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

df=pd.read_csv("/content/Credit Score(in).csv")
df.head()
```

```
<ipython-input-7-66a6d7130ad9>:1: DtypeWarning: Columns (26) have
mixed types. Specify dtype option on import or set low_memory=False.
  df=pd.read_csv("/content/Credit Score(in).csv")
```

```
{"type":"dataframe","variable_name":"df"}
```

```python
df.describe()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"Monthly_Inhand_Salary\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
28721.020500119503,\n        \"min\": 303.6454167,\n        \"max\":
84998.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
4194.170849592996,\n          3093.745,\n          84998.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Num_Bank_Accounts\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 35262.45571123334,\n        \"min\": -
1.0,\n        \"max\": 100000.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          17.09128,\n          6.0,\n
100000.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Num_Credit_Card\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 35274.94819246088,\n        \"min\":
0.0,\n        \"max\": 100000.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          22.47443,\n          5.0,\n
100000.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Interest_Rate\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 35090.285071687635,\n        \"min\":
1.0,\n        \"max\": 100000.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          72.46604,\n          13.0,\n
100000.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Delay_from_due_date\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 35347.57054259866,\n        \"min\": -
5.0,\n        \"max\": 100000.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          21.06878,\n          18.0,\n
100000.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Num_Credit_Inquiries\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 34529.0265942048,\n        \"min\":
```

```
0.0,\n        \"max\": 98035.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          27.75425103279441,\n          6.0,\n
98035.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Credit_Utilization_Ratio\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 35345.02488629802,\n
\"min\": 5.116875070147291,\n        \"max\": 100000.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
32.2851725189402,\n          32.30578367,\n          100000.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Total_EMI_per_month\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
41793.58784756242,\n        \"min\": 0.0,\n        \"max\": 100000.0,\
n        \"num_unique_values\": 8,\n        \"samples\": [\n
1403.1182166160252,\n          69.2494733,\n          100000.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 27 columns):
 #   Column                    Non-Null Count   Dtype  
---  ------                    --------------   -----  
 0   ID                        100000 non-null  object 
 1   Customer_ID               100000 non-null  object 
 2   Month                     100000 non-null  object 
 3   Name                      90015 non-null   object 
 4   Age                       100000 non-null  object 
 5   SSN                       100000 non-null  object 
 6   Occupation                100000 non-null  object 
 7   Annual_Income             100000 non-null  object 
 8   Monthly_Inhand_Salary     84998 non-null   float64
 9   Num_Bank_Accounts         100000 non-null  int64  
 10  Num_Credit_Card           100000 non-null  int64  
 11  Interest_Rate             100000 non-null  int64  
 12  Num_of_Loan               100000 non-null  object 
 13  Type_of_Loan              88592 non-null   object 
 14  Delay_from_due_date       100000 non-null  int64  
 15  Num_of_Delayed_Payment    92998 non-null   object 
 16  Changed_Credit_Limit      100000 non-null  object 
 17  Num_Credit_Inquiries      98035 non-null   float64
 18  Credit_Mix                100000 non-null  object 
 19  Outstanding_Debt          100000 non-null  object 
 20  Credit_Utilization_Ratio  100000 non-null  float64
 21  Credit_History_Age        90970 non-null   object 
 22  Payment_of_Min_Amount     100000 non-null  object 
 23  Total_EMI_per_month       100000 non-null  float64
 24  Amount_invested_monthly   95521 non-null   object 
 25  Payment_Behaviour         100000 non-null  object 
 26  Monthly_Balance           98800 non-null   object 
dtypes: float64(4), int64(4), object(19)
memory usage: 20.6+ MB
```

a.Remove any duplicate entries

```
df.isnull().sum()
```

```
ID                              0
Customer_ID                     0
Month                           0
Name                         9985
Age                             0
SSN                             0
Occupation                      0
Annual_Income                   0
Monthly_Inhand_Salary       15002
Num_Bank_Accounts               0
Num_Credit_Card                 0
Interest_Rate                   0
Num_of_Loan                     0
Type_of_Loan                11408
Delay_from_due_date             0
Num_of_Delayed_Payment       7002
Changed_Credit_Limit            0
Num_Credit_Inquiries         1965
Credit_Mix                      0
Outstanding_Debt                0
Credit_Utilization_Ratio        0
Credit_History_Age           9030
Payment_of_Min_Amount           0
Total_EMI_per_month             0
Amount_invested_monthly      4479
Payment_Behaviour               0
Monthly_Balance              1200
dtype: int64
```

```
df_clean=df.dropna(subset=['Name'])
df_clean.isnull().sum()
```

```
ID                              0
Customer_ID                     0
Month                           0
Name                            0
Age                             0
SSN                             0
Occupation                      0
Annual_Income                   0
```

