Import modules

In [243]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

loading the dataset

In [244]:
```
data=pd.read_csv('C:/Users/harshitha/Documents/cd/creditcards.csv')
```

```
C:\Users\harshitha\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3063: DtypeWarning: Columns (16) have
mixed types.Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [245]:
```
data.head()
```
Out[245]:

OfOpenCreditLinesAndLoans	NumberOfTimes90DaysLate	NumberRealEstateLoansOrLines	NumberOfTime60-89DaysPastDueNotWorse	NumberOfDependents	Good_Bad
13.0	0.0	6.0	0.0	2	Bad
4.0	0.0	0.0	0.0	1	Good
2.0	1.0	0.0	0.0	0	Good
5.0	0.0	0.0	0.0	0	Good
7.0	0.0	1.0	0.0	0	Good

In [246]:
```
data.tail()
```
Out[246]:

	NPA Status	RevolvingUtilizationOfUnsecuredLines	age	Gender	Region	MonthlyIncome	Rented_OwnHouse	Occupation	Education	59Days
149997	0.0	0.246044	58.0	Male	North	NaN	Rented	Officer2	Professional	
149998	0.0	0.000000	30.0	Male	North	5716.0	Rented	Non-officer	Professional	
149999	0.0	0.850283	64.0	Male	North	8158.0	Ownhouse	Self_Emp	Professional	
150000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
150001	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [247]:
```
data.describe()
```

Out[247]:

	NPA Status	RevolvingUtilizationOfUnsecuredLines	age	MonthlyIncome	NumberOfTime30-59DaysPastDueNotWorse	DebtRatio	MonthlyIncon
count	150000.000000	150000.000000	150000.000000	1.202690e+05	150000.000000	150000.000000	1.202690e
mean	0.066840	6.048438	52.295207	6.670221e+03	0.421033	353.005076	6.670221e
std	0.249746	249.755371	14.771866	1.438467e+04	4.192781	2037.818523	1.438467e
min	0.000000	0.000000	0.000000	0.000000e+00	0.000000	0.000000	0.000000e
25%	0.000000	0.029867	41.000000	3.400000e+03	0.000000	0.175074	3.400000e
50%	0.000000	0.154181	52.000000	5.400000e+03	0.000000	0.366508	5.400000e
75%	0.000000	0.559046	63.000000	8.249000e+03	0.000000	0.868254	8.249000e
max	1.000000	50708.000000	109.000000	3.008750e+06	98.000000	329664.000000	3.008750e

