In [ ]:
```
#for installing Libraries
```

In [ ]:
```
#For importing Libraries

import pandas as pd #impoting pandas libraries
import os #importing OS libraries
```

In [ ]:
```
loans_DF=pd.read_csv("datasets/loans.csv") #impoting csv file into pandas
```

In [ ]:
```
#DATA SUMMARY
# The Data has 443 samples starting from 0 to 442 with 8 features
# there is not Nan or missing values
# we have 3 categorical and 5 numerical data types
# observed there are no duplicates in the data
# noticed there are 4 types of loans in the data, home-121 loans, cash-108 loans, c
# observed in repaid feature that 0(unpaid)=206 and 1(paid)=237
```

In [ ]:
```
loans_DF.info() #checking general information
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 443 entries, 0 to 442
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   client_id    443 non-null    int64  
 1   loan_type    443 non-null    object 
 2   loan_amount  443 non-null    int64  
 3   repaid       443 non-null    int64  
 4   loan_id      443 non-null    int64  
 5   loan_start   443 non-null    object 
 6   loan_end     443 non-null    object 
 7   rate         443 non-null    float64
dtypes: float64(1), int64(4), object(3)
memory usage: 27.8+ KB
```

In [ ]:
```
loans_DF.sample(100) #RANDOMLY PICKED 10 DATA SAMPLE
```

Out[ ]:

	client_id	loan_type	loan_amount	repaid	loan_id	loan_start	loan_end	rate
435	26945	home	653	0	11230	2002-08-08	2004-05-01	2.95
392	46958	other	12575	1	11355	2011-09-14	2013-12-21	4.79
287	32885	other	4949	1	10891	2010-07-02	2012-12-26	1.48
259	29841	credit	10537	1	10157	2010-08-04	2013-03-11	3.43
440	26945	other	9329	0	10154	2001-12-17	2004-07-22	5.65
...	...	...	...	...	...	...	...	...
167	35214	home	667	0	11731	2007-02-01	2009-04-06	5.11
32	49545	other	9580	0	11403	2001-07-28	2003-03-15	1.64
404	41472	home	13657	1	10166	2000-04-11	2001-09-08	5.68
361	26695	cash	3660	0	10341	2011-12-19	2013-11-21	2.55
35	49545	credit	11597	0	11234	2001-09-16	2003-07-13	4.38

100 rows × 8 columns

In [ ]:
```
loans_DF.index #checking the range of the data file
```

Out[ ]:
```
RangeIndex(start=0, stop=443, step=1)
```

In [ ]:
```
loans_DF.columns #Checking the data features
```

Out[ ]:
```
Index(['client_id', 'loan_type', 'loan_amount', 'repaid', 'loan_id',
       'loan_start', 'loan_end', 'rate'],
      dtype='object')
```

In [ ]:
```
loans_DF.dtypes
```

Out[ ]:
```
client_id        int64
loan_type       object
loan_amount      int64
repaid           int64
loan_id          int64
loan_start      object
loan_end        object
rate           float64
dtype: object
```

In [ ]:
```
loans_DF.duplicated() #checking duplicated values in the data
```

Out[ ]:
```
0      False
1      False
2      False
3      False
4      False
       ...
438    False
439    False
440    False
441    False
442    False
Length: 443, dtype: bool
```

In [ ]:
```
loan_DF= loans_DF[['loan_amount','rate']] #extracting specifiec features from a dat
```

In [ ]:
```
cols = list(loans_DF.columns) #testing for datatypes in features

numericalCols = []
categoricalCols =[]
for col in cols:
    dataType = str(loans_DF[col].dtype)
    if 'object' in dataType:
        categoricalCols.append(col)
    else:
        numericalCols.append(col)
```

In [ ]:
```
numloans_DF =  loans_DF[['client_id', 'loan_amount', 'repaid', 'loan_id', 'rate']]
```

In [ ]:
```
numloans_DF.sample(6)
```

Out[ ]:

	client_id	loan_amount	repaid	loan_id	rate
71	46180	11679	0	10810	2.34
257	29841	7223	1	11188	5.09
104	39505	14575	1	11650	0.70
412	41472	5554	1	11792	4.60
294	32885	14162	1	11333	7.21
132	32726	851	1	10326	0.66

In [ ]:
```
catloans_DF=loans_DF[['loan_type', 'loan_start', 'loan_end']] #creating a list with
```

In [ ]:
```
catloans_DF.sample(6)
```

Out[ ]:

	loan_type	loan_start	loan_end
236	other	2007-01-06	2008-08-29
270	cash	2006-10-17	2008-06-06
427	home	2003-10-17	2005-11-04
144	credit	2001-03-22	2002-12-15
135	home	2012-01-27	2013-09-15
59	credit	2009-08-02	2012-01-04

In [ ]:
```
loans_DF['repaid'].value_counts() #count the occurence of value in a feature(column
```

Out[ ]:
```
repaid
1    237
0    206
Name: count, dtype: int64
```

In [ ]:
```
loans_DF['loan_type'].value_counts()
```

Out[ ]:
```
loan_type
home      121
cash      108
credit    107
other     107
Name: count, dtype: int64
```