```
Monthly_Inhand_Salary       13529
Num_Bank_Accounts               0
Num_Credit_Card                 0
Interest_Rate                   0
Num_of_Loan                     0
Type_of_Loan                10280
Delay_from_due_date             0
Num_of_Delayed_Payment       6306
Changed_Credit_Limit            0
Num_Credit_Inquiries         1747
Credit_Mix                      0
Outstanding_Debt                0
Credit_Utilization_Ratio        0
Credit_History_Age           8149
Payment_of_Min_Amount           0
Total_EMI_per_month             0
Amount_invested_monthly      4040
Payment_Behaviour               0
Monthly_Balance              1081
dtype: int64
```

```
df_clean.Monthly_Inhand_Salary.head()
```

```
0    1824.843333
1            NaN
2            NaN
3            NaN
4    1824.843333
Name: Monthly_Inhand_Salary, dtype: float64
```

```
df_clean['Monthly_Inhand_Salary'].fillna(df['Monthly_Inhand_Salary'].m
ean(),inplace=True)
df_clean.isnull().sum()
```

```
<ipython-input-14-f7c27dc5101f>: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

df_clean['Monthly_Inhand_Salary'].fillna(df['Monthly_Inhand_Salary'].m
ean(),inplace=True)
```

```
ID                              0
Customer_ID                     0
Month                           0
Name                            0
Age                             0
SSN                             0
Occupation                      0
```

```
Annual_Income                    0
Monthly_Inhand_Salary            0
Num_Bank_Accounts                0
Num_Credit_Card                  0
Interest_Rate                    0
Num_of_Loan                      0
Type_of_Loan                 10280
Delay_from_due_date              0
Num_of_Delayed_Payment        6306
Changed_Credit_Limit             0
Num_Credit_Inquiries          1747
Credit_Mix                       0
Outstanding_Debt                 0
Credit_Utilization_Ratio         0
Credit_History_Age            8149
Payment_of_Min_Amount            0
Total_EMI_per_month              0
Amount_invested_monthly       4040
Payment_Behaviour                0
Monthly_Balance               1081
dtype: int64
```

```
df.Type_of_Loan.head()
```

```
0    Auto Loan, Credit-Builder Loan, Personal Loan,...
1    Auto Loan, Credit-Builder Loan, Personal Loan,...
2    Auto Loan, Credit-Builder Loan, Personal Loan,...
3    Auto Loan, Credit-Builder Loan, Personal Loan,...
4    Auto Loan, Credit-Builder Loan, Personal Loan,...
Name: Type_of_Loan, dtype: object
```

```
df_clean['Type_of_Loan'].fillna(df['Type_of_Loan'].mode()[0],
inplace=True)
df_clean.isnull().sum()
```

```
<ipython-input-18-46ea1389a53c>: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
  df_clean['Type_of_Loan'].fillna(df['Type_of_Loan'].mode()[0],
inplace=True)
```

```
ID                              0
Customer_ID                     0
Month                           0
Name                            0
Age                             0
SSN                             0
```

```
Occupation                     0
Annual_Income                  0
Monthly_Inhand_Salary          0
Num_Bank_Accounts              0
Num_Credit_Card                0
Interest_Rate                  0
Num_of_Loan                    0
Type_of_Loan                   0
Delay_from_due_date            0
Num_of_Delayed_Payment      6306
Changed_Credit_Limit           0
Num_Credit_Inquiries        1747
Credit_Mix                     0
Outstanding_Debt               0
Credit_Utilization_Ratio       0
Credit_History_Age          8149
Payment_of_Min_Amount          0
Total_EMI_per_month            0
Amount_invested_monthly     4040
Payment_Behaviour              0
Monthly_Balance             1081
dtype: int64
```

```
df_clean['Num_of_Delayed_Payment'].fillna(df['Num_of_Delayed_Payment']
.mode()[0], inplace=True)
df_clean.isnull().sum()
```

```
<ipython-input-26-875cc4bdf33c>: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

df_clean['Num_of_Delayed_Payment'].fillna(df['Num_of_Delayed_Payment']
.mode()[0], inplace=True)

ID                             0
Customer_ID                    0
Month                          0
Name                           0
Age                            0
SSN                            0
Occupation                     0
Annual_Income                  0
Monthly_Inhand_Salary          0
Num_Bank_Accounts              0
Num_Credit_Card                0
Interest_Rate                  0
Num_of_Loan                    0
```