```
In [248]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150002 entries, 0 to 150001
Data columns (total 18 columns):
 #   Column                                Non-Null Count   Dtype  
---  ------                                --------------   -----  
 0   NPA Status                            150000 non-null  float64
 1   RevolvingUtilizationOfUnsecuredLines  150000 non-null  float64
 2   age                                   150000 non-null  float64
 3   Gender                                150000 non-null  object 
 4   Region                                150000 non-null  object 
 5   MonthlyIncome                         120269 non-null  float64
 6   Rented_OwnHouse                       150000 non-null  object 
 7   Occupation                            150000 non-null  object 
 8   Education                             150000 non-null  object 
 9   NumberOfTime30-59DaysPastDueNotWorse  150000 non-null  float64
 10  DebtRatio                             150000 non-null  float64
 11  MonthlyIncome.1                       120269 non-null  float64
 12  NumberOfOpenCreditLinesAndLoans       150000 non-null  float64
 13  NumberOfTimes90DaysLate               150000 non-null  float64
 14  NumberRealEstateLoansOrLines          150000 non-null  float64
 15  NumberOfTime60-89DaysPastDueNotWorse  150000 non-null  float64
 16  NumberOfDependents                    146078 non-null  object 
 17  Good_Bad                              150000 non-null  object 
dtypes: float64(11), object(7)
memory usage: 20.6+ MB
```

In [249]:
```
data.apply(lambda x:len(x.unique()))
```

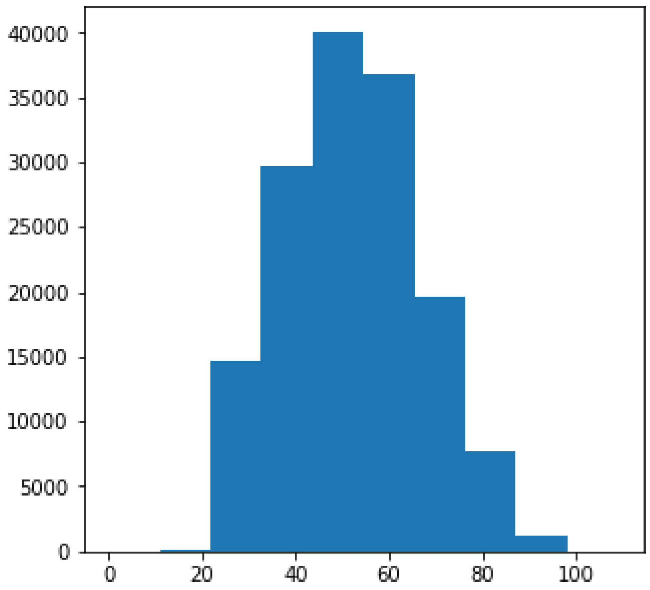
Out[249]:
```
NPA Status                                   3
RevolvingUtilizationOfUnsecuredLines    125732
age                                         87
Gender                                       3
Region                                       6
MonthlyIncome                            13595
Rented_OwnHouse                              3
Occupation                                   6
Education                                    6
NumberOfTime30-59DaysPastDueNotWorse        17
DebtRatio                               114194
MonthlyIncome.1                          13595
NumberOfOpenCreditLinesAndLoans             59
NumberOfTimes90DaysLate                     20
NumberRealEstateLoansOrLines                29
NumberOfTime60-89DaysPastDueNotWorse        14
NumberOfDependents                          26
Good_Bad                                     3
dtype: int64
```

Exploratory Data Analysis

In [250]:

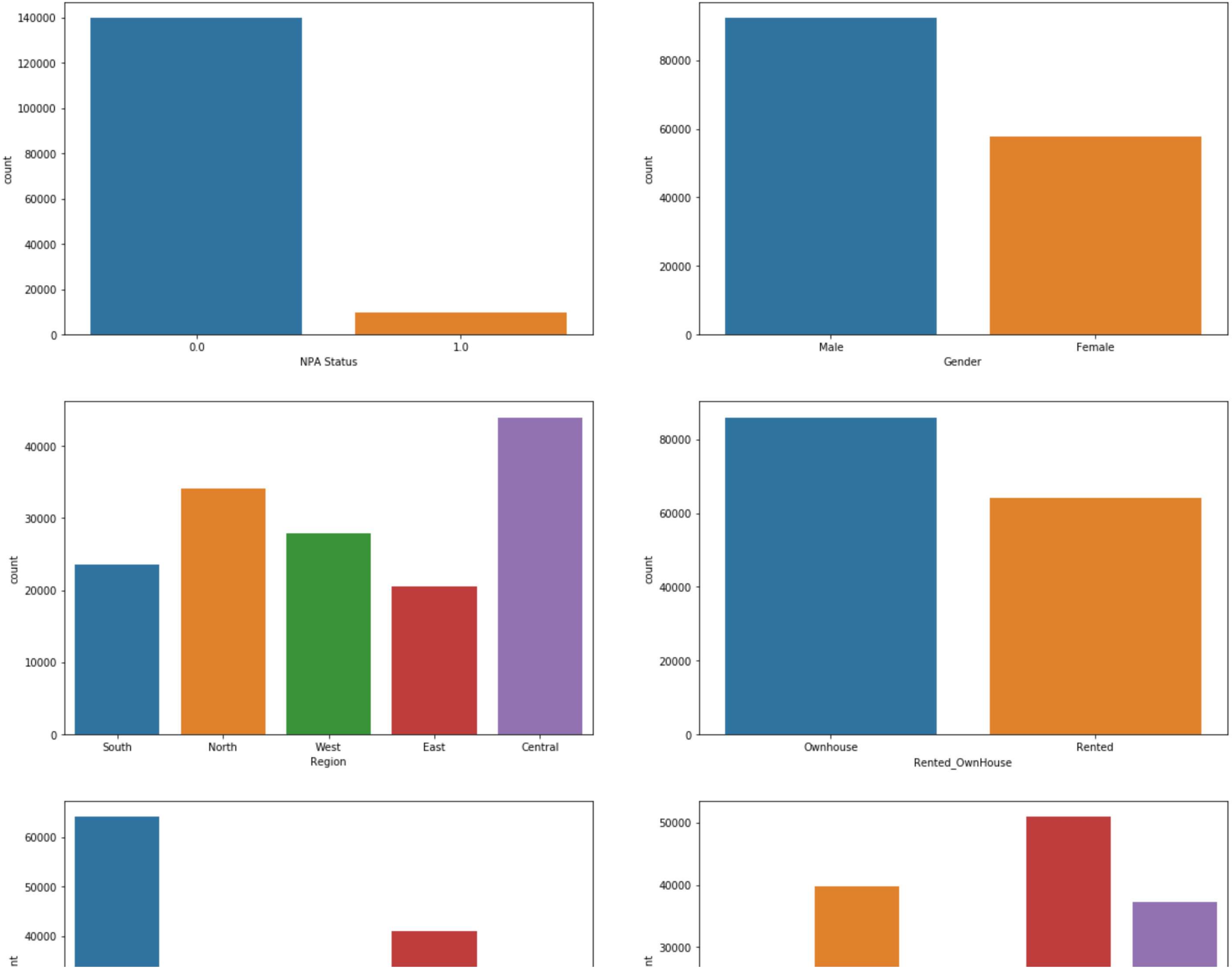
```
fig1 , ax1 = plt.subplots(figsize = (5,5))
plt.hist( 'age' , data = data)
plt.show()
```

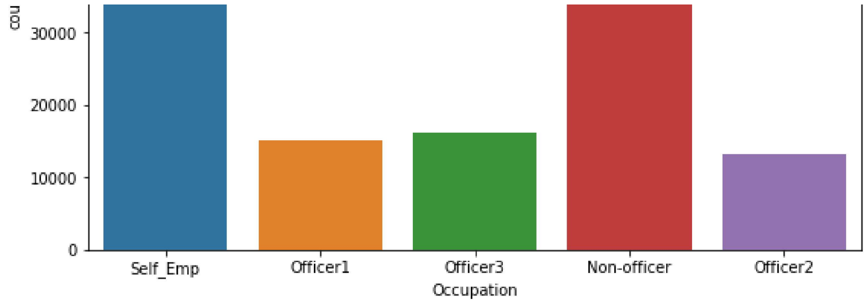
```
C:\Users\harshitha\anaconda3\lib\site-packages\numpy\lib\histograms.py:839: RuntimeWarning: invalid value encountered
in greater_equal
  keep = (tmp_a >= first_edge)
C:\Users\harshitha\anaconda3\lib\site-packages\numpy\lib\histograms.py:840: RuntimeWarning: invalid value encountered
in less_equal
  keep &= (tmp_a <= last_edge)
```

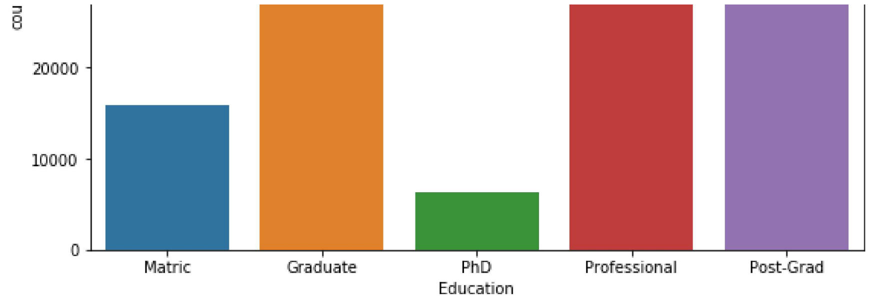


In [251]:

