In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df=pd.read_csv(r'C:\Users\Elsaka\AMIT\Assignments\credit_score.csv')
pd.options.display.max_columns=None
df.head()
```

Out[2]:

	ID	Customer_ID	Month	Name	Age	SSN	Occupation	Annual_Income	Monthly_Inhand_Salary	Num_Bank_Accounts	Num_Credit_Card	Interest_Rate	Num_of_Loan	Type_of_Loan	Delay_from_due_date	Num_of_Del
0	0x1602	CUS_0xd40	January	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	1824.843333	3	4	3	4	Auto Loan, Credit-Builder Loan, Personal Loan,...	3	
1	0x1603	CUS_0xd40	February	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	NaN	3	4	3	4	Auto Loan, Credit-Builder Loan, Personal Loan,...	-1	
2	0x1604	CUS_0xd40	March	Aaron Maashoh	-500	821-00-0265	Scientist	19114.12	NaN	3	4	3	4	Auto Loan, Credit-Builder Loan, Personal Loan,...	3	
3	0x1605	CUS_0xd40	April	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	NaN	3	4	3	4	Auto Loan, Credit-Builder Loan, Personal Loan,...	5	
4	0x1606	CUS_0xd40	May	Aaron Maashoh	23	821-00-0265	Scientist	19114.12	1824.843333	3	4	3	4	Auto Loan, Credit-Builder Loan, Personal Loan,...	6	

check if we have duplicated rows

In [3]:
```
df.duplicated().sum()
```

Out[3]:
```
0
```

In [4]:
```
df.shape
```

Out[4]:
```
(100000, 28)
```

> - ID,customer_id,name,SNN---------> They won't affect on the credit score because it is a unique personal information so we can drop them

In [5]:
```
df.drop(['ID','Customer_ID','Name' ,'SSN'],axis=1 , inplace=True)
```

alternative to info()

In [7]:
```
def columns_info(df):
    colms = []
    dtypes = []
    unique = []
    nunique = []
    nulls = []

    for colm in df.columns:
        colms.append(colm)
        dtypes.append(df[colm].dtypes)
        unique.append(df[colm].unique())
        nunique.append(df[colm].nunique())
        nulls.append(df[colm].isna().sum())

    return pd.DataFrame({'Columns':colms ,
                         'Data Types': dtypes,
                         'Unique values':unique,
                         'Number of unique': nunique,
                         'missing values':nulls})
columns_info(df)
```

Out[7]:

	Columns	Data Types	Unique values	Number of unique	missing values
0	Month	object	[January, February, March, April, May, June, J...	8	0
1	Age	object	[23, -500, 28_, 28, 34, 54, 55, 21, 31, 33, 34...	1788	0
2	Occupation	object	[Scientist, _______, Teacher, Engineer, Entrep...	16	0
3	Annual_Income	object	[19114.12, 34847.84, 34847.84_, 143162.64, 306...	18940	0