```
Type_of_Loan                   0
Delay_from_due_date            0
Num_of_Delayed_Payment         0
Changed_Credit_Limit           0
Num_Credit_Inquiries        1747
Credit_Mix                     0
Outstanding_Debt               0
Credit_Utilization_Ratio       0
Credit_History_Age          8149
Payment_of_Min_Amount          0
Total_EMI_per_month            0
Amount_invested_monthly     4040
Payment_Behaviour              0
Monthly_Balance             1081
dtype: int64
```

```
df_clean['Num_Credit_Inquiries'].fillna(df_clean['Num_Credit_Inquiries
'].mean(),inplace=True)
df_clean.isnull().sum()
```

```
<ipython-input-35-a093abfef7d4>: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

df_clean['Num_Credit_Inquiries'].fillna(df_clean['Num_Credit_Inquiries
'].mean(),inplace=True)

ID                             0
Customer_ID                    0
Month                          0
Name                           0
Age                            0
SSN                            0
Occupation                     0
Annual_Income                  0
Monthly_Inhand_Salary          0
Num_Bank_Accounts              0
Num_Credit_Card                0
Interest_Rate                  0
Num_of_Loan                    0
Type_of_Loan                   0
Delay_from_due_date            0
Num_of_Delayed_Payment         0
Changed_Credit_Limit           0
Num_Credit_Inquiries           0
Credit_Mix                     0
Outstanding_Debt               0
```

```
Credit_Utilization_Ratio       0
Credit_History_Age          8149
Payment_of_Min_Amount          0
Total_EMI_per_month            0
Amount_invested_monthly     4040
Payment_Behaviour              0
Monthly_Balance             1081
dtype: int64
```

```
df_clean['Num_Credit_Inquiries']=df_clean['Num_Credit_Inquiries'].astype('int64')
```

```
<ipython-input-36-00cd91578798>: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

df_clean['Num_Credit_Inquiries']=df_clean['Num_Credit_Inquiries'].astype('int64')
```

```
df_clean['Credit_History_Age'].value_counts()
```

```
Credit_History_Age
15 Years and 11 Months    402
19 Years and 5 Months     399
17 Years and 11 Months    399
17 Years and 9 Months     395
18 Years and 4 Months     394
                         ...
0 Years and 3 Months       16
0 Years and 2 Months       15
33 Years and 7 Months      13
33 Years and 8 Months      11
0 Years and 1 Months        2
Name: count, Length: 404, dtype: int64
```

```
df_clean['Credit_History_Age'].fillna(df['Credit_History_Age'].mode()
[0], inplace=True)
df_clean.isnull().sum()
```

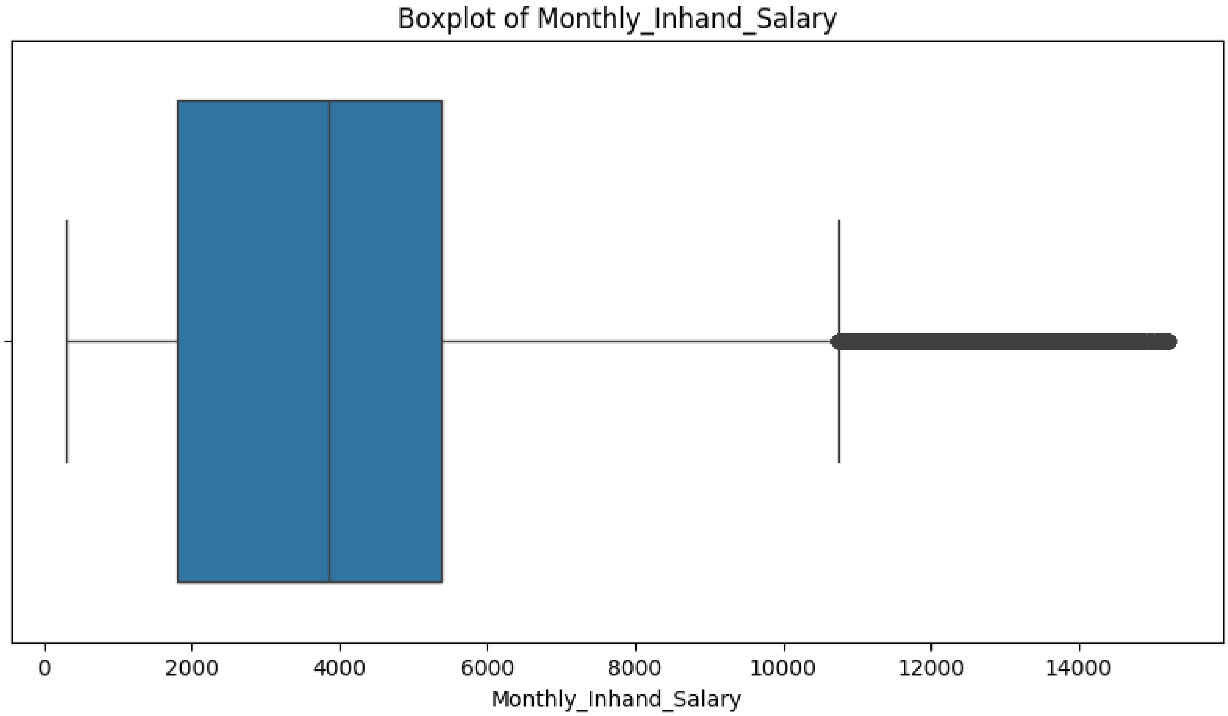
```
<ipython-input-38-0df3d1e37efc>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