```
column=['NPA Status','Gender','Region','Rented_OwnHouse','Occupation','Education']
fig, ax = plt.subplots(3, 2, figsize=(20,20))
for var, subplot in zip(column, ax.flatten()):
    sns.countplot(data[var], ax=subplot)
```


count
NPA Status
0.0
1.0
0
20000
40000
60000
80000
100000
120000
140000
count
Gender
Male
Female
0
20000
40000
60000
80000
count
Region
South
North
West
East
Central
0
10000
20000
30000
40000
count
Rented_OwnHouse
Ownhouse
Rented
0
20000
40000
60000
80000
40000
50000
60000
30000
40000
50000





preprocessing the dataset

In [252]:
```
data.isnull().sum()
```

Out[252]:
```
NPA Status                                  2
RevolvingUtilizationOfUnsecuredLines        2
age                                         2
Gender                                      2
Region                                      2
MonthlyIncome                           29733
Rented_OwnHouse                             2
Occupation                                  2
Education                                   2
NumberOfTime30-59DaysPastDueNotWorse        2
DebtRatio                                   2
MonthlyIncome.1                         29733
NumberOfOpenCreditLinesAndLoans             2
NumberOfTimes90DaysLate                     2
NumberRealEstateLoansOrLines                2
NumberOfTime60-89DaysPastDueNotWorse        2
NumberOfDependents                       3924
Good_Bad                                    2
dtype: int64
```

In [253]:
```
data.tail()
```

Out[253]:

	NPA Status	RevolvingUtilizationOfUnsecuredLines	age	Gender	Region	MonthlyIncome	Rented_OwnHouse	Occupation	Education	59Days
149997	0.0	0.246044	58.0	Male	North	NaN	Rented	Officer2	Professional	
149998	0.0	0.000000	30.0	Male	North	5716.0	Rented	Non-officer	Professional	
149999	0.0	0.850283	64.0	Male	North	8158.0	Ownhouse	Self_Emp	Professional	
150000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
150001	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [254]:
```
data=data.drop([150000,150001])
```

In [255]:
```
data.isnull().sum()
```

Out[255]:
```
NPA Status                                  0
RevolvingUtilizationOfUnsecuredLines        0
age                                         0
Gender                                      0
Region                                      0
MonthlyIncome                           29731
Rented_OwnHouse                             0
Occupation                                  0
Education                                   0
NumberOfTime30-59DaysPastDueNotWorse        0
DebtRatio                                   0
MonthlyIncome.1                         29731
NumberOfOpenCreditLinesAndLoans             0
NumberOfTimes90DaysLate                     0
NumberRealEstateLoansOrLines                0
NumberOfTime60-89DaysPastDueNotWorse        0
NumberOfDependents                       3924
Good_Bad                                    0
dtype: int64
```

In [256]:
```
Comp=pd.DataFrame({"MonthlyIncome":data['MonthlyIncome'],"MonthlyIncome.1":data['MonthlyIncome.1']})
```

In [257]:
```
Comp.head(20)
```

Out[257]:

	MonthlyIncome	MonthlyIncome.1
0	9120.0	9120.0
1	2600.0	2600.0
2	3042.0	3042.0
3	3300.0	3300.0
4	63588.0	63588.0
5	3500.0	3500.0
6	NaN	NaN
7	3500.0	3500.0
8	NaN	NaN
9	23684.0	23684.0
10	2500.0	2500.0
11	6501.0	6501.0
12	12454.0	12454.0
13	13700.0	13700.0
14	0.0	0.0
15	11362.0	11362.0
16	NaN	NaN
17	8800.0	8800.0
18	3280.0	3280.0
19	333.0	333.0

In [258]:
```
Comp.tail()
```

Out[258]:

	MonthlyIncome	MonthlyIncome.1
149995	2100.0	2100.0
149996	5584.0	5584.0
149997	NaN	NaN
149998	5716.0	5716.0
149999	8158.0	8158.0

In [259]:
```
data=data.drop(["MonthlyIncome.1"],axis=1)
data.head()
```

Out[259]:

	NPA Status	RevolvingUtilizationOfUnsecuredLines	age	Gender	Region	MonthlyIncome	Rented_OwnHouse	Occupation	Education	Numb 59DaysPastD
0	1.0	0.766127	45.0	Male	South	9120.0	Ownhouse	Self_Emp	Matric	
1	0.0	0.957151	40.0	Female	South	2600.0	Ownhouse	Self_Emp	Graduate	
2	0.0	0.658180	38.0	Female	South	3042.0	Ownhouse	Self_Emp	PhD	
3	0.0	0.233810	30.0	Female	South	3300.0	Ownhouse	Self_Emp	Professional	
4	0.0	0.907239	49.0	Male	South	63588.0	Ownhouse	Self_Emp	Post-Grad	

In [260]:
```
(data['MonthlyIncome'].isnull().sum()/data.shape[0])*100
```

Out[260]: 19.820666666666668

In [261]:
```
data['NumberOfDependents']=pd.to_numeric(data['NumberOfDependents'])
```

In [262]:
```
(data['NumberOfDependents'].isnull().sum()/data.shape[0])*100
```

Out[262]: 2.616

In [263]:
```
data['MonthlyIncome']=data['MonthlyIncome'].fillna(data['MonthlyIncome'].median())
```

In [264]:
```
data['NumberOfDependents']=data['NumberOfDependents'].fillna(data['NumberOfDependents'].median())
```

In [265]:
```
data.isnull().sum()
```

Out[265]:
```
NPA Status                              0
RevolvingUtilizationOfUnsecuredLines    0
age                                     0
Gender                                  0
Region                                  0
MonthlyIncome                           0
Rented_OwnHouse                         0
Occupation                              0
Education                               0
NumberOfTime30-59DaysPastDueNotWorse    0
DebtRatio                               0
NumberOfOpenCreditLinesAndLoans         0
NumberOfTimes90DaysLate                 0
NumberRealEstateLoansOrLines            0
NumberOfTime60-89DaysPastDueNotWorse    0
NumberOfDependents                      0
Good_Bad                                0
dtype: int64
```

In [266]:
```
from scipy.stats import pearsonr
```

In [267]:
```
corr = data.corr()
corr
```

Out[267]:

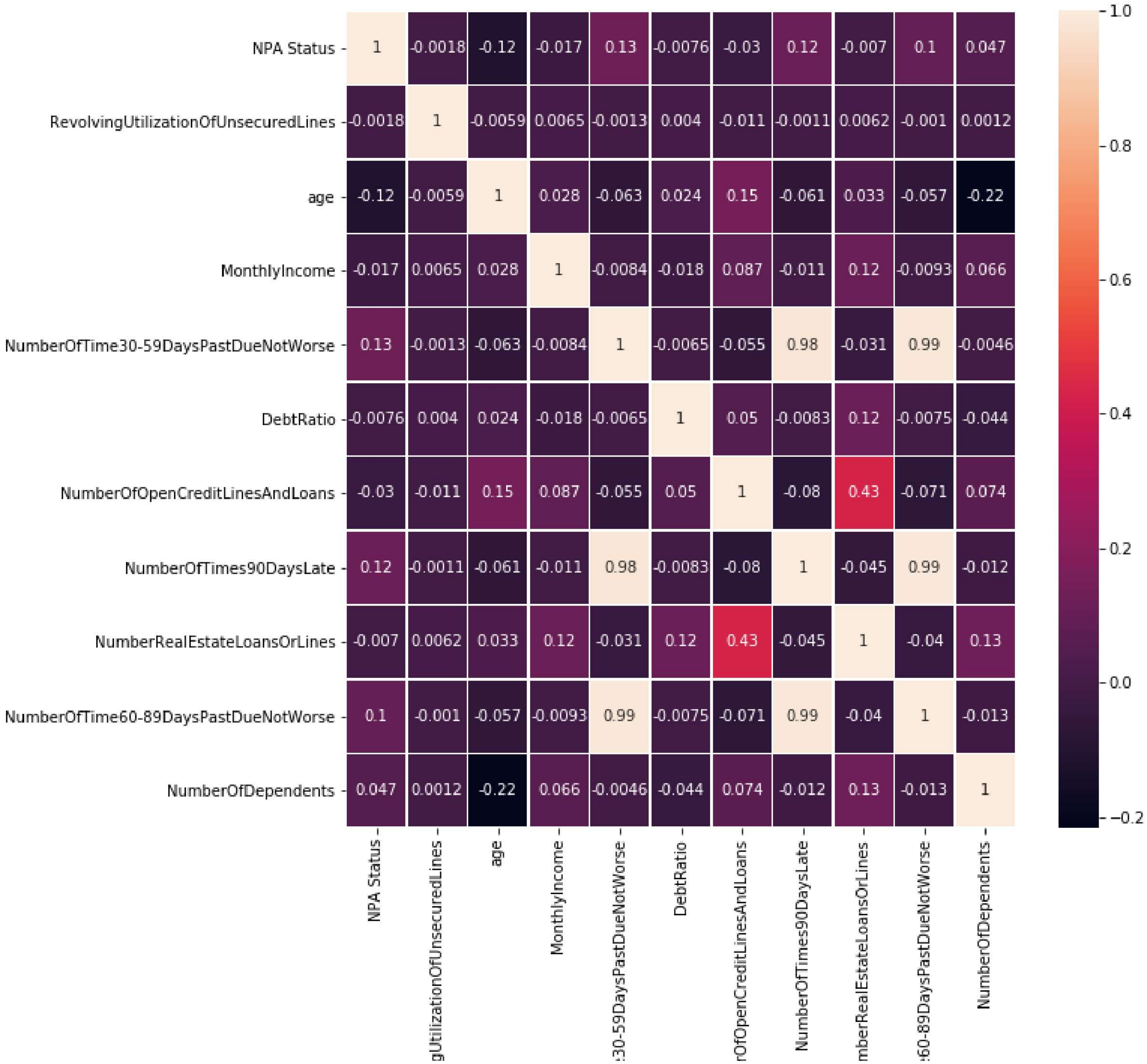
	NPA Status	RevolvingUtilizationOfUnsecuredLines	age	MonthlyIncome	NumberOfTime30-59DaysPastDueNotWorse	DebtRa
NPA Status	1.000000	-0.001802	-0.115386	-0.017151	0.125587	-0.0076
RevolvingUtilizationOfUnsecuredLines	-0.001802	1.000000	-0.005898	0.006513	-0.001314	0.0039
age	-0.115386	-0.005898	1.000000	0.027581	-0.062995	0.0241
MonthlyIncome	-0.017151	0.006513	0.027581	1.000000	-0.008370	-0.0180
NumberOfTime30-59DaysPastDueNotWorse	0.125587	-0.001314	-0.062995	-0.008370	1.000000	-0.0065
DebtRatio	-0.007602	0.003961	0.024188	-0.018006	-0.006542	1.0000
NumberOfOpenCreditLinesAndLoans	-0.029669	-0.011281	0.147705	0.086949	-0.055312	0.0495
NumberOfTimes90DaysLate	0.117175	-0.001061	-0.061005	-0.010500	0.983603	-0.0083
NumberRealEstateLoansOrLines	-0.007038	0.006235	0.033150	0.116273	-0.030565	0.1200
NumberOfTime60-89DaysPastDueNotWorse	0.102261	-0.001048	-0.057159	-0.009252	0.987005	-0.0075
NumberOfDependents	0.046869	0.001193	-0.215693	0.066314	-0.004590	-0.0444

In [268]:

