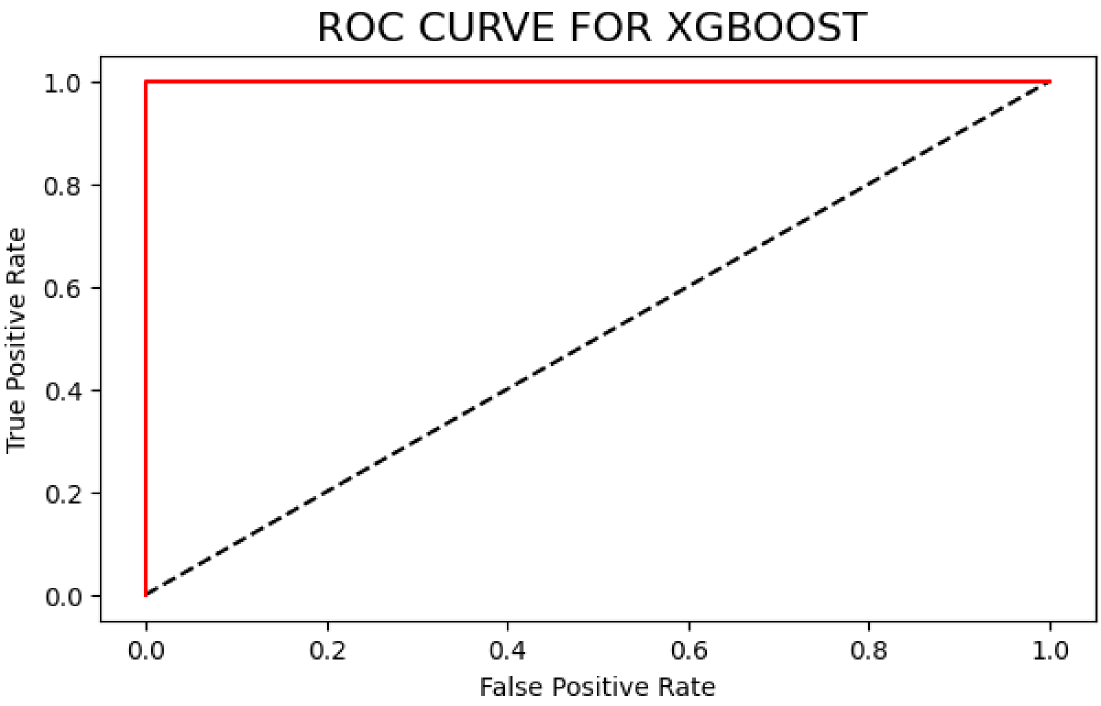
[ ]:

```
from xgboost import XGBClassifier

xgb_clf = XGBClassifier(learning_rate=0.01, random_state=0,
                        n_jobs=-1)
xgb_clf.fit(X_train, y_train);




xgb_preds = a_model.predict(X_test)
print("XGBoost Classifier accuracy")
print(metrics.accuracy_score(y_test, xgb_preds))

print('-'*60)



train=xgb_clf.score(X_train,y_train)
test=xgb_clf.score(X_test,y_test)
print(f"Training score is : {train}\nTesting score is : {test}")
print('-'*60)

print(classification_report(y_test, xgb_preds))

plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, xgb_preds),
                annot=True,fmt = "d",linecolor="k",linewidths=3)
print('-'*60)
plt.title("XGBoost Classifier Confusion Matrix",fontsize=14)
plt.show()
print('-'*60)

y_pred_prob = a_model.predict_proba(X_test)[:,1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
plt.figure(figsize=(7, 4))

plt.plot([0, 1], [0, 1], 'k--' )
plt.plot(fpr, tpr, label='XGBOOST CLASSIFIER ',color = "r")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC CURVE FOR XGBOOST ',fontsize=16)
plt.show();
```

```
XGBoost Classifier accuracy
1.0
------------------------------------------------------------
Training score is : 1.0
Testing score is : 1.0
------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3815
           1       1.00      1.00      1.00         6

    accuracy                           1.00      3821
   macro avg       1.00      1.00      1.00      3821
weighted avg       1.00      1.00      1.00      3821

------------------------------------------------------------
```



```
------------------------------------------------------------
```



[ ]:

[ ]:

11 ALL THE ALOGITHMS BAGGING , BOOSTING AND RANDOMFOREST ALL ARE WORKING GOOD ON THE DATA AND PREDITION IS ACCURATE BECAUSE ALL THE ALGORITHMS ACCURACY IS 100% SO THE DATA SET IS PREDICTING GOOD ON THE ML ALGORITHMS

[ ]:

[ ]:

[ ]:

[ ]: