Task: Determine the eligibility for granting Home loan.

Objective of this notebook is:

1. To understand the patterns in the data.
2. How to Handle the categorical features.
3. How to deal with missing data.
4. Feature Engineering
5. Finding the most important features while taking the decision of granting a loan application.
6. Understanding the Normalization and standardisation of the data.

Load data and libraries

```
import numpy as np
import pandas as pd
from scipy import stats

import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

```
#Data: https://drive.google.com/file/d/1oJbdRpTLqPu1SIBXHkzWRaLaZbvZot7w/view?usp=s
# Download data
id = "1oJbdRpTLqPu1SIBXHkzWRaLaZbvZot7w"
path = "https://docs.google.com/uc?export=download&id=" + id
print(path)
```

```
https://docs.google.com/uc?export=download&id=1oJbdRpTLqPu1SIBXHkzWRaLaZbvZot7
```

```
!wget "https://docs.google.com/uc?export=download&id=1oJbdRpTLqPu1SIBXHkzWRaLaZbvZo
```

```
--2022-05-24 15:41:26--  https://docs.google.com/uc?export=download&id=1oJbdRp
Resolving docs.google.com (docs.google.com)... 172.253.62.101, 172.253.62.113,
Connecting to docs.google.com (docs.google.com)|172.253.62.101|:443... connect
HTTP request sent, awaiting response... 303 See Other
Location: https://doc-0o-90-docs.googleusercontent.com/docs/securesc/ha0ro937g
Warning: wildcards not supported in HTTP.
--2022-05-24 15:41:26--  https://doc-0o-90-docs.googleusercontent.com/docs/sec
Resolving doc-0o-90-docs.googleusercontent.com (doc-0o-90-docs.googleusercont
Connecting to doc-0o-90-docs.googleusercontent.com (doc-0o-90-docs.googleuserc
HTTP request sent, awaiting response... 200 OK
Length: 38011 (37K) [text/csv]
Saving to: ‘train.csv’
```

```
train.csv           100%[===================>]  37.12K  --.-KB/s    in 0s

2022-05-24 15:41:26 (115 MB/s) - 'train.csv' saved [38011/38011]
```

```
!ls -lrt
```

```
total 84
drwxr-xr-x 1 root root  4096 May 17 13:39  sample_data
-rw-r--r-- 1 root root 38011 May 24 12:43 'uc?export=download&id=1oJbdRpTLqPu1
-rw-r--r-- 1 root root 38011 May 24 15:41  train.csv
```