```
fig,ax = plt.subplots(figsize=(10, 10))
sns.heatmap(corr,annot= True, linewidths=.5)
```

Out[268]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a75f2d8a08>
```



NPA Status
RevolvingUtilizationOfUnsecuredLines
age
MonthlyIncome
NumberOfTime30-59DaysPastDueNotWorse
DebtRatio
NumberOfOpenCreditLinesAndLoans
NumberOfTimes90DaysLate
NumberRealEstateLoansOrLines
NumberOfTime60-89DaysPastDueNotWorse
NumberOfDependents
1 -0.0018 -0.12 -0.017 0.13 -0.0076 -0.03 0.12 -0.007 0.1 0.047
-0.0018 1 -0.0059 0.0065 -0.0013 0.004 -0.011 -0.0011 0.0062 -0.001 0.0012
-0.12 -0.0059 1 0.028 -0.063 0.024 0.15 -0.061 0.033 -0.057 -0.22
-0.017 0.0065 0.028 1 -0.0084 -0.018 0.087 -0.011 0.12 -0.0093 0.066
0.13 -0.0013 -0.063 -0.0084 1 -0.0065 -0.055 0.98 -0.031 0.99 -0.0046
-0.0076 0.004 0.024 -0.018 -0.0065 1 0.05 -0.0083 0.12 -0.0075 -0.044
-0.03 -0.011 0.15 0.087 -0.055 0.05 1 -0.08 0.43 -0.071 0.074
0.12 -0.0011 -0.061 -0.011 0.98 -0.0083 -0.08 1 -0.045 0.99 -0.012
-0.007 0.0062 0.033 0.12 -0.031 0.12 0.43 -0.045 1 -0.04 0.13
0.1 -0.001 -0.057 -0.0093 0.99 -0.0075 -0.071 0.99 -0.04 1 -0.013
0.047 0.0012 -0.22 0.066 -0.0046 -0.044 0.074 -0.012 0.13 -0.013 1
NPA Status
gUtilizationOfUnsecuredLines
age
MonthlyIncome
30-59DaysPastDueNotWorse
DebtRatio
rOfOpenCreditLinesAndLoans
NumberOfTimes90DaysLate
mberRealEstateLoansOrLines
60-89DaysPastDueNotWorse
NumberOfDependents
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Revolvin NumberOfTime Numbe Nu NumberOfTime

