capstone-project-1

February 21, 2024

[99]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt



df1 = pd.read_csv('/content/Credit_card[1].csv')
df1
```

[99]:
```
       Ind_ID GENDER Car_Owner Propert_Owner  CHILDREN  Annual_income  \
0     5008827      M         Y             Y         0       180000.0
1     5009744      F         Y             N         0       315000.0
2     5009746      F         Y             N         0       315000.0
3     5009749      F         Y             N         0            NaN
4     5009752      F         Y             N         0       315000.0
…         …    …         …             …        …              …
1543  5028645      F         N             Y         0            NaN
1544  5023655      F         N             N         0       225000.0
1545  5115992      M         Y             Y         2       180000.0
1546  5118219      M         Y             N         0       270000.0
1547  5053790      F         Y             Y         0       225000.0

               Type_Income                      EDUCATION  \
0                Pensioner               Higher education
1     Commercial associate               Higher education
2     Commercial associate               Higher education
3     Commercial associate               Higher education
4     Commercial associate               Higher education
…                      …                             …
1543  Commercial associate               Higher education
1544  Commercial associate              Incomplete higher
1545               Working               Higher education
1546               Working  Secondary / secondary special
1547               Working               Higher education

            Marital_status       Housing_type  Birthday_count  Employed_days  \
0                  Married  House / apartment        -18772.0         365243
```

```
1                  Married  House / apartment       -13557.0           -586
2                  Married  House / apartment            NaN           -586
3                  Married  House / apartment       -13557.0           -586
4                  Married  House / apartment       -13557.0           -586
…                        …                  …              …             …
1543               Married  House / apartment       -11957.0          -2182
1544  Single / not married  House / apartment       -10229.0          -1209
1545               Married  House / apartment       -13174.0          -2477
1546        Civil marriage  House / apartment       -15292.0           -645
1547               Married  House / apartment       -16601.0          -2859

      Mobile_phone  Work_Phone  Phone  EMAIL_ID Type_Occupation  \
0                1           0      0         0             NaN
1                1           1      1         0             NaN
2                1           1      1         0             NaN
3                1           1      1         0             NaN
4                1           1      1         0             NaN
…              …          …    …       …             …
1543             1           0      0         0        Managers
1544             1           0      0         0     Accountants
1545             1           0      0         0        Managers
1546             1           1      1         0         Drivers
1547             1           0      0         0             NaN

      Family_Members
0                  2
1                  2
2                  2
3                  2
4                  2
…                …
1543               2
1544               1
1545               4
1546               2
1547               2

[1548 rows x 18 columns]
```

[100]:
```
df2 = pd.read_csv('/content/Credit_card_label[1].csv')
df2
```

[100]:
```
       Ind_ID  label
0     5008827      1
1     5009744      1
2     5009746      1
3     5009749      1
```

```
4     5009752      1
…        …     …
1543  5028645      0
1544  5023655      0
1545  5115992      0
1546  5118219      0
1547  5053790      0

[1548 rows x 2 columns]
```

[101]:
```
merge_df = pd.merge(df1,df2, on ='Ind_ID') # merging the above two data with ␣
↪'Ind_ID' column as it is the only column which matches with above data
merge_df
```

[101]:
```
       Ind_ID GENDER Car_Owner Propert_Owner  CHILDREN  Annual_income  \
0     5008827      M         Y             Y         0       180000.0
1     5009744      F         Y             N         0       315000.0
2     5009746      F         Y             N         0       315000.0
3     5009749      F         Y             N         0            NaN
4     5009752      F         Y             N         0       315000.0
…        …     …         …           …        …            …
1543  5028645      F         N             Y         0            NaN
1544  5023655      F         N             N         0       225000.0
1545  5115992      M         Y             Y         2       180000.0
1546  5118219      M         Y             N         0       270000.0
1547  5053790      F         Y             Y         0       225000.0

               Type_Income                      EDUCATION  \
0                Pensioner               Higher education
1     Commercial associate               Higher education
2     Commercial associate               Higher education
3     Commercial associate               Higher education
4     Commercial associate               Higher education
…                    …                            …
1543  Commercial associate               Higher education
1544  Commercial associate              Incomplete higher
1545               Working               Higher education
1546               Working  Secondary / secondary special
1547               Working               Higher education

            Marital_status       Housing_type  Birthday_count  Employed_days  \
0                  Married  House / apartment        -18772.0         365243
1                  Married  House / apartment        -13557.0           -586
2                  Married  House / apartment             NaN           -586
3                  Married  House / apartment        -13557.0           -586
4                  Married  House / apartment        -13557.0           -586
…                    …                 …              …             …
```

```
1543             Married  House / apartment       -11957.0          -2182
1544  Single / not married  House / apartment       -10229.0          -1209
1545             Married  House / apartment       -13174.0          -2477
1546       Civil marriage  House / apartment       -15292.0           -645
1547             Married  House / apartment       -16601.0          -2859

      Mobile_phone  Work_Phone  Phone  EMAIL_ID Type_Occupation  \
0                1           0      0         0             NaN
1                1           1      1         0             NaN
2                1           1      1         0             NaN
3                1           1      1         0             NaN
4                1           1      1         0             NaN
…              …         …    …       …           …
1543             1           0      0         0        Managers
1544             1           0      0         0     Accountants
1545             1           0      0         0        Managers
1546             1           1      1         0         Drivers
1547             1           0      0         0             NaN

      Family_Members  label
0                  2      1
1                  2      1
2                  2      1
3                  2      1
4                  2      1
…                …    …
1543               2      0
1544               1      0
1545               4      0
1546               2      0
1547               2      0

[1548 rows x 19 columns]
```

[102]:
```python
merge_df.shape
```

[102]:
```
(1548, 19)
```

[103]:
```python
merge_df.describe()
```

[103]:
```
             Ind_ID     CHILDREN  Annual_income  Birthday_count  \
count  1.548000e+03  1548.000000   1.525000e+03     1526.000000
mean   5.078920e+06     0.412791   1.913993e+05   -16040.342071
std    4.171759e+04     0.776691   1.132530e+05     4229.503202
min    5.008827e+06     0.000000   3.375000e+04   -24946.000000
25%    5.045070e+06     0.000000   1.215000e+05   -19553.000000
50%    5.078842e+06     0.000000   1.665000e+05   -15661.500000
```

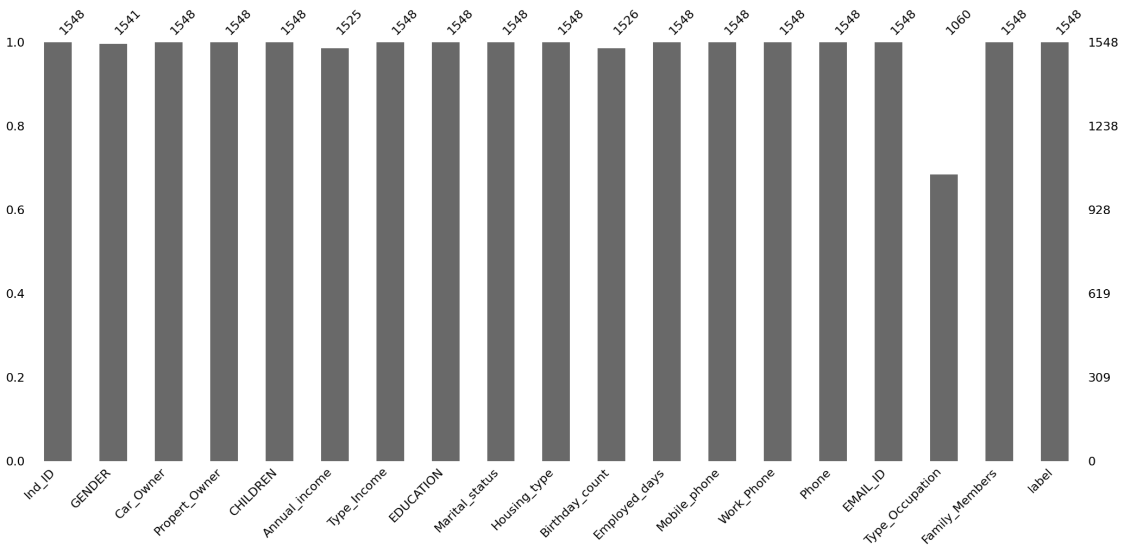
```
75%    5.115673e+06     1.000000   2.250000e+05   -12417.000000
max    5.150412e+06    14.000000   1.575000e+06    -7705.000000

       Employed_days  Mobile_phone   Work_Phone        Phone     EMAIL_ID  \
count    1548.000000        1548.0  1548.000000  1548.000000  1548.000000
mean    59364.689922           1.0     0.208010     0.309432     0.092377
std    137808.062701           0.0     0.406015     0.462409     0.289651
min    -14887.000000           1.0     0.000000     0.000000     0.000000
25%     -3174.500000           1.0     0.000000     0.000000     0.000000
50%     -1565.000000           1.0     0.000000     0.000000     0.000000
75%      -431.750000           1.0     0.000000     1.000000     0.000000
max    365243.000000           1.0     1.000000     1.000000     1.000000

       Family_Members        label
count     1548.000000  1548.000000
mean         2.161499     0.113049
std          0.947772     0.316755
min          1.000000     0.000000
25%          2.000000     0.000000
50%          2.000000     0.000000
75%          3.000000     0.000000
max         15.000000     1.000000
```

[104]:
```
merge_df.isnull().sum()/len(merge_df)*100  ## to check missing values percentage
```

[104]:
```
Ind_ID              0.000000
GENDER              0.452196
Car_Owner           0.000000
Propert_Owner       0.000000
CHILDREN            0.000000
Annual_income       1.485788
Type_Income         0.000000
EDUCATION           0.000000
Marital_status      0.000000
Housing_type        0.000000
Birthday_count      1.421189
Employed_days       0.000000
Mobile_phone        0.000000
Work_Phone          0.000000
Phone               0.000000
EMAIL_ID            0.000000
Type_Occupation    31.524548
Family_Members      0.000000
label               0.000000
dtype: float64
```

[105]:
```python
import missingno as msno   ## importing library to handle missing values
msno.bar(merge_df)
```

[105]:
```
<Axes: >
```



[106]:
```python
merge_df.info() ## to check the datatype of columns
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1548 entries, 0 to 1547
Data columns (total 19 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Ind_ID           1548 non-null   int64
 1   GENDER           1541 non-null   object
 2   Car_Owner        1548 non-null   object
 3   Propert_Owner    1548 non-null   object
 4   CHILDREN         1548 non-null   int64
 5   Annual_income    1525 non-null   float64
 6   Type_Income      1548 non-null   object
 7   EDUCATION        1548 non-null   object
 8   Marital_status   1548 non-null   object
 9   Housing_type     1548 non-null   object
 10  Birthday_count   1526 non-null   float64
 11  Employed_days    1548 non-null   int64
 12  Mobile_phone     1548 non-null   int64
 13  Work_Phone       1548 non-null   int64
 14  Phone            1548 non-null   int64
 15  EMAIL_ID         1548 non-null   int64
 16  Type_Occupation  1060 non-null   object
```

```
 17  Family_Members   1548 non-null   int64
 18  label            1548 non-null   int64
dtypes: float64(2), int64(9), object(8)
memory usage: 241.9+ KB
```

1 #### Type_occupation is a categorical column so mode imputation is used to fill missings.bold text

[107]:
```
merge_df['Type_Occupation']=merge_df['Type_Occupation'].
↪fillna(merge_df['Type_Occupation'].mode()[0])
```

[108]:
```
merge_df.isnull().sum() ### for checking the sum of null data still available␣
↪in data
```

[108]:
```
Ind_ID              0
GENDER              7
Car_Owner           0
Propert_Owner       0
CHILDREN            0
Annual_income      23
Type_Income         0
EDUCATION           0
Marital_status      0
Housing_type        0
Birthday_count     22
Employed_days       0
Mobile_phone        0
Work_Phone          0
Phone               0
EMAIL_ID            0
Type_Occupation     0
Family_Members      0
label               0
dtype: int64
```

since rest of the three col has less than 2 % miss values so the rows can be deleted.

[109]:
```
merge_df = merge_df.dropna() ## to drop the rows which have even one missing␣
↪values
```

[110]:
```
merge_df.isnull().sum() ### for checking the null values available in data
```

[110]:
```
Ind_ID             0
GENDER             0
Car_Owner          0
Propert_Owner      0
CHILDREN           0
```

```
Annual_income      0
Type_Income        0
EDUCATION          0
Marital_status     0
Housing_type       0
Birthday_count     0
Employed_days      0
Mobile_phone       0
Work_Phone         0
Phone              0
EMAIL_ID           0
Type_Occupation    0
Family_Members     0
label              0
dtype: int64
```

[111]:
```
merge_df.describe()
```

[111]:
```
             Ind_ID     CHILDREN  Annual_income  Birthday_count  \
count  1.496000e+03  1496.000000   1.496000e+03     1496.000000
mean   5.079217e+06     0.415775   1.907750e+05   -16036.192513
std    4.168109e+04     0.780784   1.131384e+05     4226.506557
min    5.008827e+06     0.000000   3.375000e+04   -24946.000000
25%    5.045349e+06     0.000000   1.210500e+05   -19543.000000
50%    5.079010e+06     0.000000   1.660500e+05   -15686.000000
75%    5.115801e+06     1.000000   2.250000e+05   -12417.000000
max    5.150412e+06    14.000000   1.575000e+06    -7705.000000

       Employed_days  Mobile_phone   Work_Phone        Phone     EMAIL_ID  \
count    1496.000000        1496.0  1496.000000  1496.000000  1496.000000
mean    59290.681818           1.0     0.205882     0.304813     0.094251
std    137766.774169           0.0     0.404480     0.460482     0.292276
min    -14887.000000           1.0     0.000000     0.000000     0.000000
25%     -3229.250000           1.0     0.000000     0.000000     0.000000
50%     -1575.500000           1.0     0.000000     0.000000     0.000000
75%      -431.000000           1.0     0.000000     1.000000     0.000000
max    365243.000000           1.0     1.000000     1.000000     1.000000

       Family_Members        label
count     1496.000000  1496.000000
mean         2.165107     0.106952
std          0.951752     0.309155
min          1.000000     0.000000
25%          2.000000     0.000000
50%          2.000000     0.000000
75%          3.000000     0.000000
max         15.000000     1.000000
```

[112]:
```python
merge_df['Employed_days'].replace(365243,np.nan,inplace=True) ## replace with␣
↪Nan where this value present
```

```
<ipython-input-112-d2fccb8d0f41>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  merge_df['Employed_days'].replace(365243,np.nan,inplace=True) ## replace with
Nan where this value present
```

[112]:
```python

```

[112]:
```python

```

[231]:
```python
merge_df.isnull().sum()
```

[231]:
```
Ind_ID              0
GENDER              0
Car_Owner           0
Propert_Owner       0
CHILDREN            0
Type_Income         0
EDUCATION           0
Marital_status      0
Housing_type        0
Birthday_count      0
Employed_days       0
Mobile_phone        0
Work_Phone          0
Phone               0
EMAIL_ID            0
Type_Occupation     0
Family_Members      0
label               0
Education_order     0
Annual_income_log   0
Employed_days_ex    0
dtype: int64
```

[114]:
```python
merge_df['Employed_days'].median()  ## find median of this column
```

[114]:
```
-1979.5
```

[232]:
```python

```

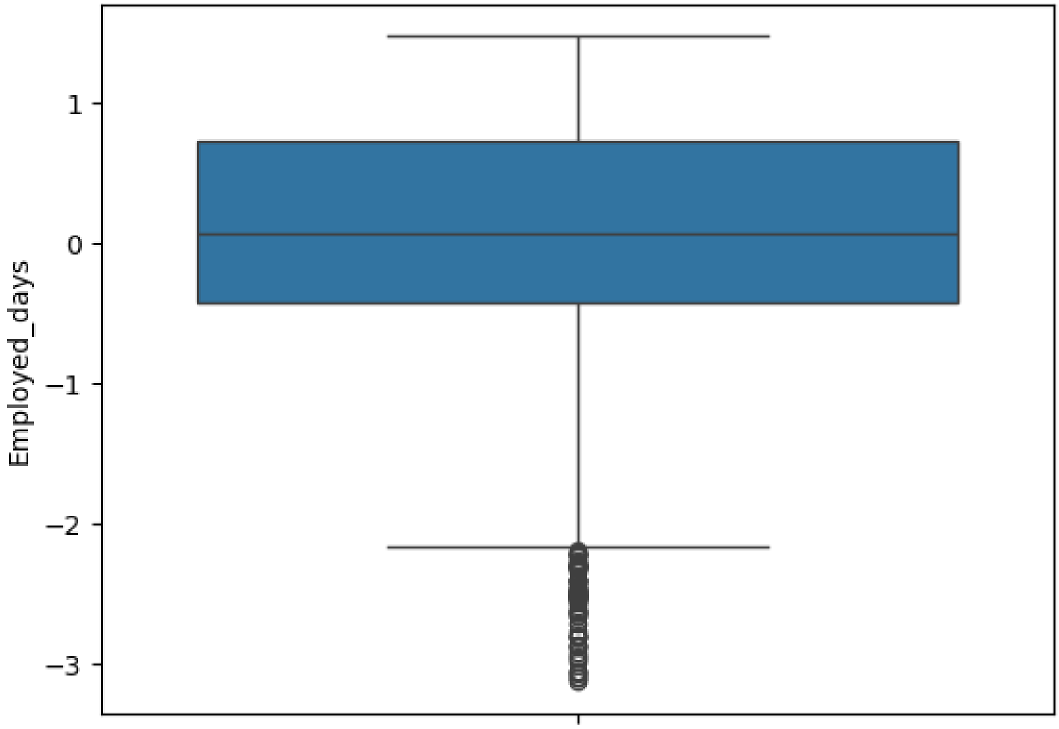
```
merge_df['Employed_days']=merge_df['Employed_days'].
↪fillna(merge_df['Employed_days'].median())# as there is an outliers in this␣
↪column so we are going to replace the value with median value because null␣
↪percentage in this column is less than 30%
```

[116]:
```
merge_df.isnull().sum()
```

[116]:
```
Ind_ID             0
GENDER             0
Car_Owner          0
Propert_Owner      0
CHILDREN           0
Annual_income      0
Type_Income        0
EDUCATION          0
Marital_status     0
Housing_type       0
Birthday_count     0
Employed_days      0
Mobile_phone       0
Work_Phone         0
Phone              0
EMAIL_ID           0
Type_Occupation    0
Family_Members     0
label              0
dtype: int64
```

[234]:
```
import seaborn as sns

sns.boxplot(merge_df['Employed_days'])
```

[234]:
```
<Axes: ylabel='Employed_days'>
```



[118]:
```
### Outlier removal using IQR
```

[119]:
```
Q1=merge_df['Employed_days'].quantile(.25)
Q3=merge_df['Employed_days'].quantile(.75)
```

[120]:
```
Q1,Q3
```

[120]: (-3229.25, -1169.5)

[121]:
```
IQR = Q3-Q1
```

[122]:
```
IQR
```

[122]: 2059.75

[123]:
```
#lower Limit
# Upper limit

ll = (Q1-1.5*(IQR))
ll
```

[123]: -6318.875

[124]:
```
Ul = Q3 + 1.5*IQR
Ul
```

[124]: 1920.125

[125]:
```
ll,Ul
```

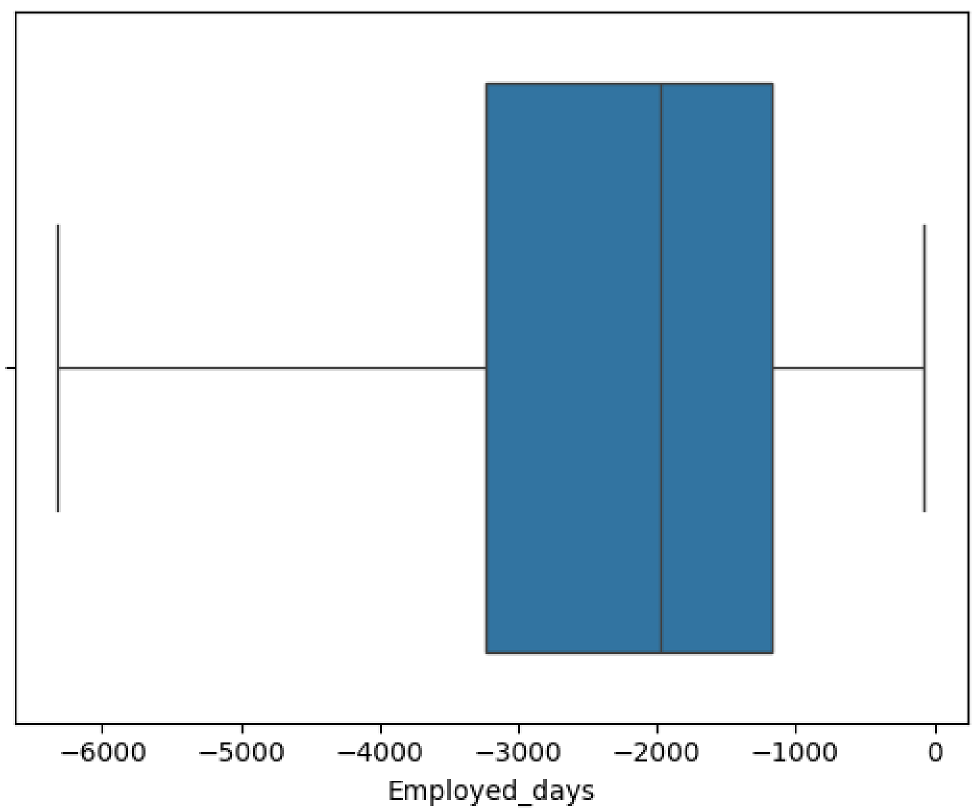
[125]: (-6318.875, 1920.125)

[126]:
```
merge_df['Employed_days']  =  merge_df['Employed_days'].clip(lower = -6318.875 ␣
↪, upper = 1920.125)
```

```
<ipython-input-126-045dd7d9d3db>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  merge_df['Employed_days']  =  merge_df['Employed_days'].clip(lower = -6318.875
, upper = 1920.125)
```

[127]:
```
sns.boxplot(x = 'Employed_days', data = merge_df)###__ checking for outliers
```

[127]: <Axes: xlabel='Employed_days'>



[128]:
```
merge_df = merge_df[(merge_df['Employed_days'] > ll) &
↪(merge_df['Employed_days'] < Ul)] ## select the rows greater than lower
↪fence and

↪    ## less than higher fence.
```

[128]:
```
```

[129]:
```
merge_df
```

[129]:
```
        Ind_ID GENDER Car_Owner Propert_Owner  CHILDREN  Annual_income  \
0      5008827      M         Y             Y         0       180000.0
1      5009744      F         Y             N         0       315000.0
4      5009752      F         Y             N         0       315000.0
6      5009754      F         Y             N         0       315000.0
7      5009894      F         N             N         0       180000.0
…          …      …         …             …         …              …
1542   5118268      M         Y             N         1       360000.0
1544   5023655      F         N             N         0       225000.0
1545   5115992      M         Y             Y         2       180000.0
1546   5118219      M         Y             N         0       270000.0
1547   5053790      F         Y             Y         0       225000.0

               Type_Income                      EDUCATION  \
0                Pensioner               Higher education
1     Commercial associate               Higher education
4     Commercial associate               Higher education
6     Commercial associate               Higher education
7                Pensioner  Secondary / secondary special
…                        …                              …
1542         State servant  Secondary / secondary special
1544  Commercial associate              Incomplete higher
1545               Working               Higher education
1546               Working  Secondary / secondary special
1547               Working               Higher education

            Marital_status       Housing_type  Birthday_count  Employed_days  \
0                  Married  House / apartment        -18772.0        -1979.5
1                  Married  House / apartment        -13557.0         -586.0
4                  Married  House / apartment        -13557.0         -586.0
6                  Married  House / apartment        -13557.0         -586.0
7                  Married  House / apartment        -22134.0        -1979.5
…                        …                  …               …              …
1542               Married  House / apartment        -11294.0        -3536.0
```

```
1544  Single / not married  House / apartment       -10229.0       -1209.0
1545               Married  House / apartment       -13174.0       -2477.0
1546        Civil marriage  House / apartment       -15292.0        -645.0
1547               Married  House / apartment       -16601.0       -2859.0

      Mobile_phone  Work_Phone  Phone  EMAIL_ID Type_Occupation  \
0                1           0      0         0        Laborers
1                1           1      1         0        Laborers
4                1           1      1         0        Laborers
6                1           1      1         0        Laborers
7                1           0      0         0        Laborers
…              …         …    …       …             …
1542             1           0      1         0         Drivers
1544             1           0      0         0     Accountants
1545             1           0      0         0        Managers
1546             1           1      1         0         Drivers
1547             1           0      0         0        Laborers

      Family_Members  label
0                  2      1
1                  2      1
4                  2      1
6                  2      1
7                  2      1
…                …    …
1542               3      0
1544               1      0
1545               4      0
1546               2      0
1547               2      0

[1382 rows x 19 columns]
```

[130]:
```
merge_df.describe()
```

[130]:
```
             Ind_ID     CHILDREN  Annual_income  Birthday_count  \
count  1.382000e+03  1382.000000   1.382000e+03     1382.000000
mean   5.079363e+06     0.413169   1.920582e+05   -15886.369754
std    4.175350e+04     0.779480   1.159408e+05     4302.343169
min    5.008827e+06     0.000000   3.375000e+04   -24946.000000
25%    5.045286e+06     0.000000   1.170000e+05   -19480.000000
50%    5.078883e+06     0.000000   1.665000e+05   -15372.000000
75%    5.115914e+06     1.000000   2.250000e+05   -12250.750000
max    5.150412e+06    14.000000   1.575000e+06    -7705.000000

       Employed_days  Mobile_phone    Work_Phone         Phone      EMAIL_ID  \
count    1382.000000        1382.0  1382.000000  1382.000000  1382.000000
```

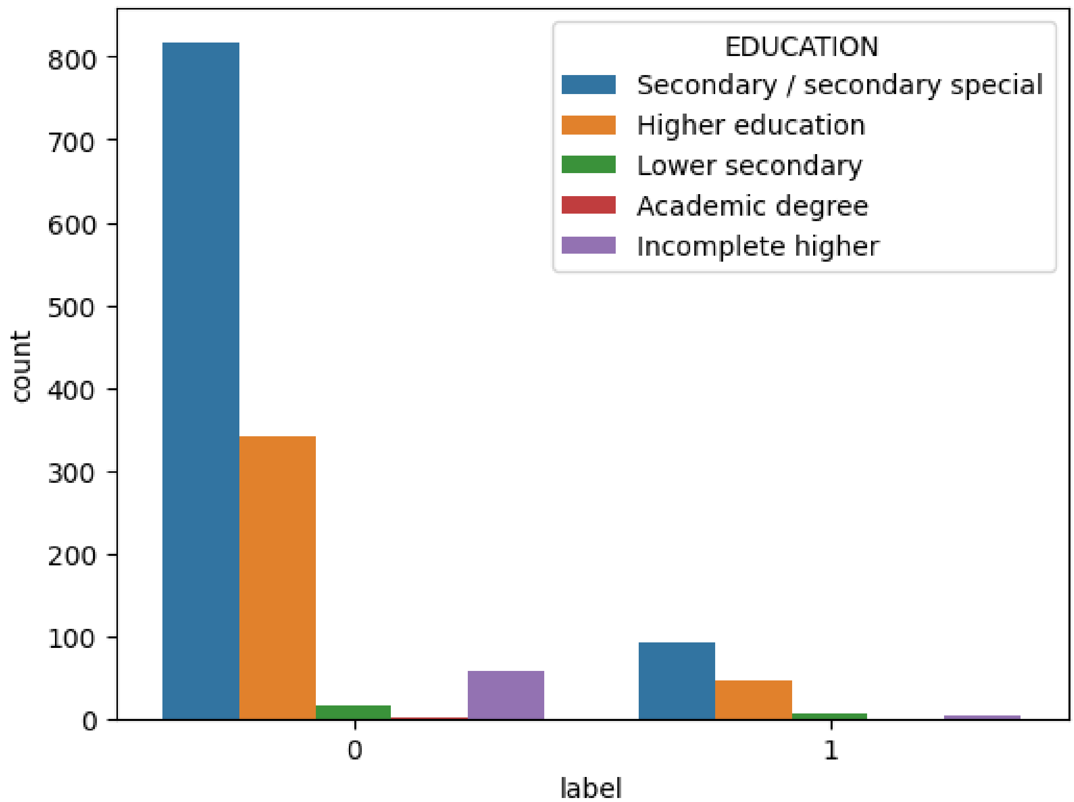
```
mean   -2075.330680          1.0     0.199711     0.298842     0.098408
std     1354.678666          0.0     0.399927     0.457916     0.297973
min    -6317.000000          1.0     0.000000     0.000000     0.000000
25%    -2654.000000          1.0     0.000000     0.000000     0.000000
50%    -1979.500000          1.0     0.000000     0.000000     0.000000
75%    -1081.000000          1.0     0.000000     1.000000     0.000000
max      -73.000000          1.0     1.000000     1.000000     1.000000

       Family_Members        label
count     1382.000000  1382.000000
mean         2.162084     0.108538
std          0.950511     0.311172
min          1.000000     0.000000
25%          2.000000     0.000000
50%          2.000000     0.000000
75%          3.000000     0.000000
max         15.000000     1.000000
```

Now the data has no missing values and outliers so the data can be exported to CSV file for SQL analysis .

[131]:
```
merge_df.to_csv(r'D:\New folder\cleaned_data.csv', index=False)
```

Gender vs Car owner

[132]:
```
sns.countplot(x='label',hue='EDUCATION',data=merge_df)# ___ showing relation␣
↪between credit label and education
plt.show()
```



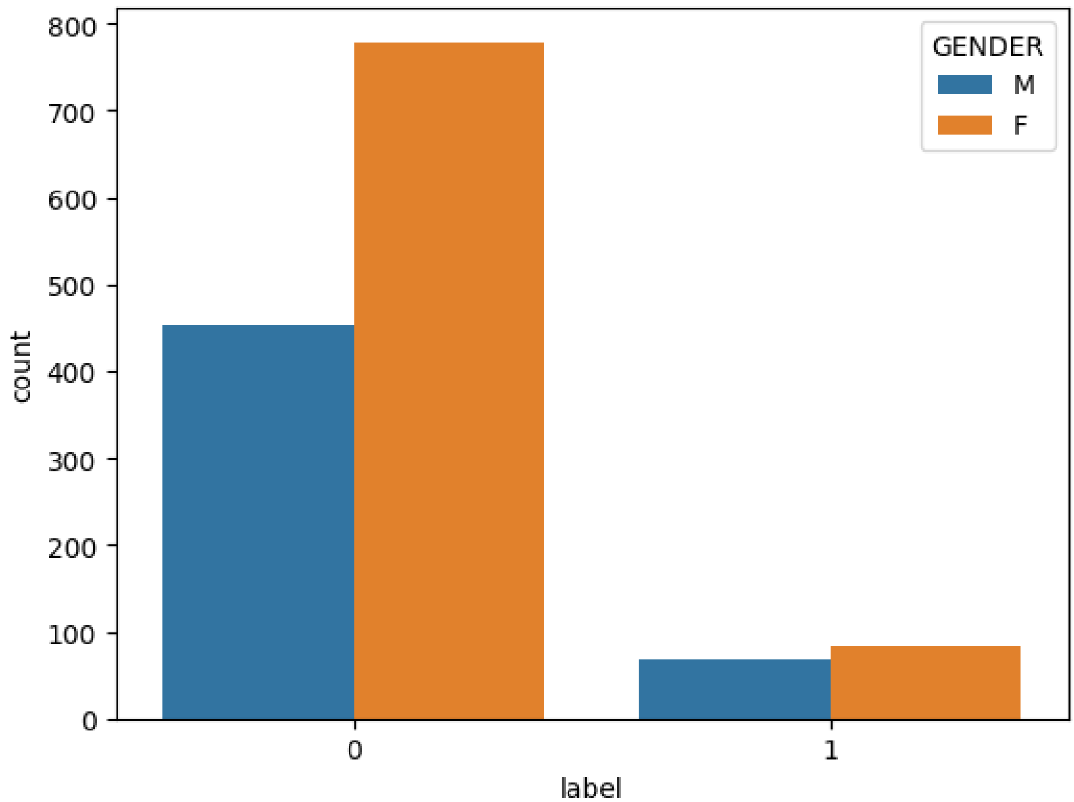
[133]:
```
merge_df.GENDER.value_counts()
```

[133]:
```
F    862
M    520
Name: GENDER, dtype: int64
```

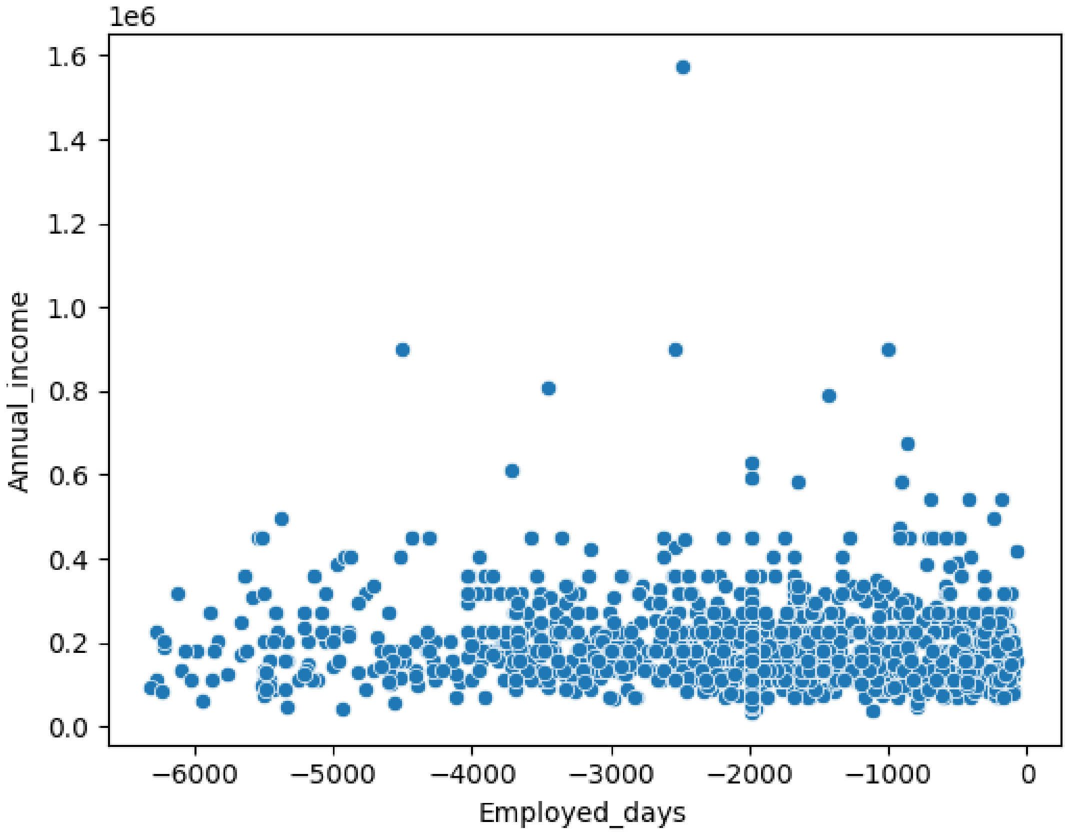
[134]:
```
pd.crosstab(merge_df.label, merge_df.GENDER)
```

[134]:
```
GENDER    F    M
label
0       779  453
1        83   67
```

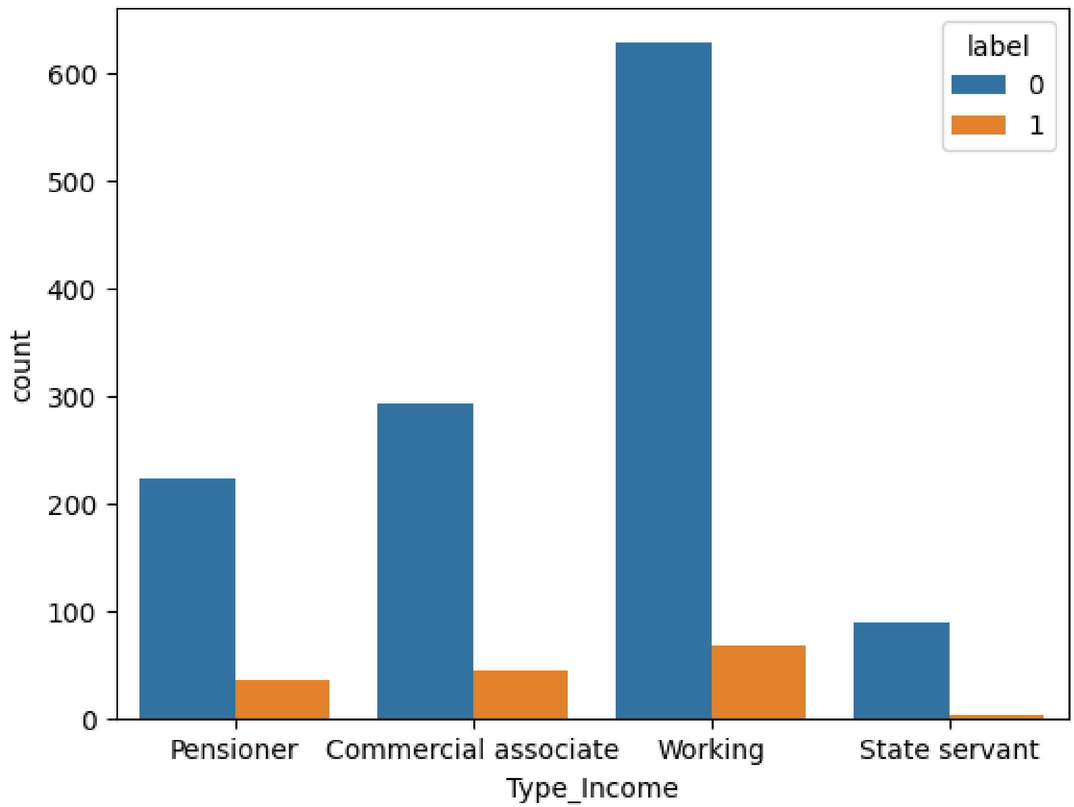
[135]:
```
sns.countplot(x='label',hue='GENDER',data=merge_df)###  with this graph we can␣
↪clearly visualise that more no of  females have good credit as compare to␣
↪males
plt.show()
```




```
[136]: sns.scatterplot(y='Annual_income',x = 'Employed_days',data =merge_df)### here
       ↪we can clearly observe that as the working days is increasing  annualincome
       ↪is getting decrease
       plt.show()
```



[137]:
```
sns.countplot(x = 'Type_Income', hue ='label',data = merge_df)
plt.show()
```



[138]:
```
merge_df.Type_Income.value_counts()
```

[138]:
```
Working                 696
Commercial associate    337
Pensioner               258
State servant            91
Name: Type_Income, dtype: int64
```

[139]:
```
pd.crosstab(merge_df.label ,merge_df.Type_Income) ## commercial associate and␣
↪Pensioner has good credit score  as compare to state servant and working one␣
↪and they will be more likely to repay the loan
```

[139]:
```
Type_Income  Commercial associate  Pensioner  State servant  Working
label
0                             293        222             88      629
1                              44         36              3       67
```

[140]:
```
merge_df
```

[140]:
```
       Ind_ID GENDER Car_Owner Propert_Owner  CHILDREN  Annual_income  \
0     5008827      M         Y             Y         0       180000.0
```

```
1     5009744      F         Y             N         0       315000.0
4     5009752      F         Y             N         0       315000.0
6     5009754      F         Y             N         0       315000.0
7     5009894      F         N             N         0       180000.0
…       …    …        …            …       …            …
1542  5118268      M         Y             N         1       360000.0
1544  5023655      F         N             N         0       225000.0
1545  5115992      M         Y             Y         2       180000.0
1546  5118219      M         Y             N         0       270000.0
1547  5053790      F         Y             Y         0       225000.0

               Type_Income                      EDUCATION  \
0                Pensioner               Higher education
1     Commercial associate               Higher education
4     Commercial associate               Higher education
6     Commercial associate               Higher education
7                Pensioner  Secondary / secondary special
…                  …                            …
1542         State servant  Secondary / secondary special
1544  Commercial associate              Incomplete higher
1545               Working               Higher education
1546               Working  Secondary / secondary special
1547               Working               Higher education

            Marital_status       Housing_type  Birthday_count  Employed_days  \
0                  Married  House / apartment        -18772.0        -1979.5
1                  Married  House / apartment        -13557.0         -586.0
4                  Married  House / apartment        -13557.0         -586.0
6                  Married  House / apartment        -13557.0         -586.0
7                  Married  House / apartment        -22134.0        -1979.5
…                    …                …              …             …
1542               Married  House / apartment        -11294.0        -3536.0
1544  Single / not married  House / apartment        -10229.0        -1209.0
1545               Married  House / apartment        -13174.0        -2477.0
1546        Civil marriage  House / apartment        -15292.0         -645.0
1547               Married  House / apartment        -16601.0        -2859.0

      Mobile_phone  Work_Phone  Phone  EMAIL_ID Type_Occupation  \
0                1           0      0         0        Laborers
1                1           1      1         0        Laborers
4                1           1      1         0        Laborers
6                1           1      1         0        Laborers
7                1           0      0         0        Laborers
…            …         …    …       …             …
1542             1           0      1         0         Drivers
1544             1           0      0         0     Accountants
1545             1           0      0         0        Managers
```

```
1546                1            1       1          0          Drivers
1547                1            0       0          0         Laborers

      Family_Members  label
0                  2      1
1                  2      1
4                  2      1
6                  2      1
7                  2      1
…               …     …
1542               3      0
1544               1      0
1545               4      0
1546               2      0
1547               2      0

[1382 rows x 19 columns]
```

DATA WRANGLING

converting the categorical text data into numerical data

[141]:
```
merge_df['EDUCATION'].unique()
```

[141]:
```
array(['Higher education', 'Secondary / secondary special',
       'Lower secondary', 'Incomplete higher', 'Academic degree'],
      dtype=object)
```

[142]:
```
## Ordinal ENCODING

from sklearn.preprocessing import OrdinalEncoder ###
oe = OrdinalEncoder(categories = [['Lower secondary','Secondary / secondary␣
↪special','Incomplete higher','Higher education','Academic degree']])
merge_df['Education_order'] = oe.fit_transform(merge_df[['EDUCATION']])
```

1.1 Nominal ENCODING

merge_df['enc'] = pd.get_dummies(merge_df['Type_Income'],drop_first=True)

[143]:
```
enc = pd.get_dummies(merge_df['Type_Income'],drop_first=True)
```

[144]:
```
data = pd.concat([merge_df,enc],axis=1)
```

[145]:
```
data.head()
```

[145]:
```
    Ind_ID GENDER Car_Owner Propert_Owner  CHILDREN  Annual_income  \
0  5008827      M         Y             Y         0       180000.0
1  5009744      F         Y             N         0       315000.0
```

```
4  5009752      F         Y             N        0       315000.0
6  5009754      F         Y             N        0       315000.0
7  5009894      F         N             N        0       180000.0

            Type_Income                      EDUCATION Marital_status  \
0             Pensioner               Higher education        Married
1  Commercial associate               Higher education        Married
4  Commercial associate               Higher education        Married
6  Commercial associate               Higher education        Married
7             Pensioner  Secondary / secondary special        Married

        Housing_type  … Work_Phone  Phone  EMAIL_ID  Type_Occupation  \
0  House / apartment  …          0      0         0         Laborers
1  House / apartment  …          1      1         0         Laborers
4  House / apartment  …          1      1         0         Laborers
6  House / apartment  …          1      1         0         Laborers
7  House / apartment  …          0      0         0         Laborers

   Family_Members  label Education_order  Pensioner  State servant  Working
0               2      1             3.0          1              0        0
1               2      1             3.0          0              0        0
4               2      1             3.0          0              0        0
6               2      1             3.0          0              0        0
7               2      1             1.0          1              0        0

[5 rows x 23 columns]
```

[146]:
```
data.drop(columns=['Type_Income','EDUCATION'],axis=1,inplace=True)
```

[147]:
```
data.head(
)
```

[147]:
```
    Ind_ID GENDER Car_Owner Propert_Owner  CHILDREN  Annual_income  \
0  5008827      M         Y             Y        0       180000.0
1  5009744      F         Y             N        0       315000.0
4  5009752      F         Y             N        0       315000.0
6  5009754      F         Y             N        0       315000.0
7  5009894      F         N             N        0       180000.0

  Marital_status       Housing_type  Birthday_count  Employed_days  …  \
0        Married  House / apartment        -18772.0        -1979.5  …
1        Married  House / apartment        -13557.0         -586.0  …
4        Married  House / apartment        -13557.0         -586.0  …
6        Married  House / apartment        -13557.0         -586.0  …
7        Married  House / apartment        -22134.0        -1979.5  …

   Work_Phone  Phone  EMAIL_ID  Type_Occupation Family_Members  label  \
```

```
0          0     0        0       Laborers            2      1
1          1     1        0       Laborers            2      1
4          1     1        0       Laborers            2      1
6          1     1        0       Laborers            2      1
7          0     0        0       Laborers            2      1

   Education_order  Pensioner  State servant  Working
0              3.0          1              0        0
1              3.0          0              0        0
4              3.0          0              0        0
6              3.0          0              0        0
7              1.0          1              0        0

[5 rows x 21 columns]
```

[148]:
```
dummy_data = pd.
↪get_dummies(data[['GENDER','Car_Owner','Propert_Owner','Marital_status','Housing_type','Typ
↪= True)###  categorical encoding for coverting object data type to int
```

[149]:
```
dummy_data.head()
```

[149]:
```
   GENDER_M  Car_Owner_Y  Propert_Owner_Y  Marital_status_Married  \
0         1            1                1                       1
1         0            1                0                       1
4         0            1                0                       1
6         0            1                0                       1
7         0            0                0                       1

   Marital_status_Separated  Marital_status_Single / not married  \
0                         0                                    0
1                         0                                    0
4                         0                                    0
6                         0                                    0
7                         0                                    0

   Marital_status_Widow  Housing_type_House / apartment  \
0                     0                               1
1                     0                               1
4                     0                               1
6                     0                               1
7                     0                               1

   Housing_type_Municipal apartment  Housing_type_Office apartment  …  \
0                                 0                              0  …
1                                 0                              0  …
4                                 0                              0  …
6                                 0                              0  …
```

```
7                                0                              0  …

   Type_Occupation_Laborers  Type_Occupation_Low-skill Laborers  \
0                         1                                   0
1                         1                                   0
4                         1                                   0
6                         1                                   0
7                         1                                   0

   Type_Occupation_Managers  Type_Occupation_Medicine staff  \
0                         0                               0
1                         0                               0
4                         0                               0
6                         0                               0
7                         0                               0

   Type_Occupation_Private service staff  Type_Occupation_Realty agents  \
0                                      0                              0
1                                      0                              0
4                                      0                              0
6                                      0                              0
7                                      0                              0

   Type_Occupation_Sales staff  Type_Occupation_Secretaries  \
0                            0                            0
1                            0                            0
4                            0                            0
6                            0                            0
7                            0                            0

   Type_Occupation_Security staff  Type_Occupation_Waiters/barmen staff
0                               0                                     0
1                               0                                     0
4                               0                                     0
6                               0                                     0
7                               0                                     0

[5 rows x 29 columns]
```

[150]:
```
data_final = pd.concat([data,dummy_data],axis = 1)## by using concat function␣
 ↪joining to data
```

[151]:
```
data_final.head()
```

[151]:
```
     Ind_ID GENDER Car_Owner Propert_Owner  CHILDREN  Annual_income  \
0   5008827      M         Y             Y         0       180000.0
1   5009744      F         Y             N         0       315000.0
```

```
4  5009752      F         Y             N         0       315000.0
6  5009754      F         Y             N         0       315000.0
7  5009894      F         N             N         0       180000.0

  Marital_status       Housing_type  Birthday_count  Employed_days  …  \
0        Married  House / apartment        -18772.0        -1979.5  …
1        Married  House / apartment        -13557.0         -586.0  …
4        Married  House / apartment        -13557.0         -586.0  …
6        Married  House / apartment        -13557.0         -586.0  …
7        Married  House / apartment        -22134.0        -1979.5  …

   Type_Occupation_Laborers  Type_Occupation_Low-skill Laborers  \
0                         1                                   0
1                         1                                   0
4                         1                                   0
6                         1                                   0
7                         1                                   0

   Type_Occupation_Managers  Type_Occupation_Medicine staff  \
0                         0                               0
1                         0                               0
4                         0                               0
6                         0                               0
7                         0                               0

  Type_Occupation_Private service staff  Type_Occupation_Realty agents  \
0                                     0                              0
1                                     0                              0
4                                     0                              0
6                                     0                              0
7                                     0                              0

   Type_Occupation_Sales staff  Type_Occupation_Secretaries  \
0                            0                            0
1                            0                            0
4                            0                            0
6                            0                            0
7                            0                            0

   Type_Occupation_Security staff  Type_Occupation_Waiters/barmen staff
0                               0                                     0
1                               0                                     0
4                               0                                     0
6                               0                                     0
7                               0                                     0

[5 rows x 50 columns]
```

[152]:
```
data_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1382 entries, 0 to 1547
Data columns (total 50 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Ind_ID                                1382 non-null   int64
 1   GENDER                                1382 non-null   object
 2   Car_Owner                             1382 non-null   object
 3   Propert_Owner                         1382 non-null   object
 4   CHILDREN                              1382 non-null   int64
 5   Annual_income                         1382 non-null   float64
 6   Marital_status                        1382 non-null   object
 7   Housing_type                          1382 non-null   object
 8   Birthday_count                        1382 non-null   float64
 9   Employed_days                         1382 non-null   float64
 10  Mobile_phone                          1382 non-null   int64
 11  Work_Phone                            1382 non-null   int64
 12  Phone                                 1382 non-null   int64
 13  EMAIL_ID                              1382 non-null   int64
 14  Type_Occupation                       1382 non-null   object
 15  Family_Members                        1382 non-null   int64
 16  label                                 1382 non-null   int64
 17  Education_order                       1382 non-null   float64
 18  Pensioner                             1382 non-null   uint8
 19  State servant                         1382 non-null   uint8
 20  Working                               1382 non-null   uint8
 21  GENDER_M                              1382 non-null   uint8
 22  Car_Owner_Y                           1382 non-null   uint8
 23  Propert_Owner_Y                       1382 non-null   uint8
 24  Marital_status_Married                1382 non-null   uint8
 25  Marital_status_Separated              1382 non-null   uint8
 26  Marital_status_Single / not married   1382 non-null   uint8
 27  Marital_status_Widow                  1382 non-null   uint8
 28  Housing_type_House / apartment        1382 non-null   uint8
 29  Housing_type_Municipal apartment      1382 non-null   uint8
 30  Housing_type_Office apartment         1382 non-null   uint8
 31  Housing_type_Rented apartment         1382 non-null   uint8
 32  Housing_type_With parents             1382 non-null   uint8
 33  Type_Occupation_Cleaning staff        1382 non-null   uint8
 34  Type_Occupation_Cooking staff         1382 non-null   uint8
 35  Type_Occupation_Core staff            1382 non-null   uint8
 36  Type_Occupation_Drivers               1382 non-null   uint8
 37  Type_Occupation_HR staff              1382 non-null   uint8
 38  Type_Occupation_High skill tech staff 1382 non-null   uint8
 39  Type_Occupation_IT staff              1382 non-null   uint8
```

```
 40  Type_Occupation_Laborers               1382 non-null   uint8
 41  Type_Occupation_Low-skill Laborers     1382 non-null   uint8
 42  Type_Occupation_Managers               1382 non-null   uint8
 43  Type_Occupation_Medicine staff         1382 non-null   uint8
 44  Type_Occupation_Private service staff  1382 non-null   uint8
 45  Type_Occupation_Realty agents          1382 non-null   uint8
 46  Type_Occupation_Sales staff            1382 non-null   uint8
 47  Type_Occupation_Secretaries            1382 non-null   uint8
 48  Type_Occupation_Security staff         1382 non-null   uint8
 49  Type_Occupation_Waiters/barmen staff   1382 non-null   uint8
dtypes: float64(4), int64(8), object(6), uint8(32)
memory usage: 248.3+ KB
```

[153]:
```
data_final.
 ↪drop(columns=['GENDER','Car_Owner','Propert_Owner','Marital_status','Housing_type','Type_Oc
```

[154]:
```
data_final.head()
```

[154]:
```
    Ind_ID  CHILDREN  Annual_income  Birthday_count  Employed_days  \
0  5008827         0       180000.0        -18772.0        -1979.5
1  5009744         0       315000.0        -13557.0         -586.0
4  5009752         0       315000.0        -13557.0         -586.0
6  5009754         0       315000.0        -13557.0         -586.0
7  5009894         0       180000.0        -22134.0        -1979.5

   Mobile_phone  Work_Phone  Phone  EMAIL_ID  Family_Members  …  \
0             1           0      0         0               2  …
1             1           1      1         0               2  …
4             1           1      1         0               2  …
6             1           1      1         0               2  …
7             1           0      0         0               2  …

   Type_Occupation_Laborers  Type_Occupation_Low-skill Laborers  \
0                         1                                   0
1                         1                                   0
4                         1                                   0
6                         1                                   0
7                         1                                   0

   Type_Occupation_Managers  Type_Occupation_Medicine staff  \
0                         0                               0
1                         0                               0
4                         0                               0
6                         0                               0
7                         0                               0

   Type_Occupation_Private service staff  Type_Occupation_Realty agents  \
```

```
0                                        0                               0
1                                        0                               0
4                                        0                               0
6                                        0                               0
7                                        0                               0

   Type_Occupation_Sales staff  Type_Occupation_Secretaries  \
0                            0                            0
1                            0                            0
4                            0                            0
6                            0                            0
7                            0                            0

   Type_Occupation_Security staff  Type_Occupation_Waiters/barmen staff
0                               0                                     0
1                               0                                     0
4                               0                                     0
6                               0                                     0
7                               0                                     0

[5 rows x 44 columns]
```

[155]:
```
data_final.info()
```

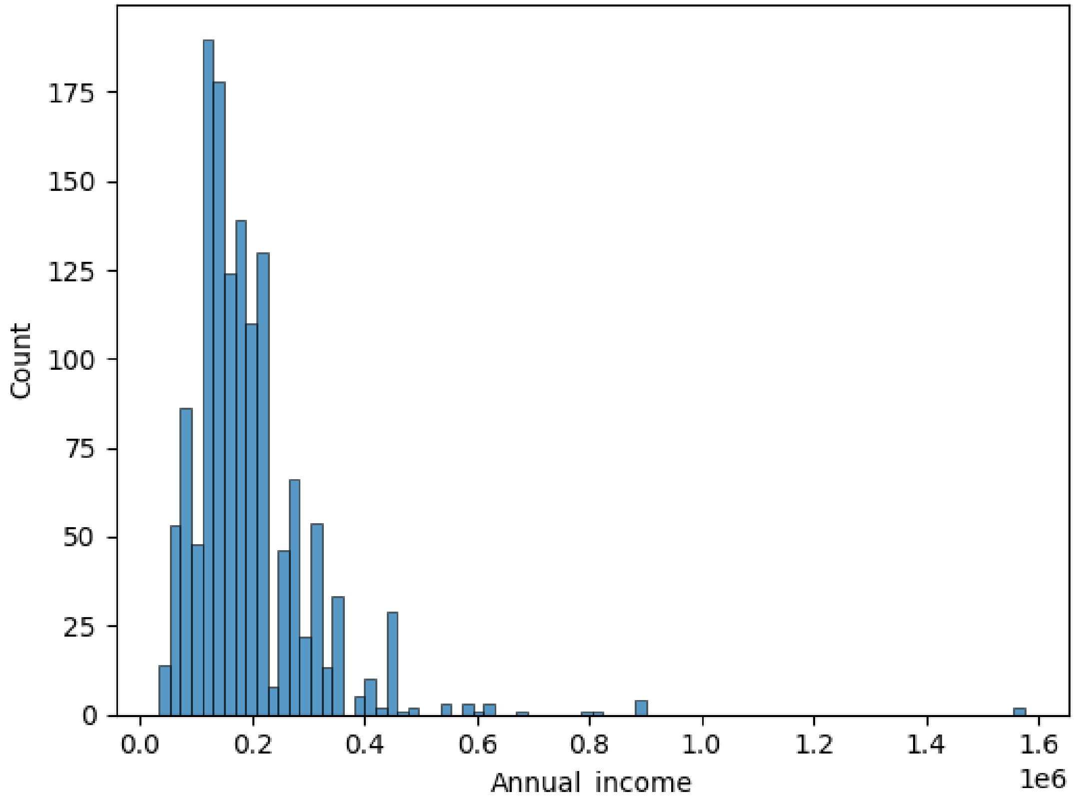
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1382 entries, 0 to 1547
Data columns (total 44 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   Ind_ID                                 1382 non-null   int64
 1   CHILDREN                               1382 non-null   int64
 2   Annual_income                          1382 non-null   float64
 3   Birthday_count                         1382 non-null   float64
 4   Employed_days                          1382 non-null   float64
 5   Mobile_phone                           1382 non-null   int64
 6   Work_Phone                             1382 non-null   int64
 7   Phone                                  1382 non-null   int64
 8   EMAIL_ID                               1382 non-null   int64
 9   Family_Members                         1382 non-null   int64
 10  label                                  1382 non-null   int64
 11  Education_order                        1382 non-null   float64
 12  Pensioner                              1382 non-null   uint8
 13  State servant                          1382 non-null   uint8
 14  Working                                1382 non-null   uint8
 15  GENDER_M                               1382 non-null   uint8
 16  Car_Owner_Y                            1382 non-null   uint8
 17  Propert_Owner_Y                        1382 non-null   uint8
```

```
 18  Marital_status_Married             1382 non-null   uint8
 19  Marital_status_Separated           1382 non-null   uint8
 20  Marital_status_Single / not married     1382 non-null   uint8
 21  Marital_status_Widow               1382 non-null   uint8
 22  Housing_type_House / apartment     1382 non-null   uint8
 23  Housing_type_Municipal apartment   1382 non-null   uint8
 24  Housing_type_Office apartment      1382 non-null   uint8
 25  Housing_type_Rented apartment      1382 non-null   uint8
 26  Housing_type_With parents          1382 non-null   uint8
 27  Type_Occupation_Cleaning staff     1382 non-null   uint8
 28  Type_Occupation_Cooking staff      1382 non-null   uint8
 29  Type_Occupation_Core staff         1382 non-null   uint8
 30  Type_Occupation_Drivers            1382 non-null   uint8
 31  Type_Occupation_HR staff           1382 non-null   uint8
 32  Type_Occupation_High skill tech staff  1382 non-null   uint8
 33  Type_Occupation_IT staff           1382 non-null   uint8
 34  Type_Occupation_Laborers           1382 non-null   uint8
 35  Type_Occupation_Low-skill Laborers     1382 non-null   uint8
 36  Type_Occupation_Managers           1382 non-null   uint8
 37  Type_Occupation_Medicine staff     1382 non-null   uint8
 38  Type_Occupation_Private service staff  1382 non-null   uint8
 39  Type_Occupation_Realty agents      1382 non-null   uint8
 40  Type_Occupation_Sales staff        1382 non-null   uint8
 41  Type_Occupation_Secretaries        1382 non-null   uint8
 42  Type_Occupation_Security staff     1382 non-null   uint8
 43  Type_Occupation_Waiters/barmen staff   1382 non-null   uint8
dtypes: float64(4), int64(8), uint8(32)
memory usage: 183.5 KB
```