4	Monthly_Inhand_Salary	float64	[1824.8433333333328, nan, 3037.986666666666, 1...	13235	15002
5	Num_Bank_Accounts	int64	[3, 2, 1, 7, 4, 0, 8, 5, 6, 9, 10, 1414, 1231,...	943	0
6	Num_Credit_Card	int64	[4, 1385, 5, 1288, 1, 7, 6, 1029, 488, 8, 1381...	1179	0
7	Interest_Rate	int64	[3, 6, 8, 4, 5, 5318, 15, 7, 12, 20, 1, 433, 1...	1750	0
8	Num_of_Loan	object	[4, 1, 3, 967, -100, 0, 0_, 2, 3_, 2_, 7, 5, 5...	434	0
9	Type_of_Loan	object	[Auto Loan, Credit-Builder Loan, Personal Loan...	6260	11408
10	Delay_from_due_date	int64	[3, -1, 5, 6, 8, 7, 13, 10, 0, 4, 9, 1, 12, 11...	73	0
11	Num_of_Delayed_Payment	object	[7, nan, 4, 8_, 6, 1, -1, 3_, 0, 8, 5, 3, 9, 1...	749	7002
12	Changed_Credit_Limit	object	[11.27, _, 6.27, 9.27, 5.42, 7.42, 6.42, 7.1, ...	4384	0
13	Num_Credit_Inquiries	float64	[4.0, 2.0, 3.0, nan, 5.0, 9.0, 8.0, 7.0, 6.0, ...	1223	1965
14	Credit_Mix	object	[_, Good, Standard, Bad]	4	0
15	Outstanding_Debt	object	[809.98, 605.03, 1303.01, 632.46, 943.86, 548....	13178	0
16	Credit_Utilization_Ratio	float64	[26.822619623699016, 31.94496005538421, 28.609...	100000	0
17	Credit_History_Age	object	[22 Years and 1 Months, nan, 22 Years and 3 Mo...	404	9030
18	Payment_of_Min_Amount	object	[No, NM, Yes]	3	0
19	Total_EMI_per_month	float64	[49.57494921489417, 18.816214573128885, 246.99...	14950	0
20	Amount_invested_monthly	object	[80.41529543900253, 118.28022162236736, 81.699...	91049	4479
21	Payment_Behaviour	object	[High_spent_Small_value_payments, Low_spent_La...	7	0
22	Monthly_Balance	object	[312.49408867943663, 284.62916249607184, 331.2...	98792	1200
23	Credit_Score	object	[Good, Standard, Poor]	3	0

Outlier Function with (I Q R _ stander deviation)

In [8]:
```
def check_outliers(colm,df):
    q1=df[colm].quantile(0.25)
    q3=df[colm].quantile(0.75)
    iqr=q3-q1
    lower_bound=q1-1.5*iqr
    upper_bound=q3+1.5*iqr
    outliers = []

    for i in range(len(df)):
        value = df.loc[i,colm]
        if value > upper_bound or value < lower_bound :
            outliers.append(value)
    return outliers
```

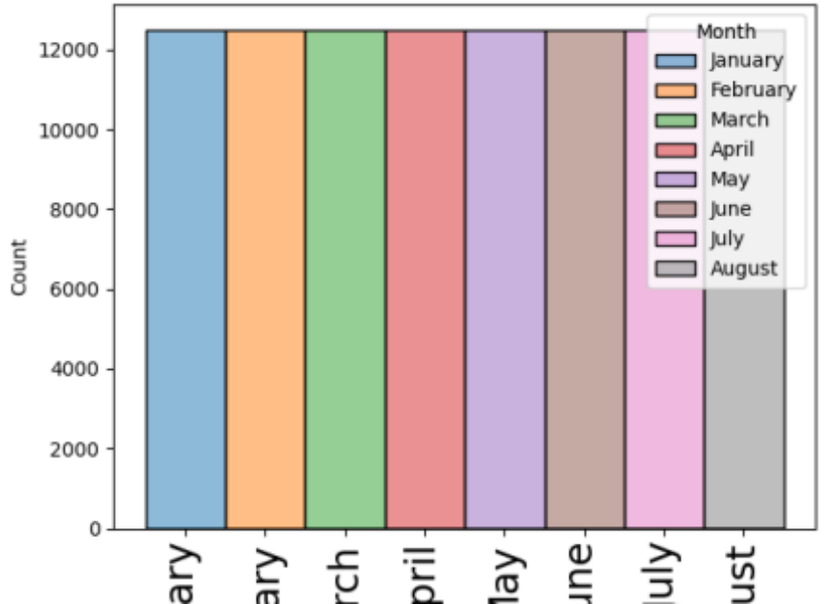
In [9]:
```
def handle_outliers(colm,df):
    q1=df[colm].quantile(0.25)
    q3=df[colm].quantile(0.75)
    iqr=q3-q1
    lower_bound= q1-1.5*iqr
    upper_bound= q3+1.5*iqr

    for i in range (len(df)):
        if df.loc[i,colm] < lower_bound :
            df.loc[i,colm] = lower_bound
        elif  df.loc[i,colm] > upper_bound:
             df.loc[i,colm] = upper_bound
```

1-Month

In [10]:
```
df['Month'].unique()
```

Out[10]:
```
array(['January', 'February', 'March', 'April', 'May', 'June', 'July',
       'August'], dtype=object)
```

In [11]:
```
plt.xticks(fontsize=20 ,rotation="vertical" )
sns.histplot(x='Month' , data=df, hue='Month' )
plt.show()
```

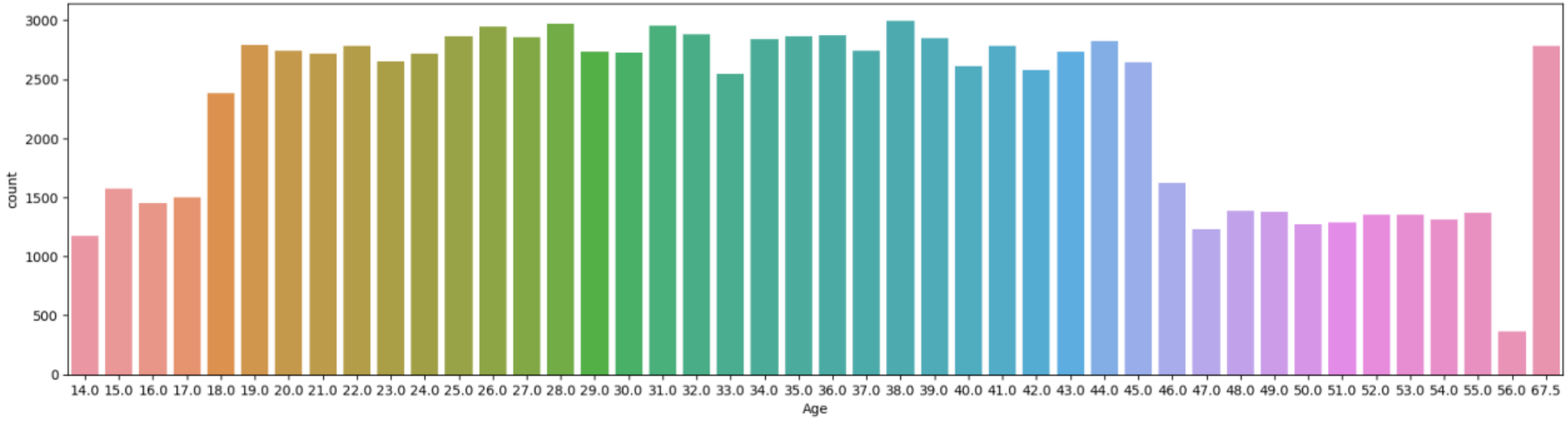


In [15]:
```
df['Age'].unique()
```

Out[15]:
```
array(['23', '-500', '28_', ..., '4808_', '2263', '1342'], dtype=object)
```

In [16]:
```
df['Age']=df['Age'].str.replace('-','')
df['Age']=df['Age'].str.replace('_','')
df['Age']=df['Age'].astype(int)
```

In [235…
```
plt.figure(figsize=(20,5))
sns.countplot(x='Age' , data= df)
plt.show()
```



In [18]:
```
df['Age'].describe()
```

Out[18]:
```
count    100000.000000
mean        119.509700
std         684.757313
min          14.000000
25%          25.000000
50%          34.000000
75%          42.000000
max        8698.000000
Name: Age, dtype: float64
```

In [19]:
```
check_outliers('Age',df)
```

Out[19]:
```
[500,
 7580,
 500,
 181,
 995,
 5079,
 6409,
 500,
 7080,
```

3- Occupation

In [23]:
```
df['Occupation'].unique()
```

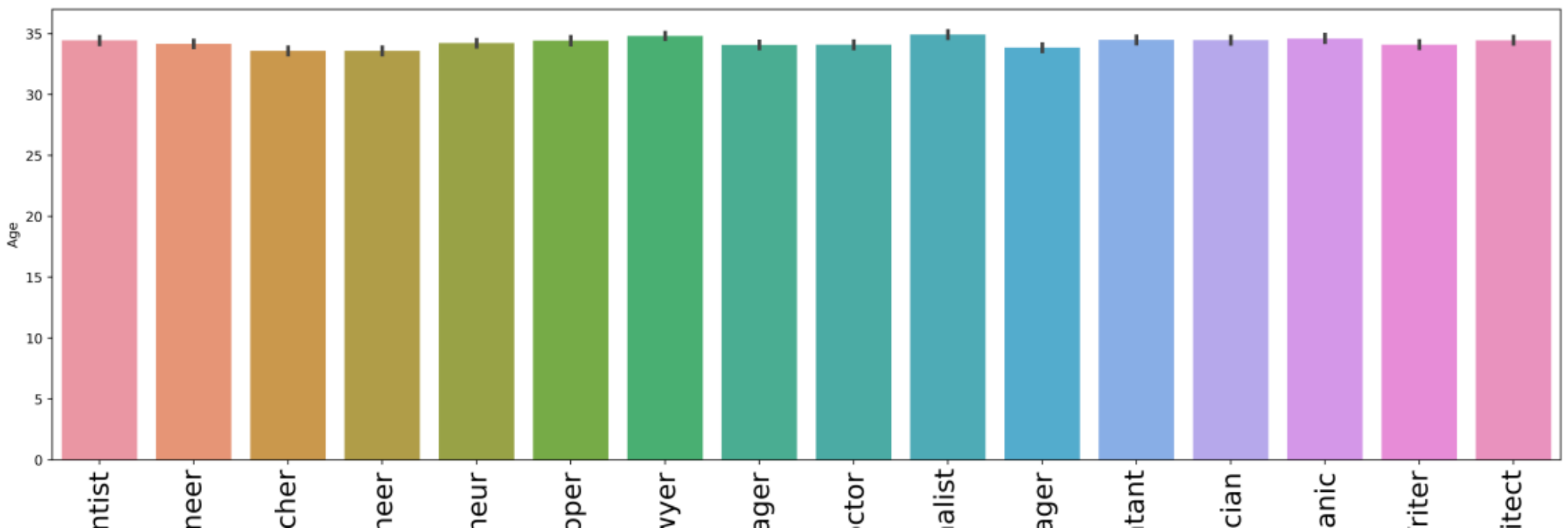
Out[23]:
```
array(['Scientist', '_______', 'Teacher', 'Engineer', 'Entrepreneur',
       'Developer', 'Lawyer', 'Media_Manager', 'Doctor', 'Journalist',
       'Manager', 'Accountant', 'Musician', 'Mechanic', 'Writer',
       'Architect'], dtype=object)
```

In [24]:
```
df['Occupation']=df['Occupation'].replace('_______','Cyber security engineer')
```

In [25]:
```
df['Occupation'].value_counts()
```

Out[25]:
```
Cyber security engineer    7062
Lawyer                     6575
Architect                  6355
Engineer                   6350
Scientist                  6299
Mechanic                   6291
Accountant                 6271
Developer                  6235
Media_Manager              6232
Teacher                    6215
Entrepreneur               6174
Doctor                     6087
Journalist                 6085
Manager                    5973
Musician                   5911
Writer                     5885
Name: Occupation, dtype: int64
```

In [26]:
```
plt.figure(figsize = (20,6), dpi = 400)
plt.xticks(fontsize=20, rotation="vertical")
sns.barplot(x='Occupation',y='Age',data =df)
plt.show()
```



10- Num_of_Loan

In [85]:
```
df['Num_of_Loan'].unique()
```

Out[85]:
```
array(['4', '1', '3', '967', '-100', '0', '0_', '2', '3_', '2_', '7', '5',
       '5_', '6', '8', '8_', '9', '9_', '4_', '7_', '1_', '1464', '6_',
       '622', '352', '472', '1017', '945', '146', '563', '341', '444',
       '720', '1485', '49', '737', '1106', '466', '728', '313', '843',
       '597_', '617', '119', '663', '640', '92_', '1019', '501', '1302',
       '39', '716', '848', '931', '1214', '186', '424', '1001', '1110',
       '1152', '457', '1433', '1187', '52', '1480', '1047', '1035',
       '1347_', '33', '193', '699', '329', '1451', '484', '132', '649',
       '995', '545', '684', '1135', '1094', '1204', '654', '58', '348',
       '614', '1363', '323', '1406', '1348', '430', '153', '1461', '905',
       '1312', '1424', '1154', '95', '1353', '1228', '819', '1006', '795',
       '359', '1209', '590', '696', '1185_', '1465', '911', '1181', '70',
       '816', '1369', '143', '1416', '455', '55', '1096', '1474', '420',
       '1131', '904', '89', '1259', '527', '1241', '449', '983', '418',
       '319', '23', '238', '638', '138', '235_', '280', '1070', '1484',
       '274', '494', '1459_', '404', '1354', '1495', '1391', '601',
       '1313', '1319', '898', '231', '752', '174', '961', '1046', '834',
       '284', '438', '288', '1463', '1151', '719', '198', '1015', '855',
       '841', '392', '1444', '103', '1320_', '745', '172', '252', '630_',
       '241', '31', '405', '1217', '1030', '1257', '137', '157', '164',
       '1088', '1236', '777', '1048', '613', '330', '1439', '321', '661',
       '952', '939', '562', '1202', '302', '943', '394', '955', '1318',
       '936', '781', '100', '1329', '1365', '860', '217', '191', '32',
       '282', '351', '1387', '757', '416', '833', '359_', '292', '1225_',
       '1227', '639', '859', '243', '267', '510', '332', '996', '597',
       '311', '492', '820', '336', '123', '540', '131_', '1311_', '1441',
       '895', '891', '50', '940', '935', '596', '29', '1182', '1129_',
       '1014', '251', '365', '291', '1447', '742', '1085', '148', '462',
       '832', '881', '1225', '1412', '785_', '1127', '910', '538', '999',
       '733', '101', '237', '87', '659', '633', '387', '447', '629',
       '831', '1384', '773', '621', '1419', '289', '143_', '285', '1393',
       '1131_', '27_', '1359', '1482', '1189', '1294', '201', '579',
       '814', '141', '1320', '581', '1171_', '295', '290', '433', '679',
       '1040', '1054', '1430', '1023', '1077', '1457', '1150', '701',
       '1382', '889', '437', '372', '1222', '126', '1159', '868', '19',
       '1297', '227_', '190', '809', '1216', '1074', '571', '520', '1274',
       '1340', '991', '316', '697', '926', '873', '1002', '378_', '65',
       '875', '867', '548', '652', '1372', '606', '1036', '1300', '17',
       '1178', '802', '1219_', '1271', '1137', '1496', '439', '196',
       '636', '192', '228', '1053', '229', '753', '1296', '1371', '254',
       '863', '464', '515', '838', '1160', '1289', '1298', '799', '182',
       '574', '527_', '242', '415', '869', '958', '54', '1265', '656',
       '275', '778', '208', '147', '350', '507', '463', '497', '1129',
       '927', '653', '662', '529', '635', '1027_', '897', '1039', '227',
       '1345', '924', '696_', '1279', '546', '1112', '1210', '526', '300',
       '1103', '504', '136', '1400', '78', '686', '1091', '344', '215',
       '84', '628', '1470', '968', '1478', '83', '1196', '1307', '1132_',
       '1008', '917', '657', '56', '18', '41', '801', '978', '216', '349',
       '966'], dtype=object)
```

In [86]:
```
df['Num_of_Loan']=df['Num_of_Loan'].str.replace('_','')
df['Num_of_Loan'] = df['Num_of_Loan'].str.replace('-','')
df['Num_of_Loan']=df['Num_of_Loan'].astype(int)
```

In [87]:
```
df['Num_of_Loan'].unique()
```

Out[87]:
```
array([   4,    1,    3,  967,  100,    0,    2,    7,    5,    6,    8,
          9, 1464,  622,  352,  472, 1017,  945,  146,  563,  341,  444,
        720, 1485,   49,  737, 1106,  466,  728,  313,  843,  597,  617,
```

```
'2047'], dtype=object)
```

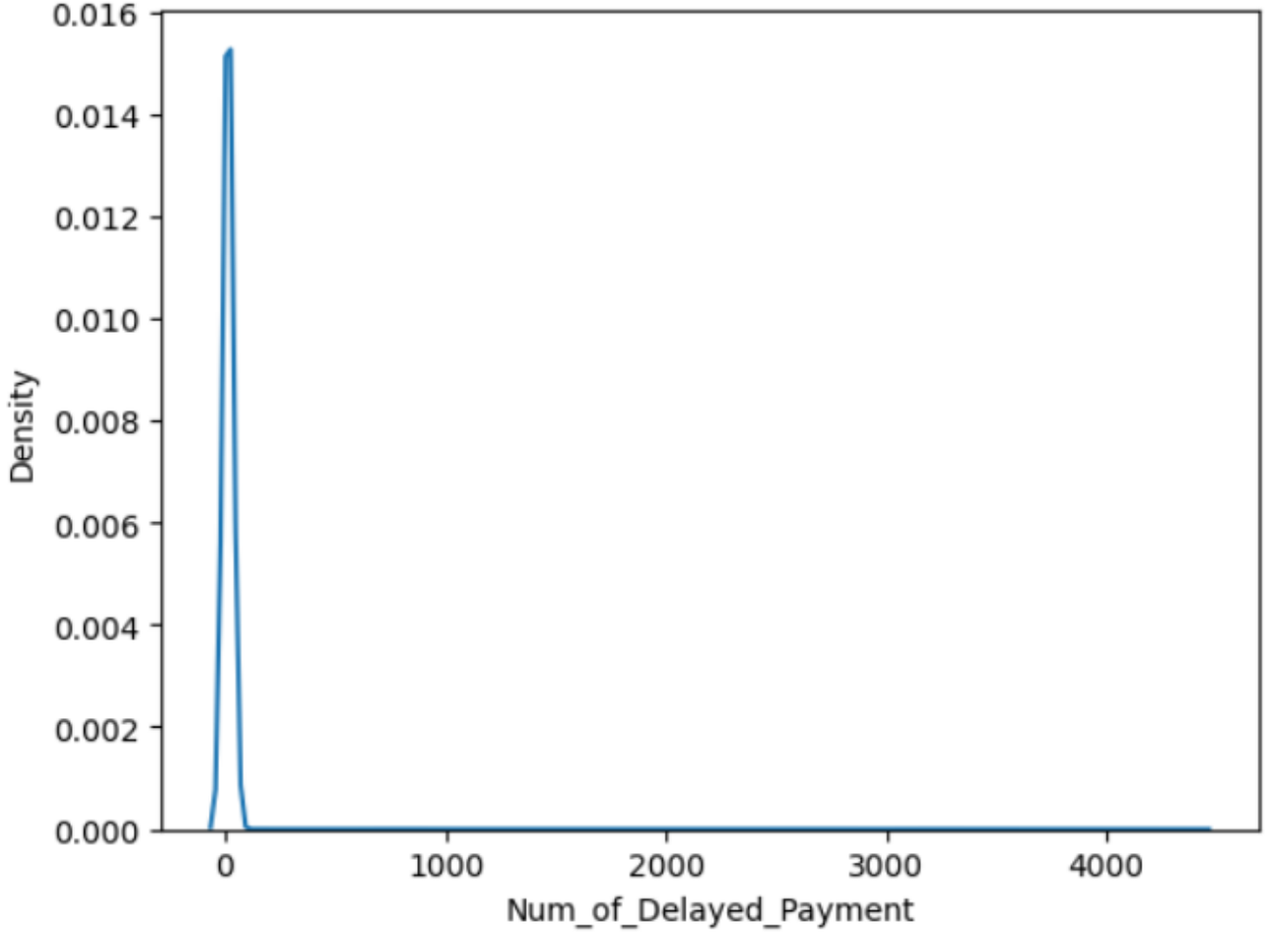
In [108…
```
df['Num_of_Delayed_Payment']=df['Num_of_Delayed_Payment'].str.replace('_','')
df['Num_of_Delayed_Payment']=df['Num_of_Delayed_Payment'].str.replace('-','')
df['Num_of_Delayed_Payment']=df['Num_of_Delayed_Payment'].astype(float)
```

In [109…
```
df['Num_of_Delayed_Payment'].isna().sum()
```

Out[109]:
```
7002
```

In [110…
```
mean=df['Num_of_Delayed_Payment'].mean()
df['Num_of_Delayed_Payment'].fillna(mean , inplace =True)
```

In [111…
```
sns.kdeplot(x= df['Num_of_Delayed_Payment'] , data =df)
plt.show()
```



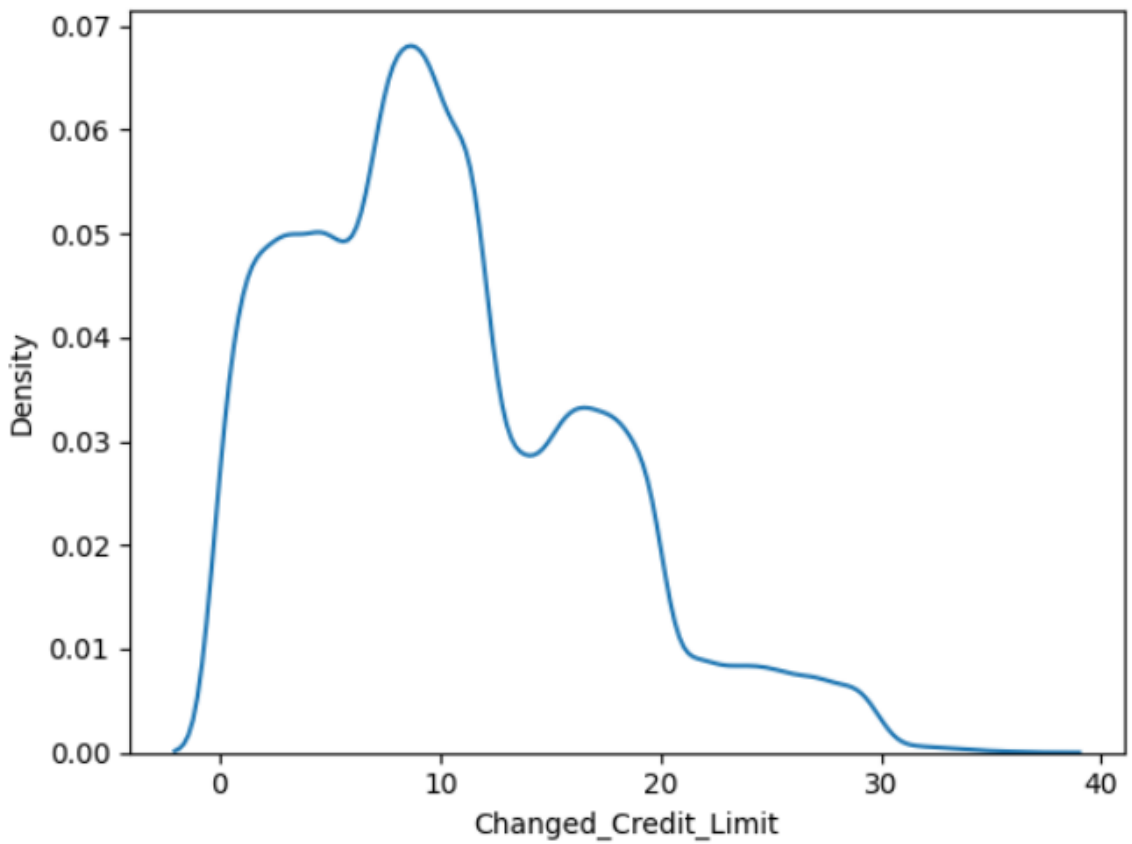
14- Changed_Credit_Limit

In [120…
```
df['Changed_Credit_Limit'].unique()
```

Out[120]:
```
array(['11.27', '_', '6.27', ..., '17.509999999999998', '25.16', '21.17'],
      dtype=object)
```

In [121…
```
df['Changed_Credit_Limit']=df['Changed_Credit_Limit'].str.replace('_','0')
df['Changed_Credit_Limit'] = df['Changed_Credit_Limit'].str.replace('-','')
df['Changed_Credit_Limit']=df['Changed_Credit_Limit'].astype(float)
df['Changed_Credit_Limit']= df['Changed_Credit_Limit'].replace('0',np.nan)
df['Changed_Credit_Limit']= df['Changed_Credit_Limit'].replace(np.nan , df['Changed_Credit_Limit'].mean())
```

In [122…
```
sns.kdeplot(x =df['Changed_Credit_Limit'] , data= df)
plt.show()
```



16- Credit_Mix

In [140…
```
df['Credit_Mix'].unique()
```

Out[140]:
```
array(['_', 'Good', 'Standard', 'Bad'], dtype=object)
```

In [141…
```
df['Credit_Mix'].value_counts()
```

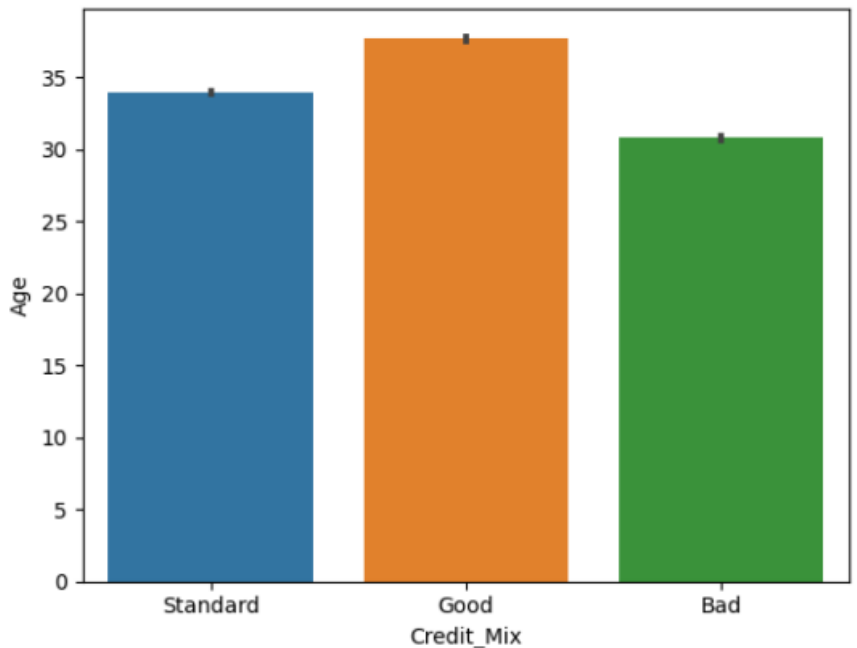
Out[141]:
```
Standard    36479
Good        24337
_           20195
Bad         18989
Name: Credit_Mix, dtype: int64
```

In [142…
```
df['Credit_Mix'] =df['Credit_Mix'].str.replace('_', 'Standard')
```

In [143…
```
df['Credit_Mix'].value_counts()
```

Out[143]:
```
Standard    56674
Good        24337
Bad         18989
Name: Credit_Mix, dtype: int64
```

In [144…
```
sns.barplot(x = df['Credit_Mix'] , y = df['Age'], data = df)
plt.show()
```



17-Outstanding_Debt

In [148…
```
df['Outstanding_Debt'].unique()
```

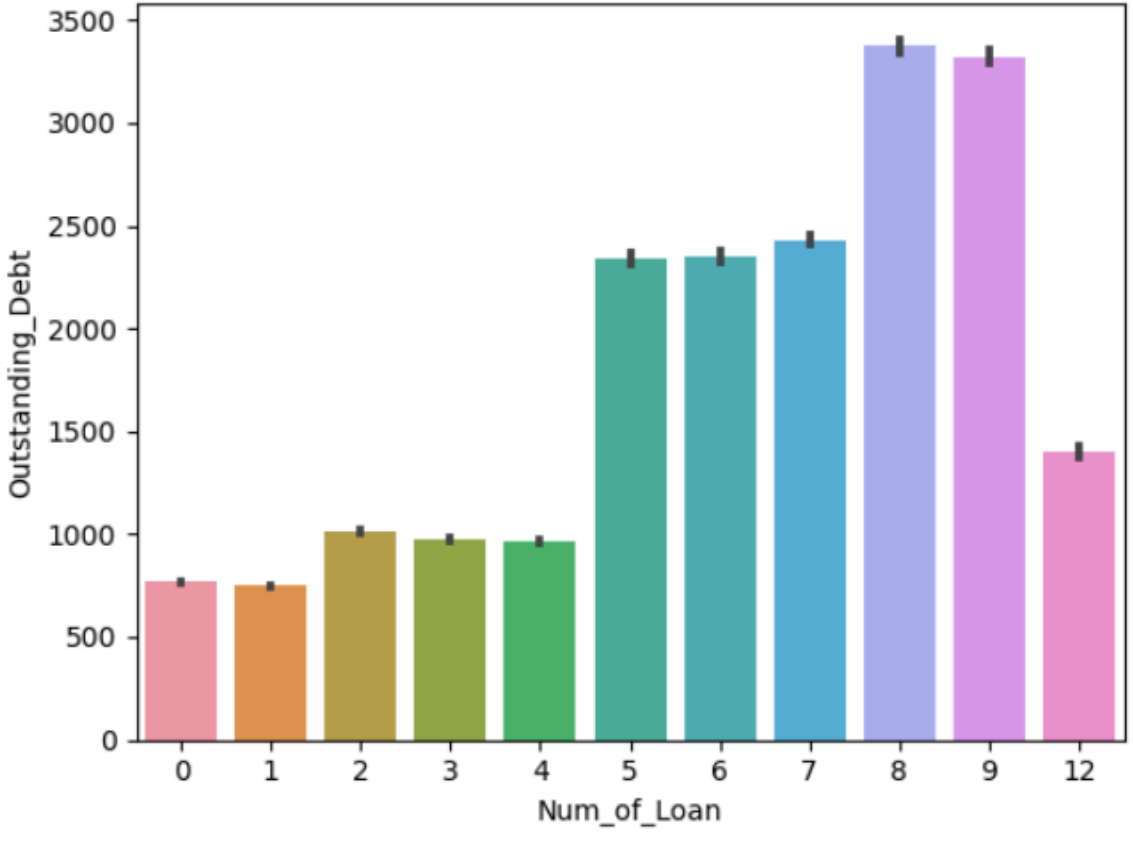
Out[148]:
```
array(['809.98', '605.03', '1303.01', ..., '3571.7_', '3571.7', '502.38'],
      dtype=object)
```

In [149…
```
df['Outstanding_Debt']= df['Outstanding_Debt'].str.replace('_', '')
df['Outstanding_Debt'] = df['Outstanding_Debt'] .str.replace('-', '')
df['Outstanding_Debt'] =df['Outstanding_Debt'] .astype(float)
```

In [150…
```
df['Outstanding_Debt'].unique()
```

Out[150]:
```
array([ 809.98,  605.03, 1303.01, ...,  620.64, 3571.7 ,  502.38])
```

In [151…
```
sns.barplot(x= df['Num_of_Loan'] , y =df['Outstanding_Debt'] , data =df)
plt.show()
```



19- Credit_History_Age

In [164…
```
df['Credit_History_Age'].unique()
```

Out[164]:
```
array(['22 Years and 1 Months', nan, '22 Years and 3 Months',
       '22 Years and 4 Months', '22 Years and 5 Months',
       '22 Years and 6 Months', '22 Years and 7 Months',
       '26 Years and 7 Months', '26 Years and 8 Months',
       '26 Years and 9 Months', '26 Years and 10 Months',
       '26 Years and 11 Months', '27 Years and 0 Months',
       '27 Years and 1 Months', '27 Years and 2 Months',
       '17 Years and 9 Months', '17 Years and 10 Months',
       '17 Years and 11 Months', '18 Years and 1 Months',
       '18 Years and 2 Months', '18 Years and 3 Months',
       '18 Years and 4 Months', '17 Years and 3 Months',
       '17 Years and 4 Months', '17 Years and 5 Months',
       '17 Years and 6 Months', '17 Years and 7 Months',
       '17 Years and 8 Months', '30 Years and 8 Months',
       '30 Years and 9 Months', '30 Years and 10 Months',
       '30 Years and 11 Months', '31 Years and 0 Months',
       '31 Years and 1 Months', '31 Years and 2 Months',
       '31 Years and 3 Months', '32 Years and 0 Months',
       '32 Years and 2 Months', '32 Years and 3 Months',
       '32 Years and 5 Months', '32 Years and 6 Months',
       '30 Years and 7 Months', '14 Years and 8 Months',
       '14 Years and 9 Months', '14 Years and 10 Months',
       '14 Years and 11 Months', '15 Years and 0 Months',
       '15 Years and 1 Months', '15 Years and 2 Months',
       '21 Years and 4 Months', '21 Years and 5 Months',
       '21 Years and 6 Months', '21 Years and 7 Months',
       '21 Years and 8 Months', '21 Years and 9 Months',
       '21 Years and 10 Months', '21 Years and 11 Months',
       '26 Years and 6 Months', '19 Years and 2 Months',
       '19 Years and 3 Months', '19 Years and 4 Months',
       '19 Years and 5 Months', '19 Years and 6 Months',
       '19 Years and 7 Months', '19 Years and 8 Months',
       '25 Years and 5 Months', '25 Years and 6 Months',
       '25 Years and 7 Months', '25 Years and 8 Months',
       '25 Years and 9 Months', '25 Years and 10 Months',
       '25 Years and 11 Months', '26 Years and 0 Months',
       '27 Years and 3 Months', '27 Years and 4 Months',
       '27 Years and 5 Months', '8 Years and 11 Months',
       '9 Years and 0 Months', '9 Years and 1 Months',
       '9 Years and 2 Months', '9 Years and 3 Months',
       '9 Years and 4 Months', '9 Years and 6 Months',
       '18 Years and 5 Months', '18 Years and 6 Months',
       '18 Years and 8 Months', '18 Years and 9 Months',
       '16 Years and 10 Months', '16 Years and 11 Months',
       '17 Years and 0 Months', '17 Years and 1 Months',
```

In [165…
```
df['Credit_History_Age'].isna(). sum()
```

Out[165]:
```
9030
```

In [166…
```
mode = df['Credit_History_Age'].mode()[0]
df['Credit_History_Age'].fillna(mode , inplace =True)
```

In [167…
```
df['Credit_History_Age'].isna(). sum()
```

Out[167]:
```
0
```

In [168…
```
df['Credit_History_Age'].value_counts()
```

Out[168]:
```
15 Years and 11 Months    9476
19 Years and 4 Months      445
19 Years and 5 Months      444
17 Years and 11 Months     443
19 Years and 3 Months      441
                          ...