```
!cat train.csv
```

```
LP002795,Male,Yes,3+,Graduate,Yes,10139,0,260,360,1,Semiurban,Y
LP002798,Male,Yes,0,Graduate,No,3887,2669,162,360,1,Semiurban,Y
LP002804,Female,Yes,0,Graduate,No,4180,2306,182,360,1,Semiurban,Y
LP002807,Male,Yes,2,Not Graduate,No,3675,242,108,360,1,Semiurban,Y
LP002813,Female,Yes,1,Graduate,Yes,19484,0,600,360,1,Semiurban,Y
LP002820,Male,Yes,0,Graduate,No,5923,2054,211,360,1,Rural,Y
LP002821,Male,No,0,Not Graduate,Yes,5800,0,132,360,1,Semiurban,Y
LP002832,Male,Yes,2,Graduate,No,8799,0,258,360,0,Urban,N
LP002833,Male,Yes,0,Not Graduate,No,4467,0,120,360,,Rural,Y
LP002836,Male,No,0,Graduate,No,3333,0,70,360,1,Urban,Y
LP002837,Male,Yes,3+,Graduate,No,3400,2500,123,360,0,Rural,N
LP002840,Female,No,0,Graduate,No,2378,0,9,360,1,Urban,N
LP002841,Male,Yes,0,Graduate,No,3166,2064,104,360,0,Urban,N
LP002842,Male,Yes,1,Graduate,No,3417,1750,186,360,1,Urban,Y
LP002847,Male,Yes,,Graduate,No,5116,1451,165,360,0,Urban,N
LP002855,Male,Yes,2,Graduate,No,16666,0,275,360,1,Urban,Y
LP002862,Male,Yes,2,Not Graduate,No,6125,1625,187,480,1,Semiurban,N
LP002863,Male,Yes,3+,Graduate,No,6406,0,150,360,1,Semiurban,N
LP002868,Male,Yes,2,Graduate,No,3159,461,108,84,1,Urban,Y
LP002872,,Yes,0,Graduate,No,3087,2210,136,360,0,Semiurban,N
LP002874,Male,No,0,Graduate,No,3229,2739,110,360,1,Urban,Y
LP002877,Male,Yes,1,Graduate,No,1782,2232,107,360,1,Rural,Y
LP002888,Male,No,0,Graduate,,3182,2917,161,360,1,Urban,Y
LP002892,Male,Yes,2,Graduate,No,6540,0,205,360,1,Semiurban,Y
LP002893,Male,No,0,Graduate,No,1836,33837,90,360,1,Urban,N
LP002894,Female,Yes,0,Graduate,No,3166,0,36,360,1,Semiurban,Y
LP002898,Male,Yes,1,Graduate,No,1880,0,61,360,,Rural,N
LP002911,Male,Yes,1,Graduate,No,2787,1917,146,360,0,Rural,N
LP002912,Male,Yes,1,Graduate,No,4283,3000,172,84,1,Rural,N
LP002916,Male,Yes,0,Graduate,No,2297,1522,104,360,1,Urban,Y
LP002917,Female,No,0,Not Graduate,No,2165,0,70,360,1,Semiurban,Y
LP002925,,No,0,Graduate,No,4750,0,94,360,1,Semiurban,Y
LP002926,Male,Yes,2,Graduate,Yes,2726,0,106,360,0,Semiurban,N
LP002928,Male,Yes,0,Graduate,No,3000,3416,56,180,1,Semiurban,Y
LP002931,Male,Yes,2,Graduate,Yes,6000,0,205,240,1,Semiurban,N
LP002933,,No,3+,Graduate,Yes,9357,0,292,360,1,Semiurban,Y
LP002936,Male,Yes,0,Graduate,No,3859,3300,142,180,1,Rural,Y
LP002938,Male,Yes,0,Graduate,Yes,16120,0,260,360,1,Urban,Y
LP002940,Male,No,0,Not Graduate,No,3833,0,110,360,1,Rural,Y
LP002941,Male,Yes,2,Not Graduate,Yes,6383,1000,187,360,1,Rural,N
LP002943,Male,No,,Graduate,No,2987,0,88,360,0,Semiurban,N
LP002945,Male,Yes,0,Graduate,Yes,9963,0,180,360,1,Rural,Y
```

```
LP002948,Male,Yes,2,Graduate,No,5780,0,192,360,1,Urban,Y
LP002949,Female,No,3+,Graduate,,416,41667,350,180,,Urban,N
LP002950,Male,Yes,0,Not Graduate,,2894,2792,155,360,1,Rural,Y
LP002953,Male,Yes,3+,Graduate,No,5703,0,128,360,1,Urban,Y
LP002958,Male,No,0,Graduate,No,3676,4301,172,360,1,Rural,Y
LP002959,Female,Yes,1,Graduate,No,12000,0,496,360,1,Semiurban,Y
LP002960,Male,Yes,0,Not Graduate,No,2400,3800,,180,1,Urban,N
LP002961,Male,Yes,1,Graduate,No,3400,2500,173,360,1,Semiurban,Y
LP002964,Male,Yes,2,Not Graduate,No,3987,1411,157,360,1,Rural,Y
LP002974,Male,Yes,0,Graduate,No,3232,1950,108,360,1,Rural,Y
LP002978,Female,No,0,Graduate,No,2900,0,71,360,1,Rural,Y
LP002979,Male,Yes,3+,Graduate,No,4106,0,40,180,1,Rural,Y
LP002983,Male,Yes,1,Graduate,No,8072,240,253,360,1,Urban,Y
LP002984,Male,Yes,2,Graduate,No,7583,0,187,360,1,Urban,Y
LP002990,Female,No,0,Graduate,Yes,4583,0,133,360,0,Semiurban,N
```

```
data = pd.read_csv('./train.csv')
data.shape
```

```
(614, 13)
```

```
data.columns
```

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

```
data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIn
0	LP001002	Male	No	0	Graduate	No	
1	LP001003	Male	Yes	1	Graduate	No	
2	LP001005	Male	Yes	0	Graduate	Yes	
3	LP001006	Male	Yes	0	Not Graduate	No	
4	LP001008	Male	No	0	Graduate	No	

```
data.dtypes
#object => typically categorical/IDs
#Int64, Float64
```

```
Loan_ID              object
Gender               object
Married              object
Dependents           object
```

```
Education             object
Self_Employed         object
ApplicantIncome        int64
CoapplicantIncome    float64
LoanAmount           float64
Loan_Amount_Term     float64
Credit_History       float64
Property_Area         object
Loan_Status           object
dtype: object
```

```
data['Dependents'].value_counts()
```

```
0     345
1     102
2     101
3+     51
Name: Dependents, dtype: int64
```

```
# drop loanID column
data = data.drop('Loan_ID',axis = 1)
```

Basic Data Exploration

[ ] ↳ *9 cells hidden*

Basic Data visualization: Univariate

```
data['Loan_Status'].value_counts()
```

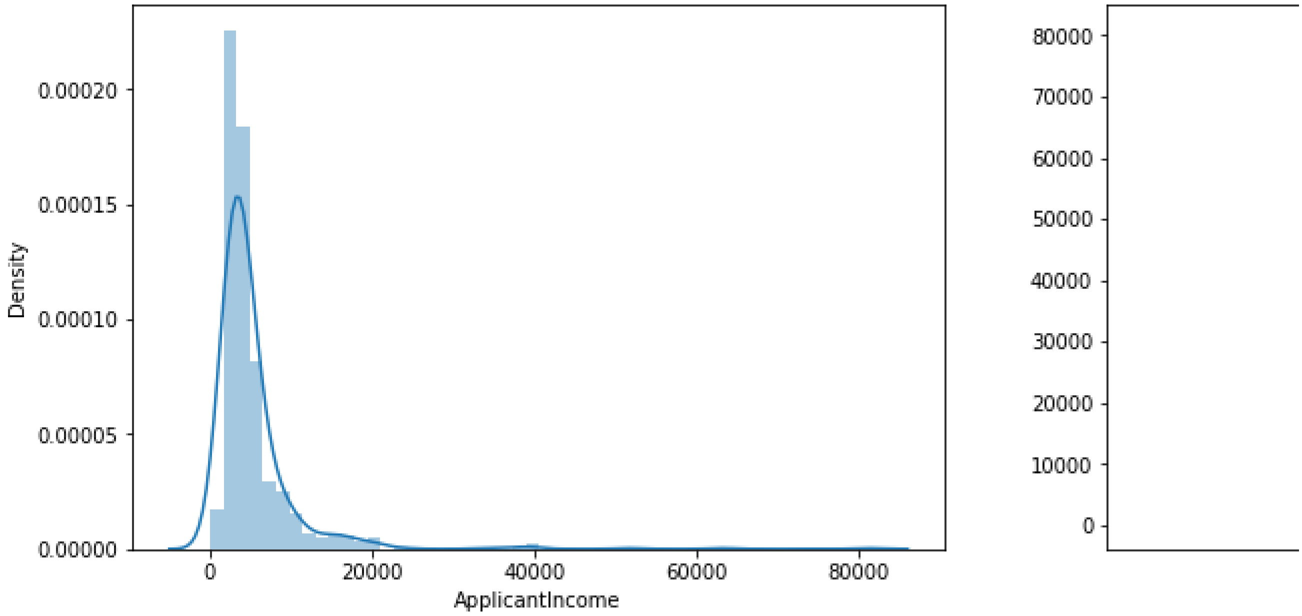
```
Y    422
N    192
Name: Loan_Status, dtype: int64
```

```
#Q: How many loans the company has approved in the past?
sns.countplot(data=data, x='Loan_Status')
plt.show()
```