[155]:

[156]:
```
sns.histplot(merge_df['Annual_income'])# here we can observe that data is right␣
 ↪skew data
```

[156]: <Axes: xlabel='Annual_income', ylabel='Count'>



[157]:
```python
merge_df['Annual_income'].skew()
```

[157]: 3.9815052254085748

[158]:
```python
merge_df['Annual_income_log'] = np.log(merge_df['Annual_income'])####____␣
↪applying log fuction on data to remove right skewness
```

[159]:
```python
merge_df
```

[159]:
```
        Ind_ID GENDER Car_Owner Propert_Owner  CHILDREN  Annual_income  \
0      5008827      M         Y             Y         0       180000.0
1      5009744      F         Y             N         0       315000.0
4      5009752      F         Y             N         0       315000.0
6      5009754      F         Y             N         0       315000.0
7      5009894      F         N             N         0       180000.0
…          …      …         …             …         …             …
1542   5118268      M         Y             N         1       360000.0
1544   5023655      F         N             N         0       225000.0
1545   5115992      M         Y             Y         2       180000.0
1546   5118219      M         Y             N         0       270000.0
1547   5053790      F         Y             Y         0       225000.0
```

```
                Type_Income                         EDUCATION  \
0                 Pensioner                  Higher education
1      Commercial associate                  Higher education
4      Commercial associate                  Higher education
6      Commercial associate                  Higher education
7                 Pensioner  Secondary / secondary special
…                        …                                 …
1542          State servant  Secondary / secondary special
1544   Commercial associate                Incomplete higher
1545                Working                  Higher education
1546                Working  Secondary / secondary special
1547                Working                  Higher education

             Marital_status       Housing_type  …  Employed_days  \
0                   Married  House / apartment  …        -1979.5
1                   Married  House / apartment  …         -586.0
4                   Married  House / apartment  …         -586.0
6                   Married  House / apartment  …         -586.0
7                   Married  House / apartment  …        -1979.5
…                        …                  …  …              …
1542                Married  House / apartment  …        -3536.0
1544   Single / not married  House / apartment  …        -1209.0
1545                Married  House / apartment  …        -2477.0
1546         Civil marriage  House / apartment  …         -645.0
1547                Married  House / apartment  …        -2859.0

      Mobile_phone  Work_Phone  Phone  EMAIL_ID  Type_Occupation  \
0                1           0      0         0         Laborers
1                1           1      1         0         Laborers
4                1           1      1         0         Laborers
6                1           1      1         0         Laborers
7                1           0      0         0         Laborers
…              …           …      …         …                …
1542             1           0      1         0          Drivers
1544             1           0      0         0      Accountants
1545             1           0      0         0         Managers
1546             1           1      1         0          Drivers
1547             1           0      0         0         Laborers

     Family_Members  label  Education_order  Annual_income_log
0                 2      1              3.0          12.100712
1                 2      1              3.0          12.660328
4                 2      1              3.0          12.660328
6                 2      1              3.0          12.660328
7                 2      1              1.0          12.100712
…               …      …                …                  …
```

```
1542                3      0              1.0          12.793859
1544                1      0              2.0          12.323856
1545                4      0              3.0          12.100712
1546                2      0              1.0          12.506177
1547                2      0              3.0          12.323856

[1382 rows x 21 columns]
```

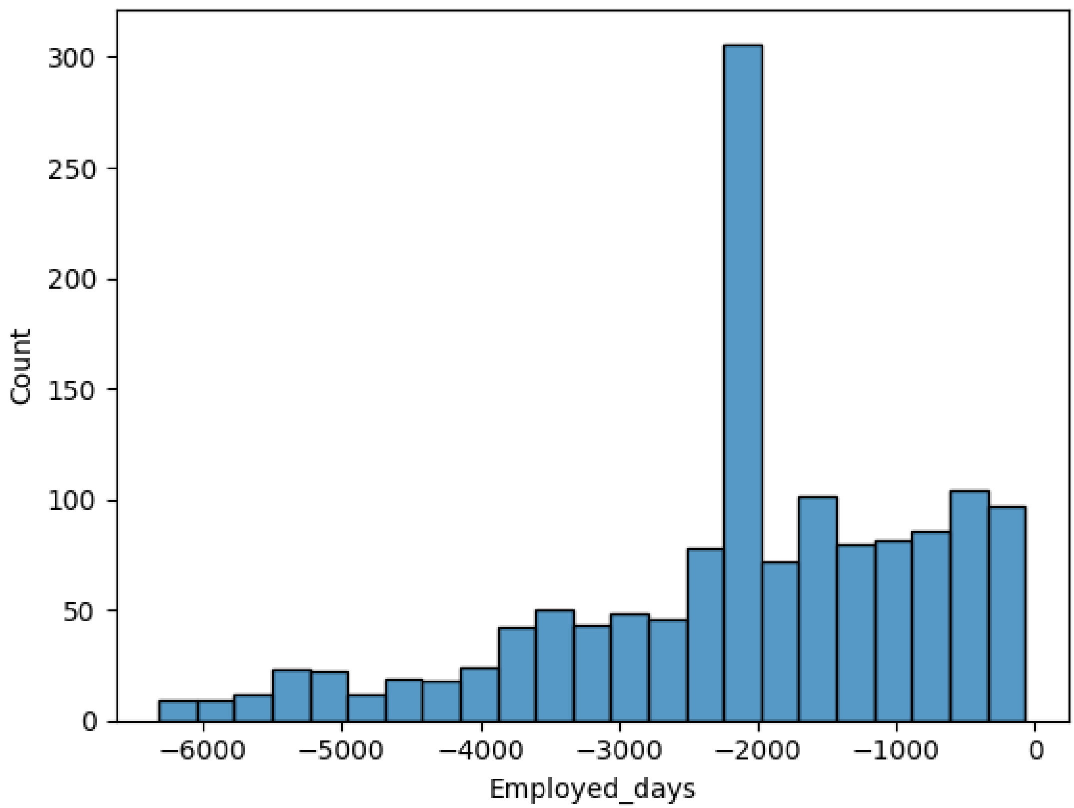
[160]:
```python
merge_df=merge_df.drop('Annual_income',axis=1)
```

[161]:
```python
merge_df.head()
```

[161]:
```
    Ind_ID GENDER Car_Owner Propert_Owner  CHILDREN           Type_Income  \
0  5008827      M         Y             Y         0             Pensioner
1  5009744      F         Y             N         0  Commercial associate
4  5009752      F         Y             N         0  Commercial associate
6  5009754      F         Y             N         0  Commercial associate
7  5009894      F         N             N         0             Pensioner

                       EDUCATION Marital_status       Housing_type  \
0               Higher education        Married  House / apartment
1               Higher education        Married  House / apartment
4               Higher education        Married  House / apartment
6               Higher education        Married  House / apartment
7  Secondary / secondary special        Married  House / apartment

   Birthday_count  Employed_days  Mobile_phone  Work_Phone  Phone  EMAIL_ID  \
0        -18772.0        -1979.5             1           0      0         0
1        -13557.0         -586.0             1           1      1         0
4        -13557.0         -586.0             1           1      1         0
6        -13557.0         -586.0             1           1      1         0
7        -22134.0        -1979.5             1           0      0         0

  Type_Occupation  Family_Members  label  Education_order  Annual_income_log
0        Laborers               2      1              3.0          12.100712
1        Laborers               2      1              3.0          12.660328
4        Laborers               2      1              3.0          12.660328
6        Laborers               2      1              3.0          12.660328
7        Laborers               2      1              1.0          12.100712
```

[162]:
```python
sns.histplot(merge_df['Employed_days'])#____ after applying histogram we can clearly observe that datais  left  skew
```

[162]:
```
<Axes: xlabel='Employed_days', ylabel='Count'>
```



[163]:
```
merge_df['Employed_days_ex'] = np.exp(merge_df['Employed_days'])##___ applying␣
↪exponatial function on the data to remove left skewness
```

[235]:
```
merge_df['Employed_days_ex'].skew()
```

[235]: 0

[165]:
```
merge_df['Annual_income_log'].skew()
```

[165]: 0.20242304500133093

SCALING

[166]:
```
#Scaling - converting high magnitude data to low magnitude

#from sklearn.preprocessing import StandardScaler
#sc = StandardScaler()
#merge_df['Annual_income_lM'] = sc.fit_transform(merge_df[['Annual_income']])



#print(merge_df['Annual_income_lM'])
```

[167]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
merge_df['Employed_days'] = sc.fit_transform(merge_df[['Employed_days']])
print(merge_df[['Employed_days']])
```

```
      Employed_days
0          0.070766
1          1.099796
4          1.099796
6          1.099796
7          0.070766
…               …
1542      -1.078631
1544       0.639742
1545      -0.296613
1546       1.056227
1547      -0.578700

[1382 rows x 1 columns]
```

[168]:
```
merge_df.head()
```

[168]:
```
     Ind_ID GENDER Car_Owner Propert_Owner  CHILDREN           Type_Income  \
0   5008827      M         Y             Y         0             Pensioner
1   5009744      F         Y             N         0  Commercial associate
4   5009752      F         Y             N         0  Commercial associate
6   5009754      F         Y             N         0  Commercial associate
7   5009894      F         N             N         0             Pensioner

                       EDUCATION Marital_status       Housing_type  \
0               Higher education        Married  House / apartment
1               Higher education        Married  House / apartment
4               Higher education        Married  House / apartment
6               Higher education        Married  House / apartment
7  Secondary / secondary special        Married  House / apartment

   Birthday_count  …  Mobile_phone  Work_Phone  Phone  EMAIL_ID  \
0        -18772.0  …             1           0      0         0
1        -13557.0  …             1           1      1         0
4        -13557.0  …             1           1      1         0
6        -13557.0  …             1           1      1         0
7        -22134.0  …             1           0      0         0

   Type_Occupation Family_Members  label  Education_order  Annual_income_log  \
0         Laborers              2      1              3.0          12.100712
1         Laborers              2      1              3.0          12.660328
4         Laborers              2      1              3.0          12.660328
```

```
6          Laborers                2        1              3.0          12.660328
7          Laborers                2        1              1.0          12.100712

   Employed_days_ex
0      0.000000e+00
1     3.187378e-255
4     3.187378e-255
6     3.187378e-255
7      0.000000e+00

[5 rows x 21 columns]
```

CHI SQUARED TEST

[169]:
```
#since p value is <0.05 , here no columns need to be removed.

from scipy.stats import chi2_contingency
contingency_table = pd.crosstab(merge_df['Annual_income_log'],merge_df['label'])
chi2,p_value,dof,expected = chi2_contingency(contingency_table)
print(chi2)
print(p_value)
```

```
152.4295214808943
0.004633929097314911
```

[170]:
```
num_col=data_final[['Employed_days','Birthday_count','Family_Members',]]
```

[171]:
```
num_col.head()
```

[171]:

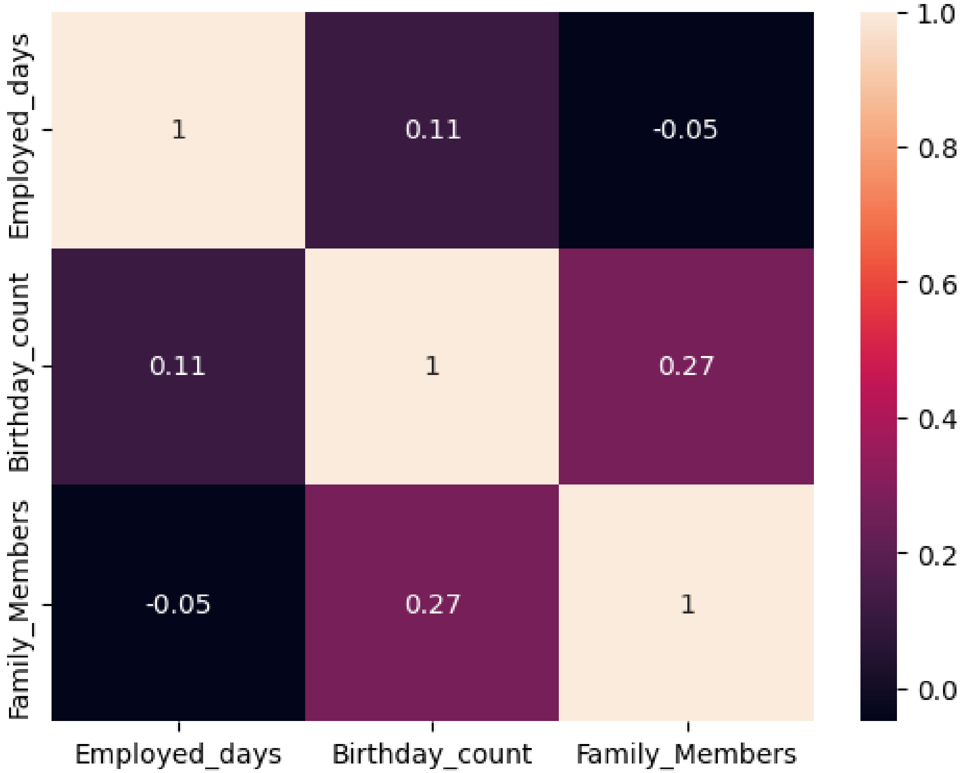
	Employed_days	Birthday_count	Family_Members
0	-1979.5	-18772.0	2
1	-586.0	-13557.0	2
4	-586.0	-13557.0	2
6	-586.0	-13557.0	2
7	-1979.5	-22134.0	2

[172]:
```
num_col.corr()  ## to see the correlation between the numerical columns
```

[172]:

	Employed_days	Birthday_count	Family_Members
Employed_days	1.000000	0.113545	-0.05010
Birthday_count	0.113545	1.000000	0.26789
Family_Members	-0.050100	0.267890	1.00000

[173]:
```
corr= num_col.corr()
sns.heatmap(corr,annot=True)
plt.show()
```

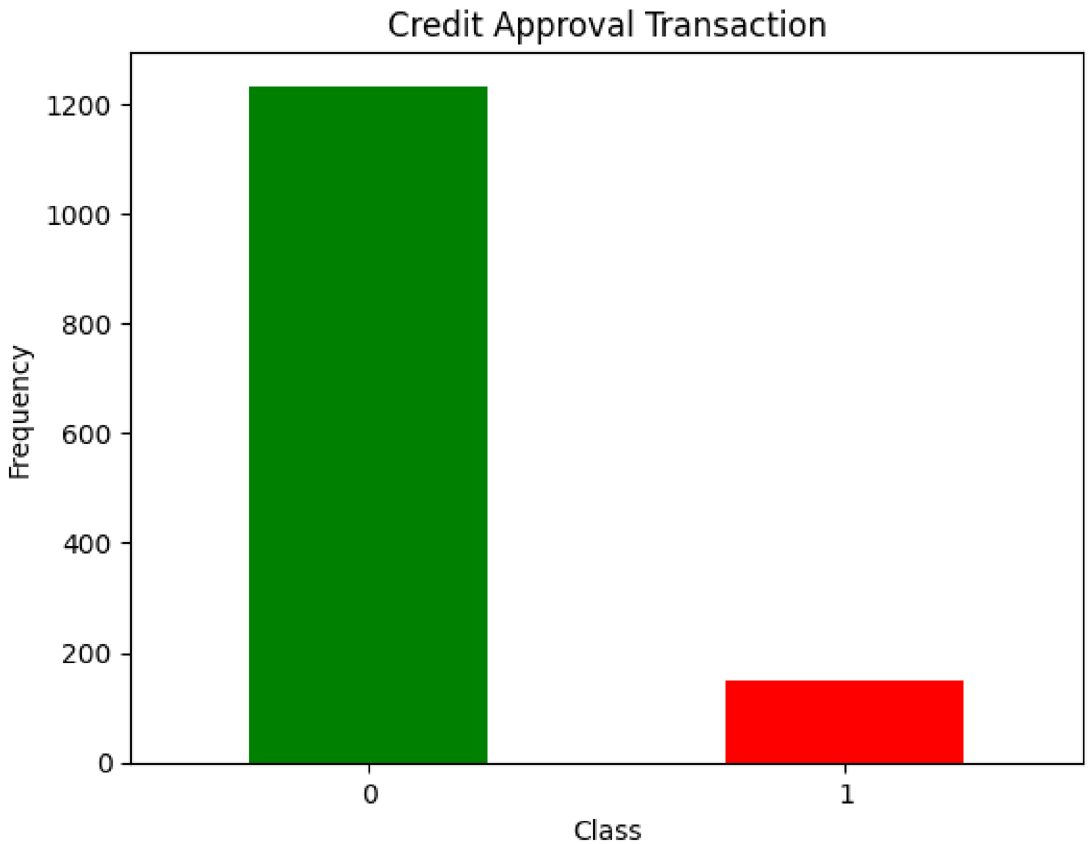


[174]:
```
data_final['label'].value_counts()
```

[174]:
```
0    1232
1     150
Name: label, dtype: int64
```

[175]:
```
classes=pd.value_counts(data_final['label'],sort=True)
classes.plot(kind='bar',color =['green','red'],rot=0)
plt.title('Credit Approval Transaction')
plt.xlabel('Class')
plt.ylabel('Frequency')
```

[175]:
```
Text(0, 0.5, 'Frequency')
```



We can clearly see that it is a perfect example of imbalanced data. The datset contains information about bank customers. The datset presents two part , out of 150 customers are not creditworthy out of 1382 customers and 1232 are good customers. There is a ratio of approx 9:1

[176]:
```
data_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1382 entries, 0 to 1547
Data columns (total 44 columns):
 #   Column                                       Non-Null Count  Dtype
---  ------                                       --------------  -----
 0   Ind_ID                                       1382 non-null   int64
 1   CHILDREN                                     1382 non-null   int64
 2   Annual_income                                1382 non-null   float64
 3   Birthday_count                               1382 non-null   float64
 4   Employed_days                                1382 non-null   float64
 5   Mobile_phone                                 1382 non-null   int64
 6   Work_Phone                                   1382 non-null   int64
 7   Phone                                        1382 non-null   int64
```

```
 8   EMAIL_ID                              1382 non-null   int64
 9   Family_Members                        1382 non-null   int64
 10  label                                 1382 non-null   int64
 11  Education_order                       1382 non-null   float64
 12  Pensioner                             1382 non-null   uint8
 13  State servant                         1382 non-null   uint8
 14  Working                               1382 non-null   uint8
 15  GENDER_M                              1382 non-null   uint8
 16  Car_Owner_Y                           1382 non-null   uint8
 17  Propert_Owner_Y                       1382 non-null   uint8
 18  Marital_status_Married                1382 non-null   uint8
 19  Marital_status_Separated              1382 non-null   uint8
 20  Marital_status_Single / not married   1382 non-null   uint8
 21  Marital_status_Widow                  1382 non-null   uint8
 22  Housing_type_House / apartment        1382 non-null   uint8
 23  Housing_type_Municipal apartment      1382 non-null   uint8
 24  Housing_type_Office apartment         1382 non-null   uint8
 25  Housing_type_Rented apartment         1382 non-null   uint8
 26  Housing_type_With parents             1382 non-null   uint8
 27  Type_Occupation_Cleaning staff        1382 non-null   uint8
 28  Type_Occupation_Cooking staff         1382 non-null   uint8
 29  Type_Occupation_Core staff            1382 non-null   uint8
 30  Type_Occupation_Drivers               1382 non-null   uint8
 31  Type_Occupation_HR staff              1382 non-null   uint8
 32  Type_Occupation_High skill tech staff 1382 non-null   uint8
 33  Type_Occupation_IT staff              1382 non-null   uint8
 34  Type_Occupation_Laborers              1382 non-null   uint8
 35  Type_Occupation_Low-skill Laborers    1382 non-null   uint8
 36  Type_Occupation_Managers              1382 non-null   uint8
 37  Type_Occupation_Medicine staff        1382 non-null   uint8
 38  Type_Occupation_Private service staff 1382 non-null   uint8
 39  Type_Occupation_Realty agents         1382 non-null   uint8
 40  Type_Occupation_Sales staff           1382 non-null   uint8
 41  Type_Occupation_Secretaries           1382 non-null   uint8
 42  Type_Occupation_Security staff        1382 non-null   uint8
 43  Type_Occupation_Waiters/barmen staff  1382 non-null   uint8
dtypes: float64(4), int64(8), uint8(32)
memory usage: 183.5 KB
```

[177]:
```python
from imblearn.combine import SMOTETomek ## importing necessary libraries
```

[178]:
```python
## Create dependent and independent features
```

[179]:
```python
columns = data_final.columns.tolist()
```

[180]:
```python
columns = [c for c in columns if c not in ['label']]
```

[181]:
```
X = data_final[columns]
```

[182]:
```
X
```

[182]:
```
      Ind_ID  CHILDREN  Annual_income  Birthday_count  Employed_days  \
0     5008827         0       180000.0        -18772.0        -1979.5
1     5009744         0       315000.0        -13557.0         -586.0
4     5009752         0       315000.0        -13557.0         -586.0
6     5009754         0       315000.0        -13557.0         -586.0
7     5009894         0       180000.0        -22134.0        -1979.5
…         …       …            …              …             …
1542  5118268         1       360000.0        -11294.0        -3536.0
1544  5023655         0       225000.0        -10229.0        -1209.0
1545  5115992         2       180000.0        -13174.0        -2477.0
1546  5118219         0       270000.0        -15292.0         -645.0
1547  5053790         0       225000.0        -16601.0        -2859.0

