Importing necessary modules

In [1]:
```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

Data

In [2]:
```
bank=pd.read_csv(r"K:\Desktop\NIIT\tables\DS1_C4_S3_Loan_Data_Practice.csv.csv")
bank.columns
```

Out[2]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

Task1: To show loan id and gender of customers who have income less than 6000 and loan approved more than 200K

In [3]:
```
bank[(bank.Loan_Status=="Y")&(bank.ApplicantIncome<6000)&(bank.LoanAmount>200)].iloc[::,[0,1]]
```

Out[3]:

	Loan_ID	Gender
5	LP001011	Male
21	LP001046	Male
159	LP001552	Male
253	LP001843	Male
255	LP001846	Female
276	LP001903	Male
361	LP002170	Male
381	LP002229	Male
502	LP002615	Male
505	LP002622	Male
530	LP002717	Male
562	LP002820	Male

Task2: To compare the number of customers who have properties in urban and semi urban

In [4]:
```
semi=bank.value_counts(bank.Property_Area)[0]
urban=bank.value_counts(bank.Property_Area)[1]
print("The difference between count of semi urban and urban = ",semi-urban)
print("Relative percent to urban is = ",round(((semi-urban)/urban) *100,2),"%")
```

```
The difference between count of semi urban and urban =  31
Relative percent to urban is =  15.35 %
```

Task3: To find records of top 5 customers based on income whose loan was approved

In [5]:
```
bank[bank.Loan_Status=="Y"].sort_values("ApplicantIncome",ascending=False)[:5:]
```

Out[5]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
333	LP002101	Male	Yes	0	Graduate	No	63337	0.0	490	180	1	Urban	Y
171	LP001585	Male	Yes	3+	Graduate	No	51763	0.0	700	300	1	Urban	Y
155	LP001536	Male	Yes	3+	Graduate	No	39999	0.0	600	180	0	Semiurban	Y
185	LP001640	Male	Yes	0	Graduate	Yes	39147	4750.0	120	360	1	Semiurban	Y
443	LP002422	Male	No	1	Graduate	No	37719	0.0	152	360	1	Semiurban	Y

Task4: To find information about incomes and loans of female customers with 2 dependents

In [6]:
```
bank[(bank.Gender=="Female")&(bank.Dependents=="2")].iloc[:,[1,6]]
```

Out[6]:

	Gender	ApplicantIncome
29	Female	3750
82	Female	1378
146	Female	14866
219	Female	4283
251	Female	3427
293	Female	5417
468	Female	210
516	Female	2031

Task5: To update loan amount term to 180 for those who have loan amount more than 200K

In [7]:
```
bank.loc[bank.LoanAmount>200,"Loan_Amount_Term"]=180
```

In [8]:
```
bank.loc[bank.LoanAmount>200]
```

Out[8]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
5	LP001011	Male	Yes	2	Graduate	Yes	5417	4196.0	267	180	1	Urban	Y
9	LP001020	Male	Yes	1	Graduate	No	12841	10968.0	349	180	1	Semiurban	N
21	LP001046	Male	Yes	1	Graduate	No	5955	5625.0	315	180	1	Urban	Y
30	LP001091	Male	Yes	1	Graduate	Yes	4166	3369.0	201	180	1	Urban	N
34	LP001100	Male	No	3+	Graduate	No	12500	3000.0	320	180	1	Rural	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...
592	LP002933	Male	No	3+	Graduate	Yes	9357	0.0	292	180	1	Semiurban	Y
594	LP002938	Male	Yes	0	Graduate	Yes	16120	0.0	260	180	1	Urban	Y
600	LP002949	Female	No	3+	Graduate	Yes	416	41667.0	350	180	0	Urban	N
604	LP002959	Female	Yes	1	Graduate	No	12000	0.0	496	180	1	Semiurban	Y
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253	180	1	Urban	Y

80 rows × 13 columns

Task6: To rename loan_Amount_Term as LoanAmountTerm and replace values 12 and 36 as 60

In [9]:
```
bank.rename(columns = {'Loan_Amount_Term':'LoanAmountTerm'}, inplace = True)
bank.LoanAmountTerm=bank.LoanAmountTerm.replace([12,36],60)
(bank.LoanAmountTerm==12) is True,(bank.LoanAmountTerm==36) is True
```

Out[9]: (False, False)

Task7: To offer credit cards for graduates self employed and have more income than 10K to display loan_id and income along with credit limit for all this customers

In [10]:
```
credit=bank[(bank.Married=="Yes")&(bank.Education=="Graduate")&(bank.ApplicantIncome>10000)]
credit["Credit_Limit"]=credit.ApplicantIncome*2
credit.iloc[:,[0,6]]
```

Out[10]:

	Loan_ID	ApplicantIncome
9	LP001020	12841
54	LP001186	11500
67	LP001233	10750
102	LP001350	13650
106	LP001369	11417
115	LP001401	14583
126	LP001448	23803
128	LP001451	10513
144	LP001508	11757
146	LP001516	14866
155	LP001536	39999
171	LP001585	51763
183	LP001637	33846
185	LP001640	39147
258	LP001859	14683
271	LP001891	11146
278	LP001907	14583
284	LP001922	20667
324	LP002065	15000
333	LP002101	63337
369	LP002191	19730
409	LP002317	81000
424	LP002364	14880
435	LP002393	10047
467	LP002501	16692
475	LP002527	16525
478	LP002531	16667
483	LP002541	10833
487	LP002547	18333
506	LP002624	20833
525	LP002699	17500
557	LP002795	10139
561	LP002813	19484
572	LP002855	16666
594	LP002938	16120
604	LP002959	12000

Task8: To identify and delete records where credit history is zero,dependents more than 3 and not graduates

In [11]:
```
index=bank[(bank.Dependents=="3+")&(bank.Education=="Not Graduate")&(bank.Credit_History==0)].index
bank.drop(index,axis=0,inplace=True)
bank[(bank.Dependents=="3+")&(bank.Education=="Not Graduate")&(bank.Credit_History==0)].index is True
```

Out[11]: False

Task9: To compare customer's income based on thier graduation

In [12]:
```
graduates=bank[bank.Education=="Graduate"].ApplicantIncome.mean()
non_graduates=bank[bank.Education!="Graduate"].ApplicantIncome.mean()
print("The difference in average salaries of graduates and non graduates = ",round(graduates-non_graduates,2))
```

```
The difference in average salaries of graduates and non graduates =  2066.41
```

Task10: To present all outputs in slideshow

In [13]:
```
print("Could not create slideshow due to technical issue")
```

```
Could not create slideshow due to technical issue
```