In [1]:
```
import pandas as pd
import matplotlib as mt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('loan_data.csv')
```

In [3]:
```
df.shape
```

Out[3]:
```
(9578, 14)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9578 entries, 0 to 9577
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   credit.policy      9578 non-null   int64  
 1   purpose            9578 non-null   object 
 2   int.rate           9578 non-null   float64
 3   installment        9578 non-null   float64
 4   log.annual.inc     9578 non-null   float64
 5   dti                9578 non-null   float64
 6   fico               9578 non-null   int64  
 7   days.with.cr.line  9578 non-null   float64
 8   revol.bal          9578 non-null   int64  
 9   revol.util         9578 non-null   float64
 10  inq.last.6mths     9578 non-null   int64  
 11  delinq.2yrs        9578 non-null   int64  
 12  pub.rec            9578 non-null   int64  
 13  not.fully.paid     9578 non-null   int64  
dtypes: float64(6), int64(7), object(1)
memory usage: 1.0+ MB
```

In [5]:
```
df.describe()
```

Out[5]:

	credit.policy	int.rate	installment	log.annual.inc	dti	
count	9578.000000	9578.000000	9578.000000	9578.000000	9578.000000	9578.00
mean	0.804970	0.122640	319.089413	10.932117	12.606679	710.84
std	0.396245	0.026847	207.071301	0.614813	6.883970	37.97
min	0.000000	0.060000	15.670000	7.547502	0.000000	612.00
25%	1.000000	0.103900	163.770000	10.558414	7.212500	682.00
50%	1.000000	0.122100	268.950000	10.928884	12.665000	707.00
75%	1.000000	0.140700	432.762500	11.291293	17.950000	737.00
max	1.000000	0.216400	940.140000	14.528354	29.960000	827.00

In [6]:
```
df.head()
```

Out[6]:

	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	da
0	1	debt_consolidation	0.1189	829.10	11.350407	19.48	737	
1	1	credit_card	0.1071	228.22	11.082143	14.29	707	
2	1	debt_consolidation	0.1357	366.86	10.373491	11.63	682	
3	1	debt_consolidation	0.1008	162.34	11.350407	8.10	712	
4	1	credit_card	0.1426	102.92	11.299732	14.97	667	

```
In [7]: df.tail()
```

Out[7]:

	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico
9573	0	all_other	0.1461	344.76	12.180755	10.39	672
9574	0	all_other	0.1253	257.70	11.141862	0.21	722
9575	0	debt_consolidation	0.1071	97.81	10.596635	13.09	687
9576	0	home_improvement	0.1600	351.58	10.819778	19.18	692
9577	0	debt_consolidation	0.1392	853.43	11.264464	16.28	732

```
In [8]: df.dtypes
```

```
Out[8]: credit.policy          int64
        purpose               object
        int.rate             float64
        installment          float64
        log.annual.inc       float64
        dti                  float64
        fico                   int64
        days.with.cr.line    float64
        revol.bal              int64
        revol.util           float64
        inq.last.6mths         int64
        delinq.2yrs            int64
        pub.rec                int64
        not.fully.paid         int64
        dtype: object
```

```
In [9]: df['credit.policy'].mean()
```

```
Out[9]: 0.8049697222802256
```

```
In [10]: df.min()
```

Out[10]:

```
credit.policy                0
purpose              all_other
int.rate                  0.06
installment              15.67
log.annual.inc        7.547502
dti                        0.0
fico                       612
days.with.cr.line   178.958333
revol.bal                    0
revol.util                 0.0
inq.last.6mths               0
delinq.2yrs                  0
pub.rec                      0
not.fully.paid               0
dtype: object
```

In [11]:

```
df.max()
```

Out[11]:

```
credit.policy                    1
purpose             small_business
int.rate                    0.2164
installment                 940.14
log.annual.inc           14.528354
dti                          29.96
fico                           827
days.with.cr.line      17639.95833
revol.bal                  1207359
revol.util                   119.0
inq.last.6mths                  33
delinq.2yrs                     13
pub.rec                          5
not.fully.paid                   1
dtype: object
```

In [12]:

```
df.mode()
```

Out[12]:

	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days
0	1	debt_consolidation	0.1253	317.72	11.0021	0.0	687	

In [13]:

```
pd.get_dummmies(df,dtype='int')
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[13], line 1
----> 1 pd.get_dummmies(df,dtype='int')

AttributeError: module 'pandas' has no attribute 'get_dummmies'
```

In [18]:

```
df.tail()
```

Out[18]:

	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico
9573	0	all_other	0.1461	344.76	12.180755	10.39	672
9574	0	all_other	0.1253	257.70	11.141862	0.21	722
9575	0	debt_consolidation	0.1071	97.81	10.596635	13.09	687
9576	0	home_improvement	0.1600	351.58	10.819778	19.18	692
9577	0	debt_consolidation	0.1392	853.43	11.264464	16.28	732

In [17]:

```
df["ApplicantIncome"]=df["ApplicantIncome"].fillna(df["ApplicantIncome"].mea
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~/anaconda3/lib/python3.11/site-packages/pandas/core/indexes/base.py:37
91, in Index.get_loc(self, key)
   3790 try:
-> 3791     return self._engine.get_loc(casted_key)
   3792 except KeyError as err:

File index.pyx:152, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:181, in pandas._libs.index.IndexEngine.get_loc()

File pandas/_libs/hashtable_class_helper.pxi:7080, in pandas._libs.hashtabl
e.PyObjectHashTable.get_item()

File pandas/_libs/hashtable_class_helper.pxi:7088, in pandas._libs.hashtabl
e.PyObjectHashTable.get_item()

KeyError: 'ApplicantIncome'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[17], line 1
----> 1 df["ApplicantIncome"]=df["ApplicantIncome"].fillna(df["ApplicantInco
me"].mean())

File ~/anaconda3/lib/python3.11/site-packages/pandas/core/frame.py:3893, in
DataFrame.__getitem__(self, key)
   3891 if self.columns.nlevels > 1:
   3892     return self._getitem_multilevel(key)
-> 3893 indexer = self.columns.get_loc(key)
   3894 if is_integer(indexer):
   3895     indexer = [indexer]

File ~/anaconda3/lib/python3.11/site-packages/pandas/core/indexes/base.py:37
98, in Index.get_loc(self, key)
   3793     if isinstance(casted_key, slice) or (
   3794         isinstance(casted_key, abc.Iterable)
   3795         and any(isinstance(x, slice) for x in casted_key)
   3796     ):
   3797         raise InvalidIndexError(key)
-> 3798     raise KeyError(key) from err
   3799 except TypeError:
   3800     # If we have a listlike key, _check_indexing_error will raise
   3801     #  InvalidIndexError. Otherwise we fall through and re-raise
   3802     #  the TypeError.
   3803     self._check_indexing_error(key)

KeyError: 'ApplicantIncome'
```

```
In [ ]:
```