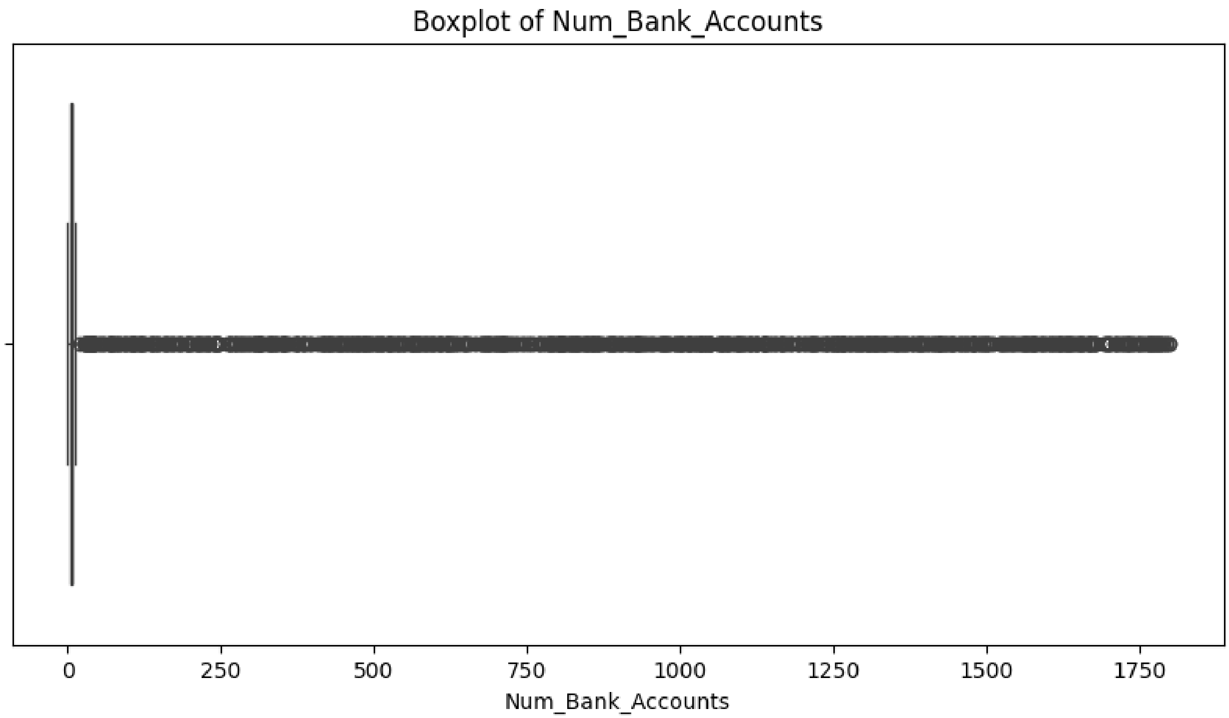
```
df_clean['Credit_History_Age'].fillna(df['Credit_History_Age'].mode()
[0], inplace=True)

ID                             0
Customer_ID                    0
Month                          0
Name                           0
Age                            0
SSN                            0
Occupation                     0
Annual_Income                  0
Monthly_Inhand_Salary          0
Num_Bank_Accounts              0
Num_Credit_Card                0
Interest_Rate                  0
Num_of_Loan                    0
Type_of_Loan                   0
Delay_from_due_date            0
Num_of_Delayed_Payment         0
Changed_Credit_Limit           0
Num_Credit_Inquiries           0
Credit_Mix                     0
Outstanding_Debt               0
Credit_Utilization_Ratio       0
Credit_History_Age             0
Payment_of_Min_Amount          0
Total_EMI_per_month            0
Amount_invested_monthly     4040
Payment_Behaviour              0
Monthly_Balance             1081
dtype: int64

df_clean['Monthly_Balance'].drop

Monthly_Balance
__-333333333333333333333333333__    9
350.0148691                         2
695.0571561                         2
312.4940887                         1
375.4955335                         1
                                   ..
425.5906855                         1
534.716045                          1
586.1116737                         1
195.7701579                         1
393.6736956                         1
Name: count, Length: 88924, dtype: int64
```