0 Years and 3 Months        20
0 Years and 2 Months        15
33 Years and 7 Months       14
33 Years and 8 Months       12
0 Years and 1 Months         2
Name: Credit_History_Age, Length: 404, dtype: int64
```

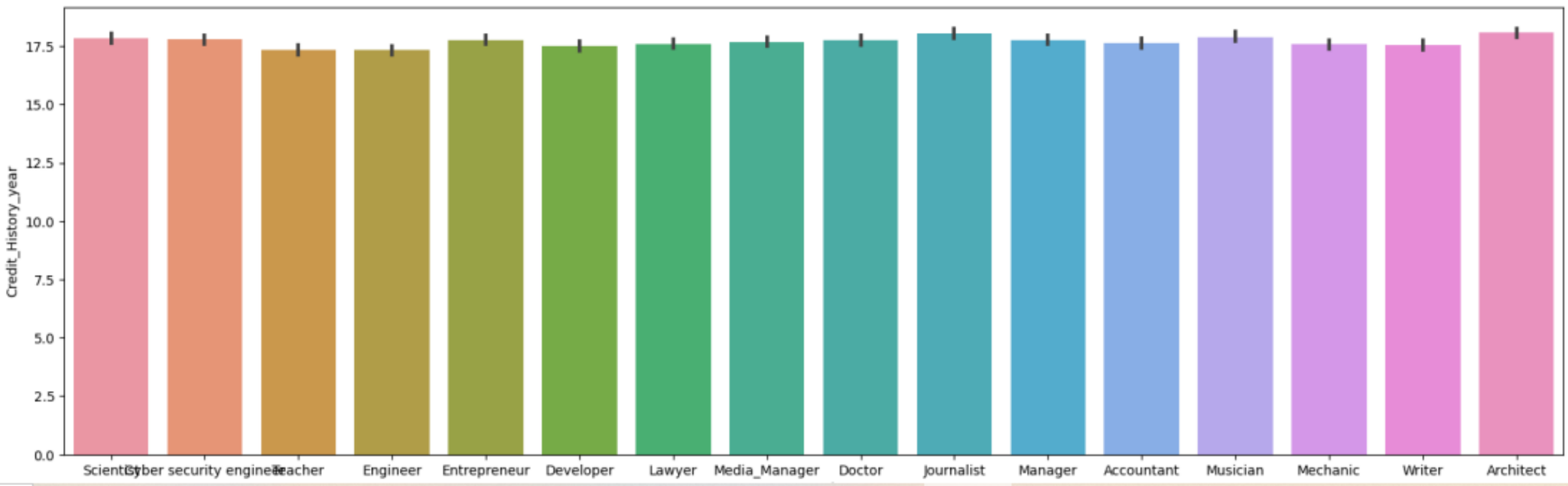
In [169…
```
df['Credit_History_year'],df['Credit_History_month'] =df['Credit_History_Age'].str.split('and' ,2).str
```

In [170…
```
df.drop('Credit_History_Age' , axis =1 ,inplace =True)
```

In [171…
```
df['Credit_History_year'] = df['Credit_History_year'].str.replace('Years','')
df['Credit_History_month'] = df['Credit_History_month'].str.replace('Months','')
```

In [172…
```
df['Credit_History_year'] = df['Credit_History_year'].astype(int)
df['Credit_History_month'] = df['Credit_History_month'].astype(int)
```

In [173…
```
plt.figure(figsize = (20,6))
sns.barplot(x='Occupation',y='Credit_History_year',data =df)
plt.show()
```



Credit_Score (target)

In [213…
```
df = pd.get_dummies(df,columns=['Occupation'],drop_first=True)
df = pd.get_dummies(df , columns= ['Type_of_Loan'] , drop_first =True )
```

In [214…
```
df.head()
```

Out[214]:

	Month	Age	Annual_Income	Monthly_Inhand_Salary	Num_Bank_Accounts	Num_Credit_Card	Interest_Rate	Num_of_Loan	Delay_from_due_date	Num_of_Delayed_Payment	Changed_Credit_Limit	Num_Credit_Inquiries	Credit_
0	1	23.0	19114.12	1824.843333	3	4.0	3	4	3	7.000000	11.27	4.0	
1	2	23.0	19114.12	4194.170850	3	4.0	3	4	0	30.946268	11.27	4.0	
2	3	67.5	19114.12	4194.170850	3	4.0	3	4	3	7.000000	0.00	4.0	
3	4	23.0	19114.12	4194.170850	3	4.0	3	4	5	4.000000	6.27	4.0	
4	5	23.0	19114.12	1824.843333	3	4.0	3	4	6	30.946268	11.27	4.0	

In [215…
```
df['Credit_Score'].unique()
```

Out[215]:
```
array(['Good', 'Standard', 'Poor'], dtype=object)
```

In [216…
```
df['Credit_Score'].value_counts()
```

Out[216]:
```
Standard    53174
Poor        28998
Good        17828
Name: Credit_Score, dtype: int64
```

> - i will use a lable encoder

machine learning part

Random Forest Classifier

In [219…
```
# sns.pairplot(data=df)
```

In [220…
```
X = df.drop('Credit_Score', axis =1).values
y = df['Credit_Score'].values
```

In [221…
```
from sklearn.model_selection import train_test_split
X_train , X_test , y_train , y_test = train_test_split (X , y , random_state=32 , test_size=0.2)
```

In [222…
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [223…
```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(random_state=42 ,n_estimators=100)
model.fit(X_train,y_train)
```

Out[223]:
```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

In [224…
```
y_pred = model.predict(X_test)
y_pred[:5]
```

Out[224]:
```
array([1, 1, 0, 1, 1], dtype=int64)
```

In [225…
```
y_test[:5]
```

Out[225]:
```
array([1, 0, 0, 1, 1], dtype=int64)
```

In [226…
```
from sklearn.metrics import accuracy_score ,f1_score, recall_score, precision_score
```

In [229…
```
print('accuracy score',accuracy_score(y_test , y_pred))
print('f1 score',f1_score(y_test , y_pred, average = 'micro'))
print('recall score',recall_score(y_test , y_pred, average = 'micro'))
print('precision score',precision_score(y_test , y_pred, average = 'micro'))
```

```
accuracy score 0.7804
f1 score 0.7803999999999999
recall score 0.7804
recall score 0.7804
```