import libraries

In [13]:

```
import pandas as pd
```

import the loan dataset

In [16]:

```
df = pd.read_csv('loan_analysis_dataset.csv')
df
```

Out[16]:

	Id	Address_State	Emp_length	Emp_status	Home_Ownership	Issue_Date	Last_Credit_Pull_Date	Last_Payment_Date
0	1077430	GA	1	Ryder	RENT	2/11/2021	9/13/2021	4/13/2021
1	1072053	CA	9	MKC Accounting	RENT	1/1/2021	12/14/2021	1/15/2021
2	1069243	CA	4	Chemat Technology Inc	RENT	1/5/2021	12/12/2021	1/9/2021
3	1041756	TX	1	barnes distribution	MORTGAGE	2/25/2021	12/12/2021	3/12/2021
4	1068350	IL	10	J&J Steel Inc	MORTGAGE	1/1/2021	12/14/2021	1/15/2021
...	...	...	...	...	...	...	...	...
38571	803452	NJ	1	Joseph M Sanzari Company	MORTGAGE	7/11/2021	5/16/2021	5/16/2021
38572	970377	NY	8	Swat Fame	RENT	10/11/2021	4/16/2021	5/16/2021
38573	875376	CA	5	Anaheim Regional Medical Center	RENT	9/11/2021	5/16/2021	5/16/2021
38574	972997	NY	5	Brooklyn Radiology	RENT	10/11/2021	5/16/2021	5/16/2021
38575	682952	NY	4	Allen Edmonds	RENT	7/11/2021	5/16/2021	5/16/2021

38576 rows × 21 columns

display top 5 rows

In [17]:

```
df.head()
```

Out[17]:

	Id	Address_State	Emp_length	Emp_status	Home_Ownership	Issue_Date	Last_Credit_Pull_Date	Last_Payment_Date	re
0	1077430	GA	1	Ryder	RENT	2/11/2021	9/13/2021	4/13/2021	
1	1072053	CA	9	MKC Accounting	RENT	1/1/2021	12/14/2021	1/15/2021	F
2	1069243	CA	4	Chemat Technology Inc	RENT	1/5/2021	12/12/2021	1/9/2021	
3	1041756	TX	1	barnes distribution	MORTGAGE	2/25/2021	12/12/2021	3/12/2021	F
4	1068350	IL	10	J&J Steel Inc	MORTGAGE	1/1/2021	12/14/2021	1/15/2021	F

5 rows × 21 columns

display last 5 rows

In [18]:

```
df.tail()
```

Out[18]:

	Id	Address_State	Emp_length	Emp_status	Home_Ownership	Issue_Date	Last_Credit_Pull_Date	Last_Payment_Date
38571	803452	NJ	1	Joseph M Sanzari Company	MORTGAGE	7/11/2021	5/16/2021	5/16/2021
38572	970377	NY	8	Swat Fame	RENT	10/11/2021	4/16/2021	5/16/2021
38573	875376	CA	5	Anaheim Regional Medical Center	RENT	9/11/2021	5/16/2021	5/16/2021
38574	972997	NY	5	Brooklyn Radiology	RENT	10/11/2021	5/16/2021	5/16/2021
38575	682952	NY	4	Allen Edmonds	RENT	7/11/2021	5/16/2021	5/16/2021

5 rows × 21 columns

find how many rows and columns are there

In [19]:
```
df.shape
```

Out[19]:
```
(38576, 21)
```

check the data info

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38576 entries, 0 to 38575
Data columns (total 21 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Id                     38576 non-null  int64  
 1   Address_State          38576 non-null  object 
 2   Emp_length             38576 non-null  int64  
 3   Emp_status             37138 non-null  object 
 4   Home_Ownership         38576 non-null  object 
 5   Issue_Date             38576 non-null  object 
 6   Last_Credit_Pull_Date  38576 non-null  object 
 7   Last_Payment_Date      38576 non-null  object 
 8   repayment              38576 non-null  object 
 9   Loan_Category          38576 non-null  object 
 10  Next_Payment_Date      38576 non-null  object 
 11  Member_Id              38576 non-null  int64  
 12  Purpose                38576 non-null  object 
 13  loan_time              38576 non-null  object 
 14  Annual_Income          38576 non-null  float64
 15  DTI                    38576 non-null  object 
 16  Installment            38576 non-null  float64
 17  Int_Rate               38576 non-null  object 
 18  Loan_Amount            38576 non-null  int64  
 19  Total_Acc              38576 non-null  int64  
 20  Total_Payment          38576 non-null  int64  
dtypes: float64(2), int64(6), object(13)
memory usage: 6.2+ MB
```

we covert the Issue date,Last_Credit_Pull_Date, Last_Payment_Date, Next_Payment_Date in to date format

In [27]:
```
df['Issue_Date'] = pd.to_datetime(df['Issue_Date'])
df['Last_Credit_Pull_Date'] = pd.to_datetime(df['Last_Credit_Pull_Date'])
df['Last_Payment_Date'] = pd.to_datetime(df['Last_Payment_Date'])
df['Next_Payment_Date'] = pd.to_datetime(df['Next_Payment_Date'])
```

In [28]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38576 entries, 0 to 38575
Data columns (total 21 columns):
 #   Column                 Non-Null Count  Dtype         
---  ------                 --------------  -----         
 0   Id                     38576 non-null  int64         
 1   Address_State          38576 non-null  object        
 2   Emp_length             38576 non-null  int64         
 3   Emp_status             37138 non-null  object        
 4   Home_Ownership         38576 non-null  object        
 5   Issue_Date             38576 non-null  datetime64[ns]
 6   Last_Credit_Pull_Date  38576 non-null  datetime64[ns]
 7   Last_Payment_Date      38576 non-null  datetime64[ns]
 8   repayment              38576 non-null  object        
 9   Loan_Category          38576 non-null  object        
 10  Next_Payment_Date      38576 non-null  datetime64[ns]
 11  Member_Id              38576 non-null  int64         
 12  Purpose                38576 non-null  object        
 13  loan_time              38576 non-null  object        
 14  Annual_Income          38576 non-null  float64       
 15  DTI                    38576 non-null  object        
 16  Installment            38576 non-null  float64       
 17  Int_Rate               38576 non-null  object        
 18  Loan_Amount            38576 non-null  int64         
 19  Total_Acc              38576 non-null  int64         
 20  Total_Payment          38576 non-null  int64         
dtypes: datetime64[ns](4), float64(2), int64(6), object(9)
memory usage: 6.2+ MB
```

find how many unique,count,top values in each column

In [29]:

```
df.describe(include = 'object').T
```

Out[29]:

	count	unique	top	freq
Address_State	38576	50	CA	6894
Emp_status	37138	28525	US Army	135
Home_Ownership	38576	5	RENT	18439
repayment	38576	3	Fully Paid	32145
Loan_Category	38576	2	Good Loan	33243
Purpose	38576	14	Debt consolidation	18214
loan_time	38576	2	36 months	28237
DTI	38576	301	14.4%	222
Int_Rate	38576	21	11%	4947

checking for null values

In [30]:

```
df.isna().sum()
```

Out[30]:

```
Id                          0
Address_State               0
Emp_length                  0
Emp_status               1438
Home_Ownership              0
Issue_Date                  0
Last_Credit_Pull_Date       0
Last_Payment_Date           0
repayment                   0
Loan_Category               0
Next_Payment_Date           0
Member_Id                   0
Purpose                     0
loan_time                   0
Annual_Income               0
DTI                         0
Installment                 0
Int_Rate                    0
Loan_Amount                 0
Total_Acc                   0
Total_Payment               0
dtype: int64
```

remove the null and missing values

In [31]:

```
df.dropna(inplace = True)
```

check for any duplicate values are present or not

In [32]:

```
df.duplicated()
```

Out[32]:

```
0        False
1        False
2        False
3        False
4        False
         ...
38571    False
38572    False
38573    False
38574    False
38575    False
Length: 37138, dtype: bool
```

now again checking our data is clean or not

In [33]:

```
df.isna().sum()
```

Out[33]:

```
Id                      0
Address_State           0
Emp_length              0
Emp_status              0
Home_Ownership          0
Issue_Date              0
Last_Credit_Pull_Date   0
Last_Payment_Date       0
repayment               0
Loan_Category           0
Next_Payment_Date       0
Member_Id               0
Purpose                 0
loan_time               0
Annual_Income           0
DTI                     0
Installment             0
Int_Rate                0
Loan_Amount             0
Total_Acc               0
Total_Payment           0
dtype: int64
```

in this dataset now no null values,missing values,no duplicated values and no outlier

now we download this cleaned dataset for visualisations

In [34]:

```
df.to_csv('loan_analysis_clean.csv', index=False)
```

In [ ]:

```

```

In [ ]:

```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js