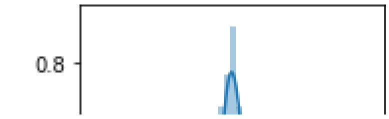

```
target = 'Loan_Status'
data[target].value_counts()

# Imbalanced data
```

```
Y    422
N    192
Name: Loan_Status, dtype: int64
```

```
#Income of the applicant
plt.subplot(121)
sns.distplot(data["ApplicantIncome"])

plt.subplot(122)
data["ApplicantIncome"].plot.box(figsize=(16,5))
plt.show()
```

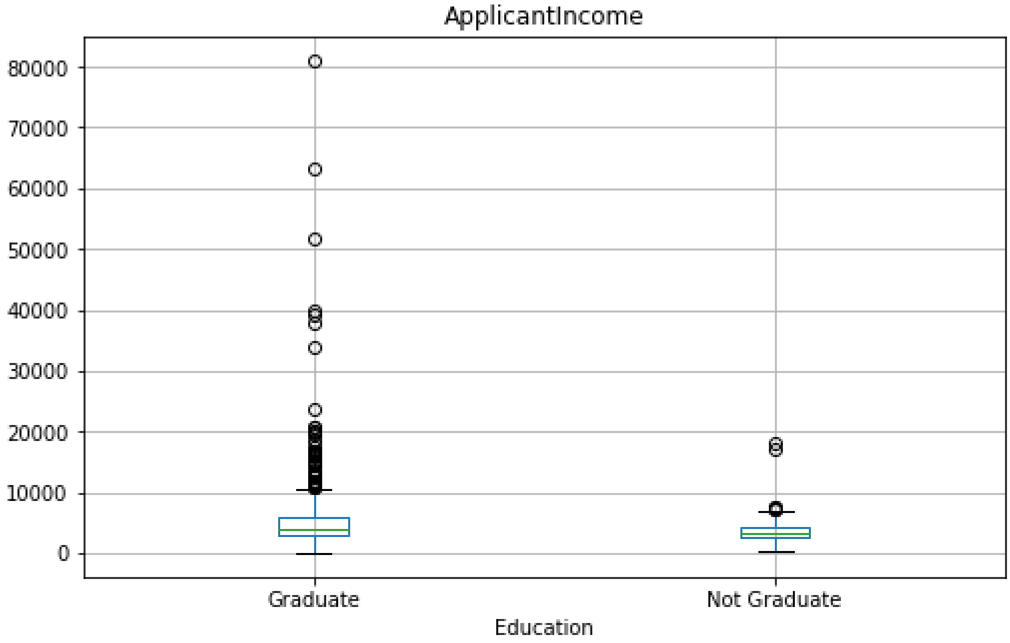



```
plt.subplot(121)
sns.distplot(np.log(data["ApplicantIncome"]))

plt.show()
```

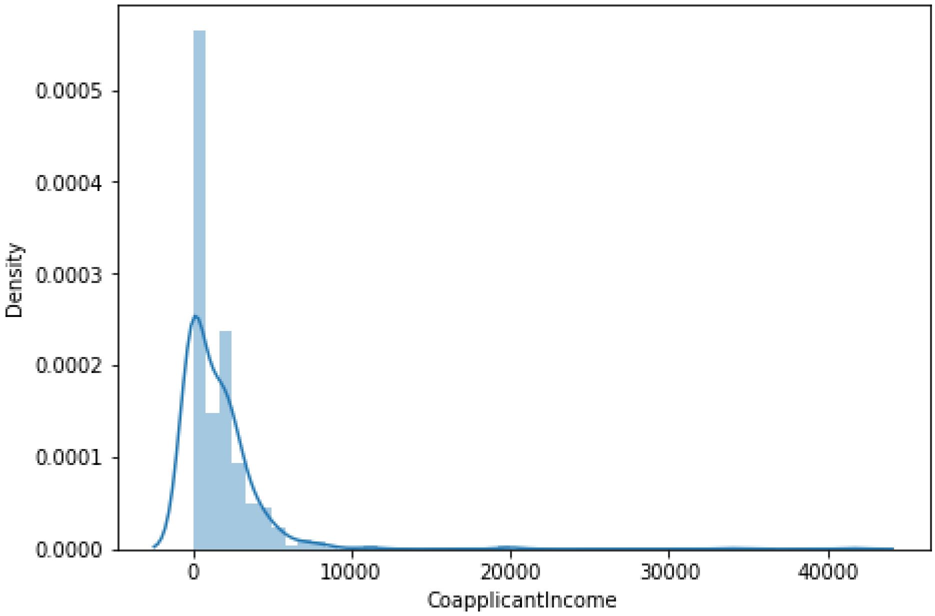


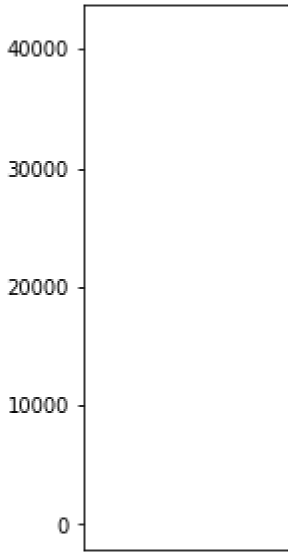

```
#Slice this data by Education
```


```
data.boxplot(column='ApplicantIncome', by="Education", figsize=(8,5))
plt.suptitle("")
plt.show()
```




```
#co-applicant income
plt.subplot(121)
sns.distplot(data["CoapplicantIncome"])

plt.subplot(122)
data["CoapplicantIncome"].plot.box(figsize=(16,5))
plt.show()
```



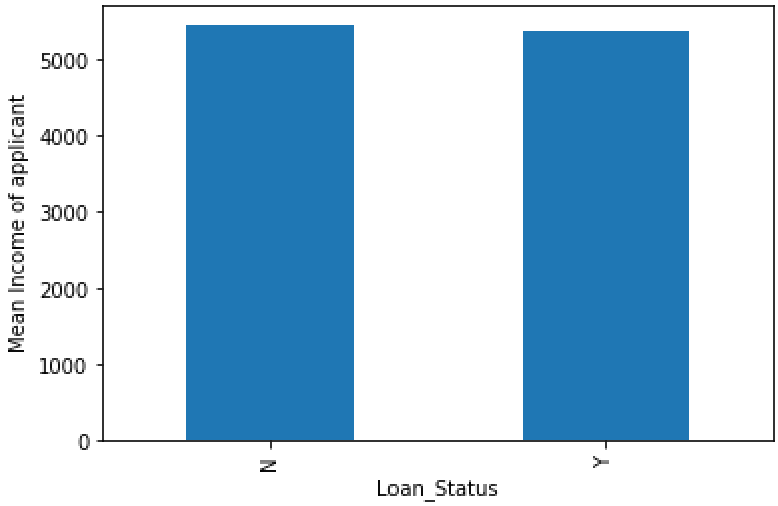


```
#Relation between "Loan_Status" and "Income"
```

```
data.groupby("Loan_Status").mean()['ApplicantIncome']
```

```
Loan_Status
N    5446.078125
Y    5384.068720
Name: ApplicantIncome, dtype: float64
```

```
data.groupby("Loan_Status").mean()['ApplicantIncome'].plot.bar()
plt.ylabel("Mean Income of applicant")
plt.show()
```



Simple Feature Engineering

```
# Feature binning: income
bins=[0,2500,4000,6000, 8000, 10000, 20000, 40000, 81000]
group=['Low','Average','medium', 'H1', 'h2', 'h3', 'h4' , 'Very high']
data['Income_bin']= pd.cut(data['ApplicantIncome'],bins,labels=group)
```

```
data.head()
```

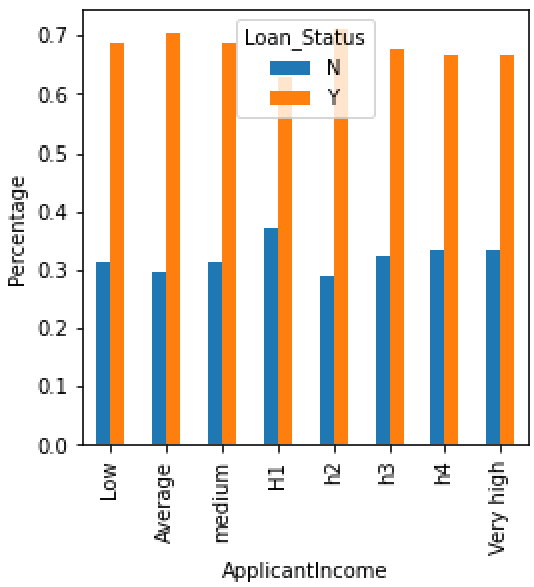
elf_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term
No	5849	0.0	NaN	360.0
No	4583	1508.0	128.0	360.0
Yes	3000	0.0	66.0	360.0
No	2583	2358.0	120.0	360.0
No	6000	0.0	141.0	360.0

Incomes

```
#observed
pd.crosstab(data["Income_bin"],data["Loan_Status"])
```

Loan_Status	N	Y
Income_bin		
Low	34	74
Average	67	159
medium	45	98
H1	20	34
h2	9	22
h3	13	27
h4	3	6
Very high	1	2

```
Income_bin = pd.crosstab(data["Income_bin"],data["Loan_Status"])

Income_bin.div(Income_bin.sum(axis=1),axis=0).plot(kind="bar",figsize=(4,4))
plt.xlabel("ApplicantIncome")
plt.ylabel("Percentage")
plt.show()

#It can be inferred that Applicant income does not affect the chances of loan appro
```



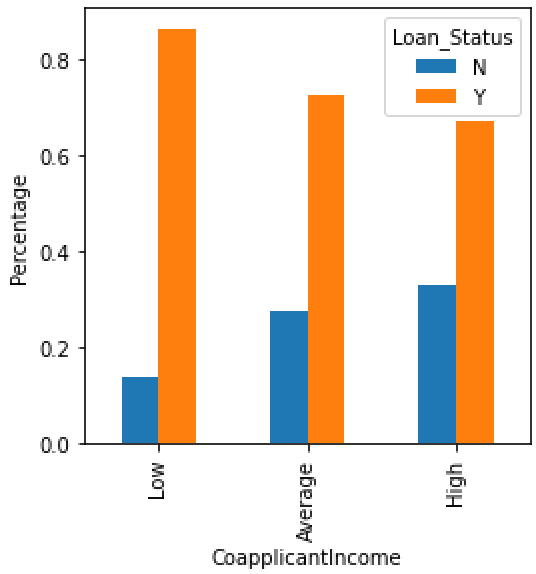

```
#co-appplicant income
```

```
bins=[0,1000,3000,42000]
group =['Low','Average','High']
data['CoapplicantIncome_bin']=pd.cut(data["CoapplicantIncome"],bins,labels=group)
```

```
pd.crosstab(data["CoapplicantIncome_bin"],data["Loan_Status"])
```

Loan_Status	N	Y
CoapplicantIncome_bin		
Low	3	19
Average	61	161
High	32	65

```
CoapplicantIncome_Bin = pd.crosstab(data["CoapplicantIncome_bin"],data["Loan_Status
CoapplicantIncome_Bin.div(CoapplicantIncome_Bin.sum(axis = 1),axis=0).plot(kind='ba
plt.xlabel("CoapplicantIncome")
plt.ylabel("Percentage")
plt.show()

## What's the problem here? Why co-applicant having low income is getting maximum l
```




```
data['CoapplicantIncome'].value_counts().head()
```

```
0.0       273
2500.0      5
2083.0      5
1666.0      5
2250.0      3
Name: CoapplicantIncome, dtype: int64
```

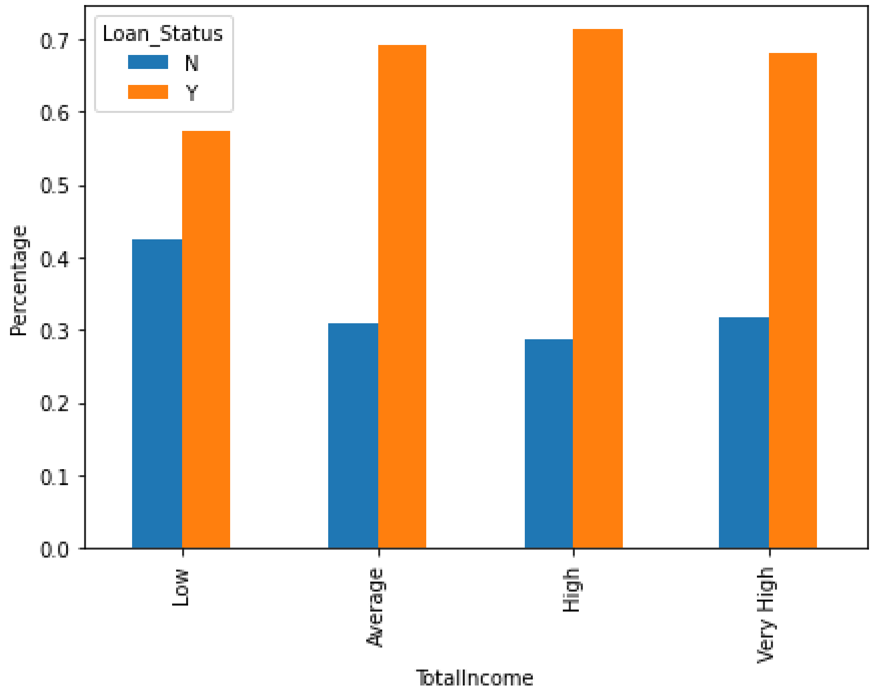
```
# New feature:  total household income
```

```
data["TotalIncome"] = data["ApplicantIncome"] + data["CoapplicantIncome"]

bins = [0,3000,5000,8000,81000]
group = ['Low','Average','High','Very High']
data["TotalIncome_bin"] = pd.cut(data["TotalIncome"],bins,labels=group)
```

```
pd.crosstab(data["TotalIncome_bin"], data["Loan_Status"])
```

Loan_Status	N	Y
TotalIncome_bin		
Low	20	27
Average	69	154
High	61	151
Very High	42	90

```
TotalIncome = pd.crosstab(data["TotalIncome_bin"],data["Loan_Status"])
TotalIncome.div(TotalIncome.sum(axis = 1),axis=0).plot(kind='bar', figsize=(7,5))
plt.xlabel("TotalIncome")
plt.ylabel("Percentage")
plt.show()

# Observation: We can see that Proportion of loans getting approved for
# applicants having low Total_Income is very less as compared to that of applicants
# with Average, High and Very High Income.
```




```
data = data.drop(["Income_bin","CoapplicantIncome_bin"],axis=1)
```

Loan Amount and Loan Term

[ ] ↳ 11 cells hidden

Dependents and Loan **approval**

[ ] ↳ 6 cells hidden

Credit Score vs Loan Approval

[ ] ↳ 3 cells hidden

Missing Values & Data Cleaning

[ ] ↳ 9 cells hidden

Categorical to Numerical encoding

1. One Hot Encoding
2. Label encoding
3. Target Encoding

```
from sklearn.preprocessing import OneHotEncoder

s = (data.dtypes == 'object')
object_cols = list(s[s].index)
object_cols
```

```
['Gender',
 'Married',
 'Education',
 'Self_Employed',
 'Property_Area',
 'Loan_Status']
```

```
data['Self_Employed'].value_counts()
```

```
## Column Standarization
```

Column Standarization

0s completed at 23:24