In [269]:
```
cat_data=data.select_dtypes(include='object')
cat_data.head()
```

Out[269]:

	Gender	Region	Rented_OwnHouse	Occupation	Education	Good_Bad
0	Male	South	Ownhouse	Self_Emp	Matric	Bad
1	Female	South	Ownhouse	Self_Emp	Graduate	Good
2	Female	South	Ownhouse	Self_Emp	PhD	Good
3	Female	South	Ownhouse	Self_Emp	Professional	Good
4	Male	South	Ownhouse	Self_Emp	Post-Grad	Good

In [270]:
```
from sklearn.preprocessing import LabelEncoder
```

In [271]:
```
lb=LabelEncoder()
cat_data=cat_data.apply(lb.fit_transform)
```

In [272]:
```
cat_data.head()
```

Out[272]:

	Gender	Region	Rented_OwnHouse	Occupation	Education	Good_Bad
0	1	3	0	4	1	0
1	0	3	0	4	0	1
2	0	3	0	4	2	1
3	0	3	0	4	4	1
4	1	3	0	4	3	1

In [273]:
```python
num_data=data.select_dtypes(exclude='object')
num_data.head()
```

Out[273]:

	NPA Status	RevolvingUtilizationOfUnsecuredLines	age	MonthlyIncome	NumberOfTime30-59DaysPastDueNotWorse	DebtRatio	NumberOfOpenCreditLinesAndLoans	Nu
0	1.0	0.766127	45.0	9120.0	2.0	0.802982	13.0	
1	0.0	0.957151	40.0	2600.0	0.0	0.121876	4.0	
2	0.0	0.658180	38.0	3042.0	1.0	0.085113	2.0	
3	0.0	0.233810	30.0	3300.0	0.0	0.036050	5.0	
4	0.0	0.907239	49.0	63588.0	1.0	0.024926	7.0	

In [274]:
```python
cat_data1=cat_data.drop(['Good_Bad'],axis=1)
cat_data1.head()
```

Out[274]:

	Gender	Region	Rented_OwnHouse	Occupation	Education
0	1	3	0	4	1
1	0	3	0	4	0
2	0	3	0	4	2
3	0	3	0	4	4
4	1	3	0	4	3

In [275]:
```python
from sklearn.feature_selection import chi2
```

In [276]:
```python
chi_scores = chi2(cat_data,y)
chi_scores
```

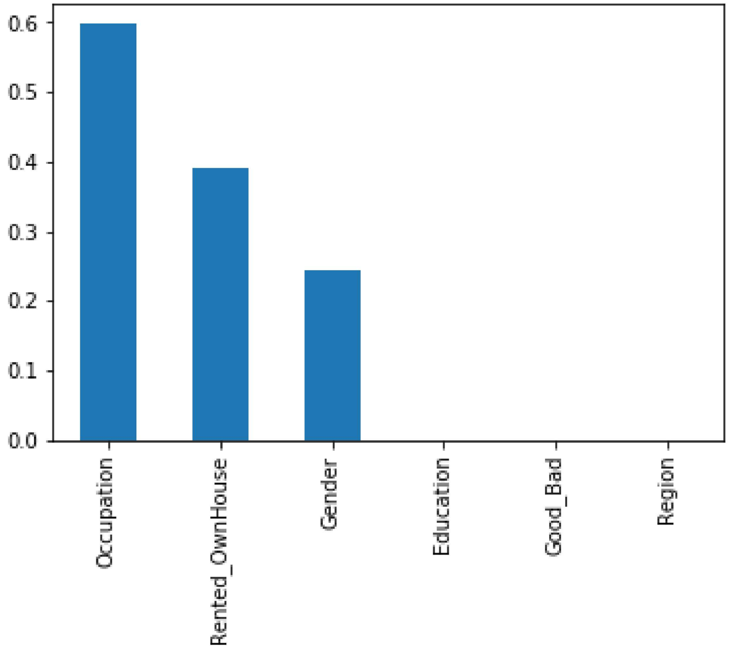
Out[276]:
```
(array([1.35736842e+00, 8.57256291e+03, 7.32621097e-01, 2.79330648e-01,
        2.18863964e+02, 1.00260000e+04]),
 array([2.43994078e-01, 0.00000000e+00, 3.92034906e-01, 5.97140270e-01,
        1.60023918e-49, 0.00000000e+00]))
```

In [277]:
```
p_values = pd.Series(chi_scores[1],index = cat_data.columns)
p_values.sort_values(ascending = False , inplace = True)
p_values
```

Out[277]:
```
Occupation          5.971403e-01
Rented_OwnHouse     3.920349e-01
Gender              2.439941e-01
Education           1.600239e-49
Good_Bad            0.000000e+00
Region              0.000000e+00
dtype: float64
```

In [278]:
```
p_values.plot.bar()
```

Out[278]: <matplotlib.axes._subplots.AxesSubplot at 0x1a75f2d7408>



model with Imbalance Output variable

In [279]:
```
x=pd.concat([cat_data1,num_data],1)
x.head()
```

Out[279]:

	Gender	Region	Rented_OwnHouse	Occupation	Education	NPA Status	RevolvingUtilizationOfUnsecuredLines	age	MonthlyIncome	Numbe 59DaysPastDu
0	1	3	0	4	1	1.0	0.766127	45.0	9120.0	
1	0	3	0	4	0	0.0	0.957151	40.0	2600.0	
2	0	3	0	4	2	0.0	0.658180	38.0	3042.0	
3	0	3	0	4	4	0.0	0.233810	30.0	3300.0	
4	1	3	0	4	3	0.0	0.907239	49.0	63588.0	