      Mobile_phone  Work_Phone  Phone  EMAIL_ID  Family_Members  …  \
0                1           0      0         0               2  …
1                1           1      1         0               2  …
4                1           1      1         0               2  …
6                1           1      1         0               2  …
7                1           0      0         0               2  …
…              …          …    …       …                …  …
1542             1           0      1         0               3  …
1544             1           0      0         0               1  …
1545             1           0      0         0               4  …
1546             1           1      1         0               2  …
1547             1           0      0         0               2  …

      Type_Occupation_Laborers  Type_Occupation_Low-skill Laborers  \
0                            1                                   0
1                            1                                   0
4                            1                                   0
6                            1                                   0
7                            1                                   0
…                          …                                 …
1542                         0                                   0
1544                         0                                   0
1545                         0                                   0
1546                         0                                   0
1547                         1                                   0

      Type_Occupation_Managers  Type_Occupation_Medicine staff  \
0                            0                               0
1                            0                               0
4                            0                               0
```

```
6                            0                               0
7                            0                               0
…                        …                               …
1542                         0                               0
1544                         0                               0
1545                         1                               0
1546                         0                               0
1547                         0                               0

      Type_Occupation_Private service staff  Type_Occupation_Realty agents  \
0                                         0                              0
1                                         0                              0
4                                         0                              0
6                                         0                              0
7                                         0                              0
…                                     …                              …
1542                                      0                              0
1544                                      0                              0
1545                                      0                              0
1546                                      0                              0
1547                                      0                              0

      Type_Occupation_Sales staff  Type_Occupation_Secretaries  \
0                               0                            0
1                               0                            0
4                               0                            0
6                               0                            0
7                               0                            0
…                           …                            …
1542                            0                            0
1544                            0                            0
1545                            0                            0
1546                            0                            0
1547                            0                            0

      Type_Occupation_Security staff  Type_Occupation_Waiters/barmen staff
0                                  0                                     0
1                                  0                                     0
4                                  0                                     0
6                                  0                                     0
7                                  0                                     0
…                              …                                     …
1542                               0                                     0
1544                               0                                     0
1545                               0                                     0
1546                               0                                     0
1547                               0                                     0
```

```
[1382 rows x 43 columns]
```

[183]:
```
Y = data_final['label'] ## Output data
```

[184]:
```
## Implementing oversamling for imbalanced dataset
```

[185]:
```
smk = SMOTETomek(random_state=42)
```

[186]:
```
X_res,Y_res=smk.fit_resample(X,Y)
```

[187]:
```
print(X_res.shape,Y_res.shape)
```

```
(2362, 43) (2362,)
```

[188]:
```
X.head()
```

[188]:
```
    Ind_ID  CHILDREN  Annual_income  Birthday_count  Employed_days  \
0  5008827         0       180000.0        -18772.0        -1979.5
1  5009744         0       315000.0        -13557.0         -586.0
4  5009752         0       315000.0        -13557.0         -586.0
6  5009754         0       315000.0        -13557.0         -586.0
7  5009894         0       180000.0        -22134.0        -1979.5

   Mobile_phone  Work_Phone  Phone  EMAIL_ID  Family_Members  …  \
0             1           0      0         0               2  …
1             1           1      1         0               2  …
4             1           1      1         0               2  …
6             1           1      1         0               2  …
7             1           0      0         0               2  …

   Type_Occupation_Laborers  Type_Occupation_Low-skill Laborers  \
0                         1                                   0
1                         1                                   0
4                         1                                   0
6                         1                                   0
7                         1                                   0

   Type_Occupation_Managers  Type_Occupation_Medicine staff  \
0                         0                               0
1                         0                               0
4                         0                               0
6                         0                               0
7                         0                               0

   Type_Occupation_Private service staff  Type_Occupation_Realty agents  \
0                                      0                              0
```

```
1                                        0                                0
4                                        0                                0
6                                        0                                0
7                                        0                                0

   Type_Occupation_Sales staff  Type_Occupation_Secretaries  \
0                            0                            0
1                            0                            0
4                            0                            0
6                            0                            0
7                            0                            0

   Type_Occupation_Security staff  Type_Occupation_Waiters/barmen staff
0                               0                                     0
1                               0                                     0
4                               0                                     0
6                               0                                     0
7                               0                                     0

[5 rows x 43 columns]
```

to compare the output variable value counts

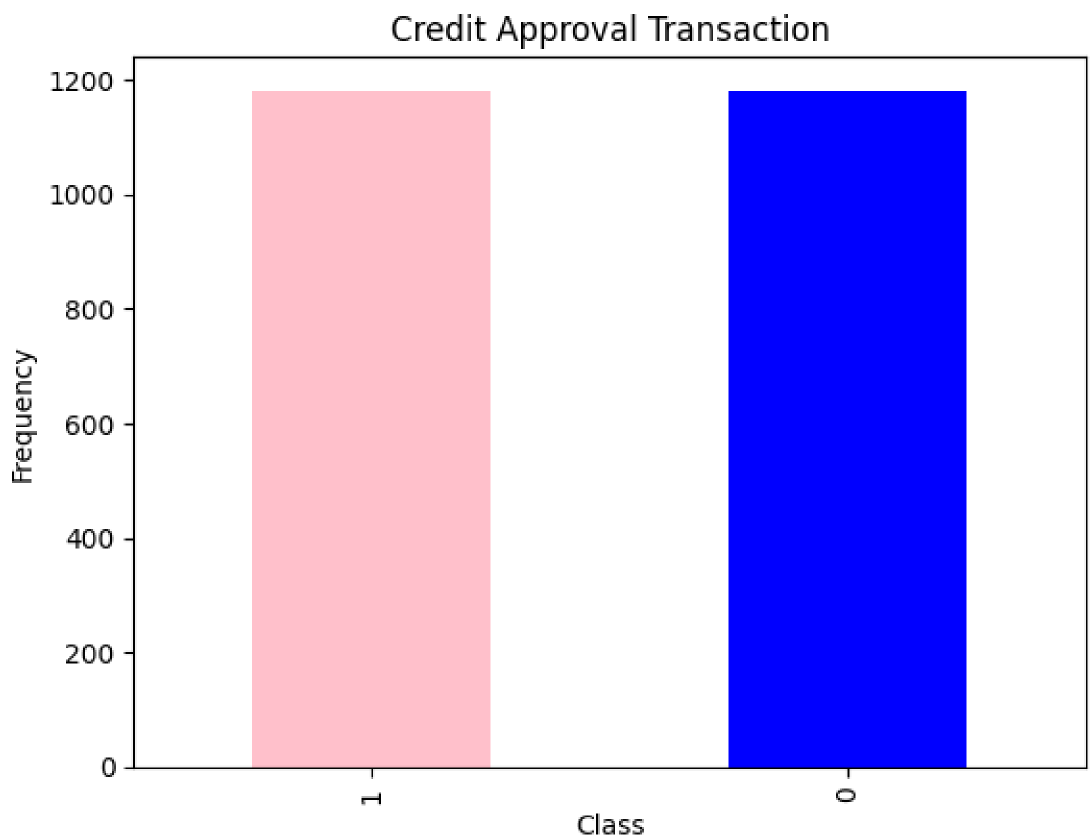
[189]:
```
from collections import Counter
```

[190]:
```
print('Original dataset shape{}'.format(Counter(Y)))
print('Resampled dataset shape{}'.format(Counter(Y_res)))
```

```
Original dataset shapeCounter({0: 1232, 1: 150})
Resampled dataset shapeCounter({1: 1181, 0: 1181})
```

[191]:
```
Y_res.value_counts().plot(kind="bar", color=[ 'pink','Blue'])
plt.title('Credit Approval Transaction')
plt.xlabel('Class')
plt.ylabel('Frequency')
```

[191]: Text(0, 0.5, 'Frequency')



Now the data has became balanced

[192]:
```
df_final = pd.concat([X_res,Y_res],axis=1) ## to conactenate the input & output␣
↪data
```

[193]:
```
df_final.head()
```

[193]:

	Ind_ID	CHILDREN	Annual_income	Birthday_count	Employed_days	\
0	5008827	0	180000.0	-18772.0	-1979.5	
1	5009744	0	315000.0	-13557.0	-586.0	
2	5009752	0	315000.0	-13557.0	-586.0	
3	5009754	0	315000.0	-13557.0	-586.0	
4	5009894	0	180000.0	-22134.0	-1979.5	

	Mobile_phone	Work_Phone	Phone	EMAIL_ID	Family_Members	…	\
0	1	0	0	0	2	…	
1	1	1	1	0	2	…	
2	1	1	1	0	2	…	
3	1	1	1	0	2	…	
4	1	0	0	0	2	…	

```
   Type_Occupation_Low-skill Laborers  Type_Occupation_Managers  \
0                                   0                         0
1                                   0                         0
2                                   0                         0
3                                   0                         0
4                                   0                         0

   Type_Occupation_Medicine staff  Type_Occupation_Private service staff  \
0                               0                                      0
1                               0                                      0
2                               0                                      0
3                               0                                      0
4                               0                                      0

   Type_Occupation_Realty agents  Type_Occupation_Sales staff  \
0                              0                            0
1                              0                            0
2                              0                            0
3                              0                            0
4                              0                            0

   Type_Occupation_Secretaries  Type_Occupation_Security staff  \
0                            0                               0
1                            0                               0
2                            0                               0
3                            0                               0
4                            0                               0

   Type_Occupation_Waiters/barmen staff  label
0                                     0      1
1                                     0      1
2                                     0      1
3                                     0      1
4                                     0      1

[5 rows x 44 columns]
```

[194]:
```python
from sklearn.model_selection import train_test_split
```

[197]:
```python
X_train, X_test, Y_train, Y_test = train_test_split(X_res, Y_res, test_size = 0.
 ↪15,  random_state = 5)
```

[198]:
```python
print(X_res.shape, X_train.shape ,Y_train.shape, X_test.shape) ## shape of test␣
 ↪and train data
```

```
(2362, 43) (2007, 43) (2007,) (355, 43)
```

[195]:
```
#### Standardisation of the data
```

[199]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

[200]:
```
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

[195]:
```
## APPly model to dataset
```

1.2 APPly model to dataset (Logistic Regression)

[201]:
```
from sklearn.linear_model import LogisticRegression
```

[202]:
```
l_reg = LogisticRegression()
l_reg.fit(X_train, Y_train)
```

[202]: LogisticRegression()

[203]:
```
y_prediction = l_reg.predict(X_test)
```

[206]:
```
from sklearn.metrics import accuracy_score
```

[211]:
```
accuracy_score(Y_test, y_prediction)
```

[211]: 0.8422535211267606

Apply Model to dataset(Decision TRee)

[212]:
```
from sklearn.tree import DecisionTreeClassifier
```

[213]:
```
DT = DecisionTreeClassifier()
DT.fit(X_train,Y_train)
```

[213]: DecisionTreeClassifier()

[215]:
```
y_prediction1 = DT.predict(X_test)
```

[216]:
```
accuracy_score(Y_test,y_prediction1) #####
```

[216]: 0.9042253521126761

Apply Model to Data SET (K- Nearest Neighbours)

[220]:
```
from sklearn.neighbors import KNeighborsClassifier
```

[223]:
```
model = KNeighborsClassifier()
model.fit(X_train,Y_train)
```

[223]:
```
KNeighborsClassifier()
```

[227]:
```
y_pred = model.predict(X_test)
```

[229]:
```
accuracy_score(Y_test,y_pred)
```

[229]:
```
0.9154929577464789
```

2 My model has given the higher accuracy through KNN model out of all three models so i am implenting this for my project because bank or finacial organisation cannot lose thier credit worthy customer and loan to thosecustomers who won't able to payback the loan amount.

Apply Confusion Matrix

[226]:
```
from sklearn.metrics import confusion_matrix
```

[230]:
```
confusion_matrix(Y_test,y_pred)
```

[230]:
```
array([[156,  15],
       [ 15, 169]])
```