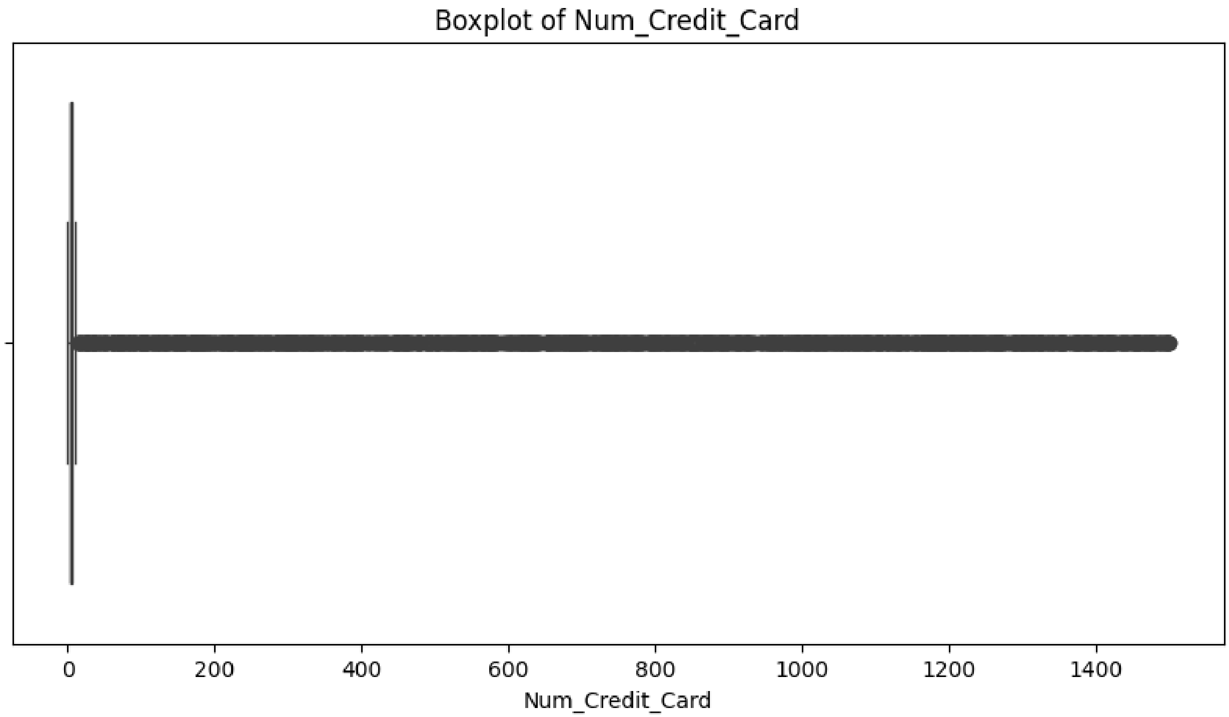
b.Identify and address any missing values, mismatch data types, inconsistencies, or outliers.

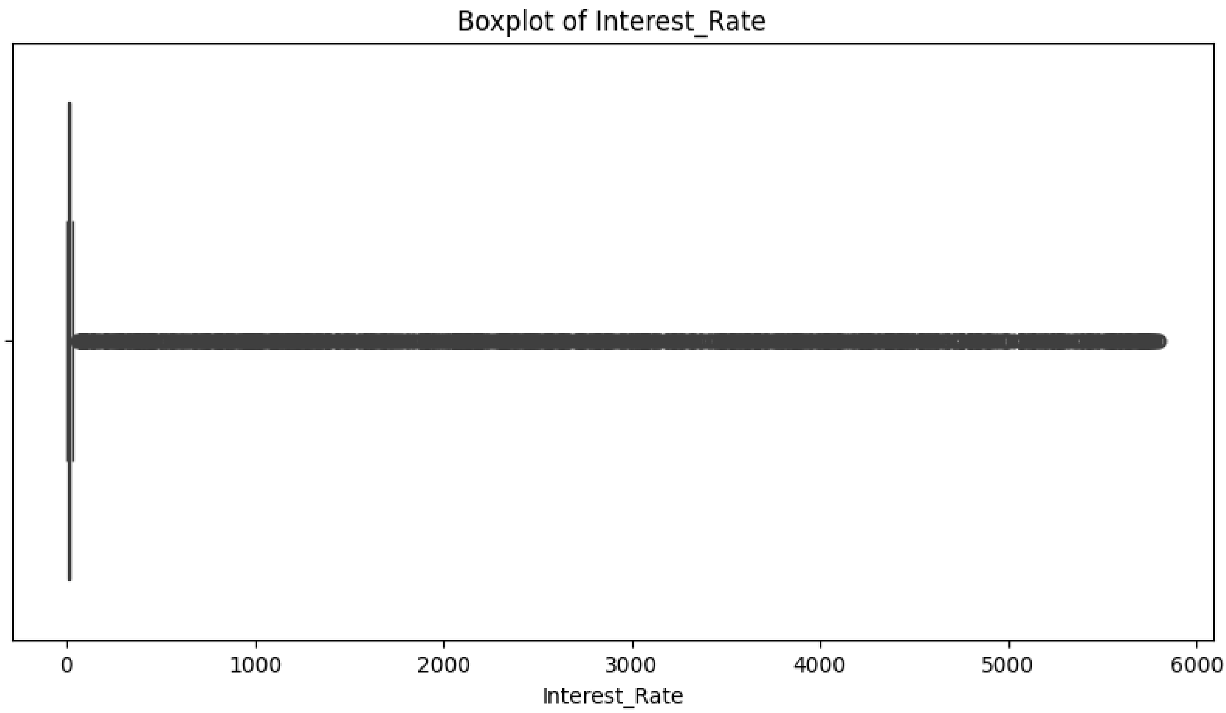
finding outliers

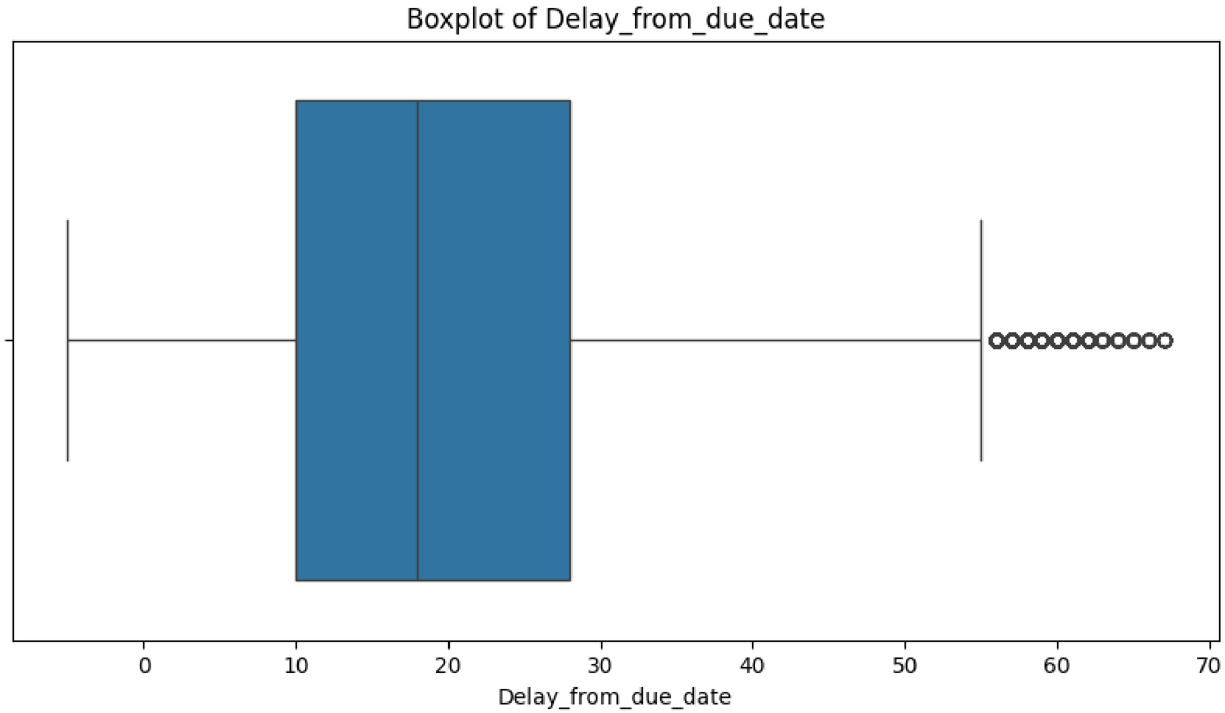
```
def plot_boxplots(df):
    numeric_cols = df.select_dtypes(include=['int64',
'float64']).columns
    for col in numeric_cols:
        plt.figure(figsize=(10, 5))
        sbn.boxplot(x=df[col])
        plt.title(f'Boxplot of {col}')
        plt.show()

plot_boxplots(df_clean)
```

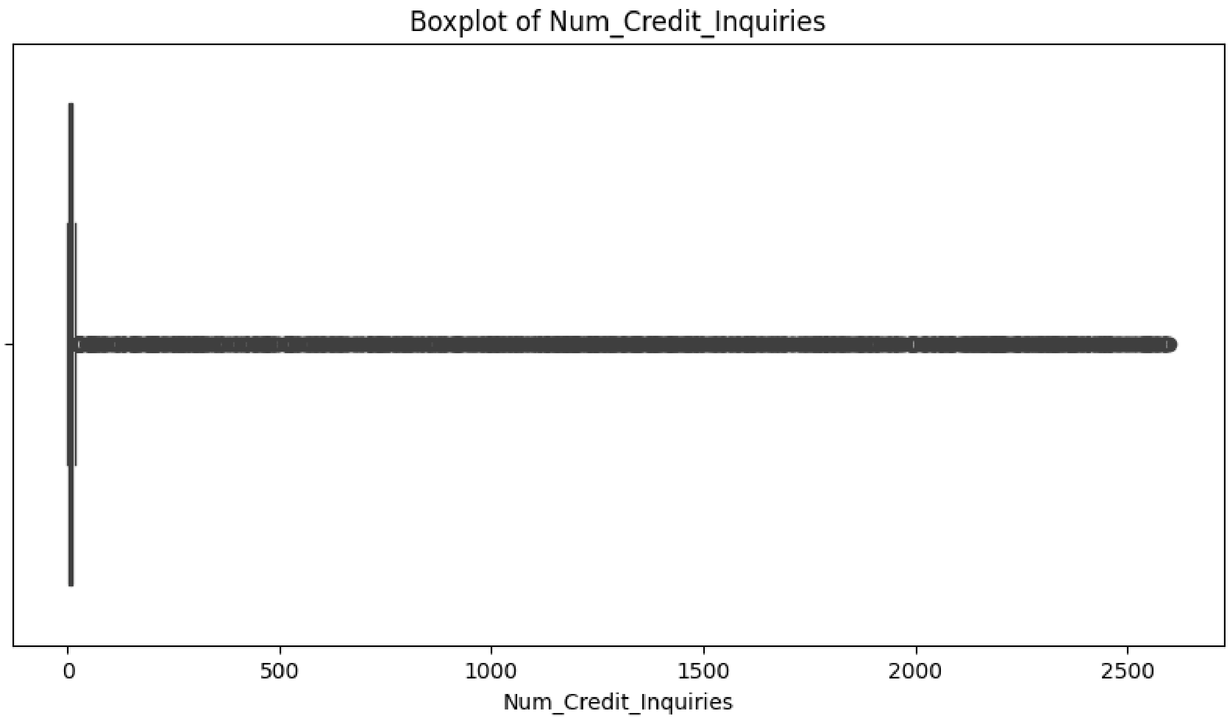


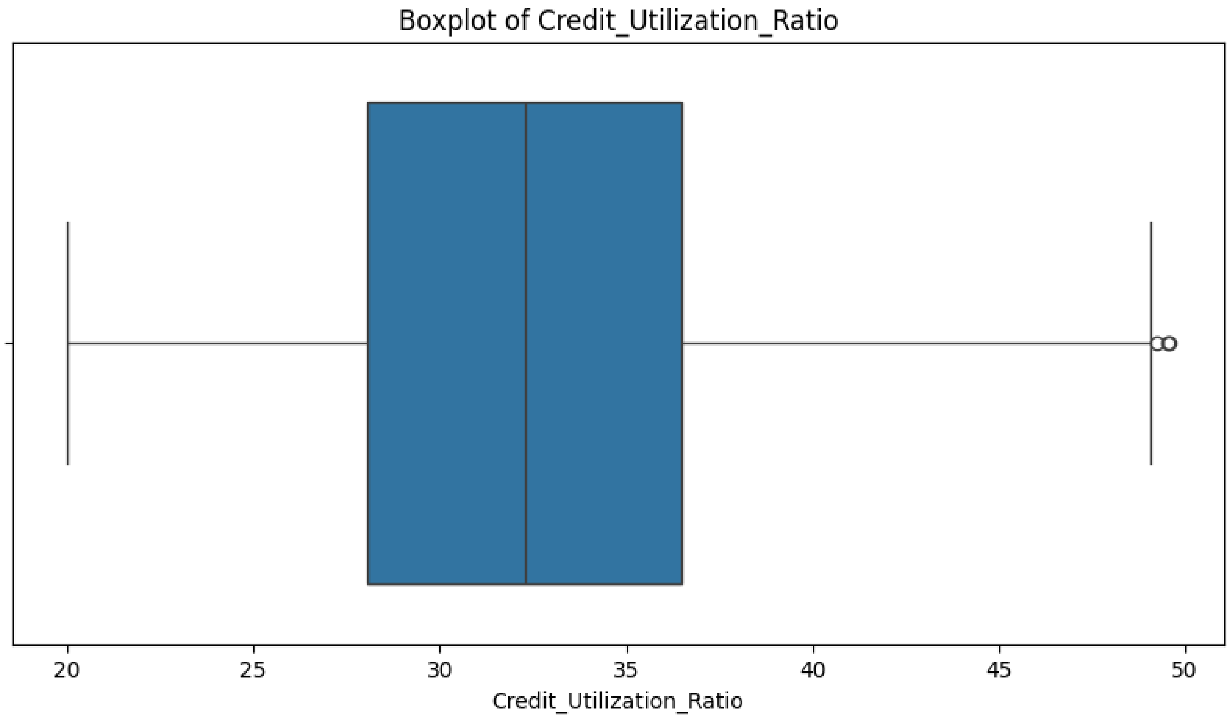

Boxplot of Num_Bank_Accounts

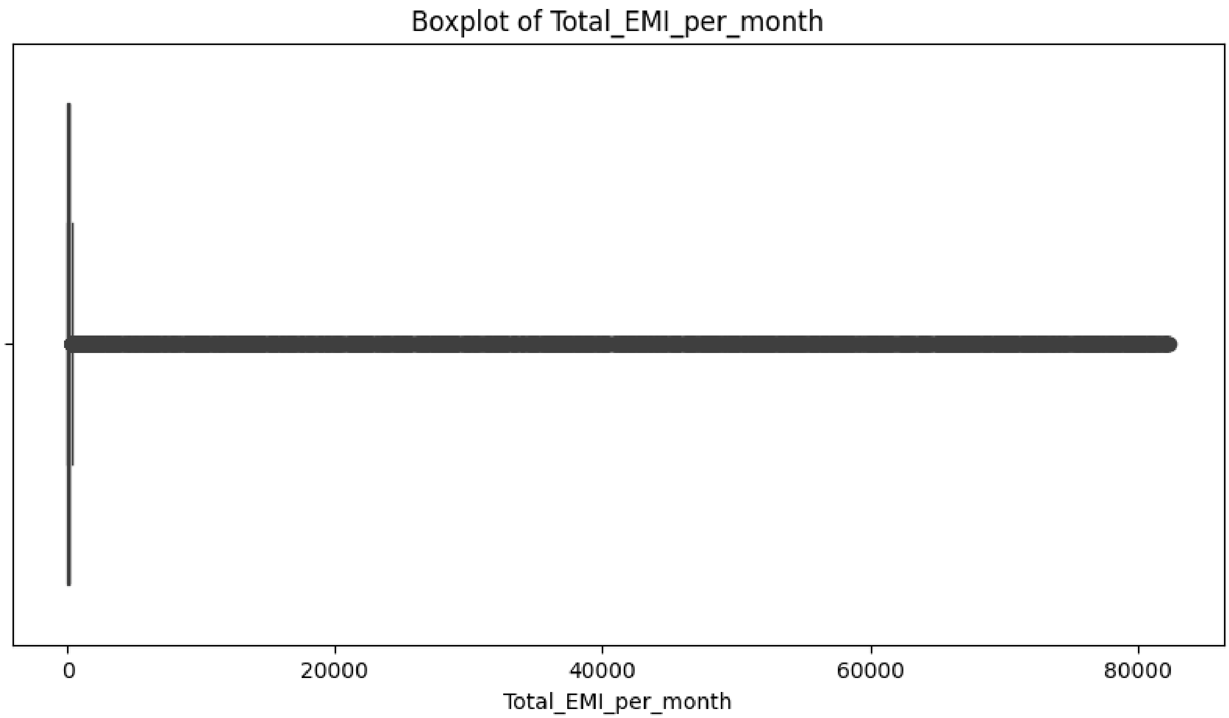

Boxplot of Num_Credit_Card


Boxplot of Interest_Rate
0
1000
2000
3000
4000
5000
6000
Interest_Rate


Boxplot of Delay_from_due_date
0
10
20
30
40
50
60
70
Delay_from_due_date


Boxplot of Num_Credit_Inquiries


Boxplot of Credit_Utilization_Ratio


Boxplot of Total_EMI_per_month
0
20000
40000
60000
80000
Total_EMI_per_month