In [280]:
```
y=cat_data['Good_Bad']
y.head()
```

Out[280]:
```
0    0
1    1
2    1
3    1
4    1
Name: Good_Bad, dtype: int32
```

In [281]:
```
from sklearn.model_selection import train_test_split
```

In [282]:
```
from sklearn.preprocessing import StandardScaler
Standard=StandardScaler()
x=Standard.fit_transform(x)
x
```

Out[282]:
```
array([[ 0.79061053,  0.8104916 , -0.8632148 , ...,  4.40954554,
        -0.05785249,  1.14052977],
       [-1.26484529,  0.8104916 , -0.8632148 , ..., -0.90128301,
        -0.05785249,  0.23720186],
       [-1.26484529,  0.8104916 , -0.8632148 , ..., -0.90128301,
        -0.05785249, -0.66612604],
       ...,
       [ 0.79061053,  0.13195312,  1.15846022, ..., -0.01614492,
        -0.05785249, -0.66612604],
       [ 0.79061053,  0.13195312,  1.15846022, ..., -0.90128301,
        -0.05785249, -0.66612604],
       [ 0.79061053,  0.13195312, -0.8632148 , ...,  0.86899317,
        -0.05785249, -0.66612604]])
```

In [283]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=1)
```

In [284]:
```
from sklearn.linear_model import LogisticRegression
```

In [285]:
```
Model=LogisticRegression()
Model.fit(x_train,y_train)
```

Out[285]:
```
LogisticRegression()
```

In [286]:
```
Prediction=Model.predict(x_test)
Prediction
```

Out[286]:
```
array([1, 1, 1, ..., 1, 1, 1])
```

In [287]:
```
A=pd.DataFrame({"Actual":y_test,"Estimated":Prediction})
A
```

Out[287]:

	Actual	Estimated
58397	1	1
108538	1	1
149880	1	1
127668	1	1
66331	1	1
...	...	...
92822	1	1
18396	1	1
48997	1	1
18776	1	1
129232	1	1

45000 rows × 2 columns

In [288]:
```
Model.score(x_test,y_test)*100
```

Out[288]: 100.0

In [289]:
```
Model.score(x_train,y_train)*100
```

Out[289]: 100.0

In [290]:
```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
```

In [291]:
```
accuracy_score(y_test,Prediction)*100
```

Out[291]: 100.0

In [292]:
```
confusion_matrix(y_test,Prediction)
```

Out[292]:
```
array([[ 2988,     0],
       [    0, 42012]], dtype=int64)
```

In [293]:
```
print(classification_report(y_test,Prediction))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2988
           1       1.00      1.00      1.00     42012

    accuracy                           1.00     45000
   macro avg       1.00      1.00      1.00     45000
weighted avg       1.00      1.00      1.00     45000
```

Balancing the Output variables

In [294]:

```
print("Before'1': {}".format(sum(y_train == 1)))
print("Before '2': {} \n".format(sum(y_train == 2)))


from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 2)
x_train_res, y_train_res = sm.fit_sample(x_train, y_train)

print('After  train_X: {}'.format(x_train_res.shape))
print('After  train_y: {} \n'.format(y_train_res.shape))

print("After '1': {}".format(sum(y_train_res == 1)))
print("After '2': {}".format(sum(y_train_res == 2)))
```

```
Before'1': 97962
Before '2': 0

After  train_X: (195924, 16)
After  train_y: (195924,)

After '1': 97962
After '2': 0
```

In [295]:

```
Machine = LogisticRegression()
Machine.fit(x_train_res, y_train_res.ravel())
```

Out[295]:

```
LogisticRegression()
```

In [296]:

```
Prediction1=Machine.predict(x_test)
Prediction1
```

Out[296]:

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

In [297]:

```
Machine.score(x_test,y_test)*100
```

Out[297]:

```
100.0
```

In [298]:
```
Machine.score(x_train_res,y_train_res)*100
```
Out[298]: 100.0

In [299]:
```
A1=pd.DataFrame({"Actual":y_test,"Estimated":Prediction1})
A1
```
Out[299]:

	Actual	Estimated
58397	1	1
108538	1	1
149880	1	1
127668	1	1
66331	1	1
...	...	...
92822	1	1
18396	1	1
48997	1	1
18776	1	1
129232	1	1

45000 rows × 2 columns

In [300]:
```
Error=np.where(A1['Actual']!=A1['Estimated'])
Error[0].shape
```
Out[300]: (0,)

In [301]:
```
(Error[0].shape[0]/Prediction1.shape[0])*100
```
Out[301]: 0.0

In [ ]: