Loan ApprovalPrediction - Preprocessing Data

Step 1: Import Raw Data and Required Libraries.

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df = pd.read_csv(r'prosperLoanDataRaw.csv')
df.head()
```

Out[2]:

	Term	LoanStatus	EmploymentStatus	EmploymentStatusDuration	IsBorrowerHomeowner	CreditScoreRangeLower	Credit
0	36	Completed	Self-employed	2.0	True	640.0	
1	36	Current	Employed	44.0	False	680.0	
2	36	Completed	Not available	NaN	False	480.0	
3	36	Current	Employed	113.0	True	800.0	
4	36	Current	Employed	44.0	True	680.0	

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113937 entries, 0 to 113936
Data columns (total 18 columns):
 #   Column                    Non-Null Count   Dtype  
---  ------                    --------------   -----  
 0   Term                      113937 non-null  int64  
 1   LoanStatus                113937 non-null  object 
 2   EmploymentStatus          111682 non-null  object 
 3   EmploymentStatusDuration  106312 non-null  float64
 4   IsBorrowerHomeowner       113937 non-null  bool   
 5   CreditScoreRangeLower     113346 non-null  float64
 6   CreditScoreRangeUpper     113346 non-null  float64
 7   OpenCreditLines           106333 non-null  float64
 8   TotalInquiries            112778 non-null  float64
 9   CurrentDelinquencies      113240 non-null  float64
 10  AvailableBankcardCredit   106393 non-null  float64
 11  DebtToIncomeRatio         105383 non-null  float64
 12  IncomeVerifiable          113937 non-null  bool   
 13  StatedMonthlyIncome       113937 non-null  float64
 14  LoanNumber                113937 non-null  int64  
 15  LoanOriginalAmount        113937 non-null  int64  
 16  MonthlyLoanPayment        113937 non-null  float64
 17  BorrowerRate              113937 non-null  float64
dtypes: bool(2), float64(11), int64(3), object(2)
memory usage: 14.1+ MB
```

Step 2: Handling NULL Values.

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
Term                          0
LoanStatus                    0
EmploymentStatus           2255
EmploymentStatusDuration   7625
IsBorrowerHomeowner           0
CreditScoreRangeLower       591
CreditScoreRangeUpper       591
OpenCreditLines            7604
TotalInquiries             1159
CurrentDelinquencies        697
AvailableBankcardCredit    7544
DebtToIncomeRatio          8554
IncomeVerifiable              0
StatedMonthlyIncome           0
LoanNumber                    0
LoanOriginalAmount            0
MonthlyLoanPayment            0
BorrowerRate                  0
dtype: int64
```

In [5]:
```
df.fillna(df.mean(), inplace=True)
```

```
C:\Users\parka\AppData\Local\Temp\ipykernel_30560\820435583.py:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  df.fillna(df.mean(), inplace=True)
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Term                          0
LoanStatus                    0
EmploymentStatus           2255
EmploymentStatusDuration      0
IsBorrowerHomeowner           0
CreditScoreRangeLower         0
CreditScoreRangeUpper         0
OpenCreditLines               0
TotalInquiries                0
CurrentDelinquencies          0
AvailableBankcardCredit       0
DebtToIncomeRatio             0
IncomeVerifiable              0
StatedMonthlyIncome           0
LoanNumber                    0
LoanOriginalAmount            0
MonthlyLoanPayment            0
BorrowerRate                  0
dtype: int64
```

In [7]:
```
df.dropna(inplace=True)
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
Term                        0
LoanStatus                  0
EmploymentStatus            0
EmploymentStatusDuration    0
IsBorrowerHomeowner         0
CreditScoreRangeLower       0
CreditScoreRangeUpper       0
OpenCreditLines             0
TotalInquiries              0
CurrentDelinquencies        0
AvailableBankcardCredit     0
DebtToIncomeRatio           0
IncomeVerifiable            0
StatedMonthlyIncome         0
LoanNumber                  0
LoanOriginalAmount          0
MonthlyLoanPayment          0
BorrowerRate                0
dtype: int64
```

Step 3: Cleaning Columns in the Dataset.

Column: IsEmployed

In [9]:
```
df.EmploymentStatus.unique()
```

Out[9]:
```
array(['Self-employed', 'Employed', 'Not available', 'Full-time', 'Other',
       'Not employed', 'Part-time', 'Retired'], dtype=object)
```

In [10]:
```
df['IsEmployed'] = np.where((df['EmploymentStatus'] == 'Not employed') | (df['EmploymentStatus'] == 'Ret
df.drop(columns=['EmploymentStatus'], inplace=True)
```

Column: IsHomeowner

In [11]:
```
df.rename(columns={'IsBorrowerHomeowner': 'IsHomeowner'}, inplace=True)
df['IsHomeowner'] = df['IsHomeowner'].astype(int)
```

Column: AverageCreditScore

In [12]:
```
df['AverageCreditScore'] = np.ceil((df['CreditScoreRangeLower'] + df['CreditScoreRangeUpper']) / 2).asty
df.drop(columns=['CreditScoreRangeLower', 'CreditScoreRangeUpper'], inplace=True)
```

Column: CurrentDelinquencies

In [13]:
```
df['AnyDelinquencies'] = np.where(df['CurrentDelinquencies'] == 0, 0, 1)
df.drop(columns=['CurrentDelinquencies'], inplace=True)
```

Column: IncomeVerifiable

In [14]:
```
df['IncomeVerifiable'] = df['IncomeVerifiable'].astype(int)
```

Column: TotalInquiries

In [15]:
```
df['TotalInquiries'] = df['TotalInquiries'].astype(int)
```

Column: LoanStatus

In [16]:
```
df.LoanStatus.unique()
```

Out[16]:
```
array(['Completed', 'Current', 'Past Due (1-15 days)', 'Defaulted',
       'Chargedoff', 'Past Due (16-30 days)', 'Past Due (61-90 days)',
       'Past Due (31-60 days)', 'Past Due (91-120 days)',
       'FinalPaymentInProgress', 'Past Due (>120 days)', 'Cancelled'],
      dtype=object)
```

In [17]:
```
excluded_values = ['Current']
df = df[~df['LoanStatus'].isin(excluded_values)]
```

In [18]:
```
df['GoodLoan'] = np.where((df['LoanStatus'] == 'Completed') | (df['LoanStatus'] == 'FinalPaymentInProgre
df.drop(columns=['LoanStatus'], inplace=True)
```

Renaming Remaining Columns

In [19]:
```
df.rename(columns={'LoanOriginalAmount': 'LoanAmount'}, inplace=True)
df.rename(columns={'MonthlyLoanPayment': 'MonthlyInstallment'}, inplace=True)
df.rename(columns={'BorrowerRate': 'InterestRate'}, inplace=True)
df.rename(columns={'EmploymentStatusDuration': 'MonthsOfEmployementExperience'}, inplace=True)
```

Cleaned Data

In [20]:
```
df.describe()
```

Out[20]:

	Term	MonthsOfEmployementExperience	IsHomeowner	OpenCreditLines	TotalInquiries	AvailableBankcardCre
count	55106.000000	55106.000000	55106.000000	55106.000000	55106.000000	55106.0000
mean	37.248213	83.022587	0.479240	8.450398	6.989184	10804.6308
std	7.800320	81.000658	0.499573	4.740902	8.031658	20839.9311
min	12.000000	0.000000	0.000000	0.000000	0.000000	0.0000
25%	36.000000	24.000000	0.000000	5.000000	2.000000	665.0000
50%	36.000000	63.000000	0.000000	8.000000	5.000000	4368.5000
75%	36.000000	104.000000	1.000000	11.000000	9.000000	11210.2254
max	60.000000	755.000000	1.000000	51.000000	379.000000	646285.0000

Step 4: Export Cleaned Dataset.

In [21]:
```
df.to_csv('prosperLoanDataCleaned.csv', index=False)
```

In [ ]:

In [ ]:

In [ ]: