

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df = pd.read_csv('LoanData.csv')
df.head()
```

Out[2]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
#   Column                      Non-Null Count  Dtype
---  -
0   Loan_ID                     614 non-null    object
1   Gender                      601 non-null    object
2   Married                     611 non-null    object
3   Dependents                  599 non-null    object
4   Education                   614 non-null    object
5   Self_Employed               582 non-null    object
6   ApplicantIncome             614 non-null    int64
7   CoapplicantIncome           614 non-null    float64
8   LoanAmount                  592 non-null    float64
9   Loan_Amount_Term            600 non-null    float64
10  Credit_History              564 non-null    float64
11  Property_Area               614 non-null    object
12  Loan_Status                 614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

In [4]:

```
df.shape
```

Out[4]:

```
(614, 13)
```

In [5]:

```
df.isnull().sum()
```

Out[5]:

```
Loan_ID          0
Gender           13
Married          3
Dependents       15
Education        0
Self_Employed    32
ApplicantIncome  0
CoapplicantIncome 0
LoanAmount       22
Loan_Amount_Term 14
Credit_History   50
Property_Area    0
Loan_Status      0
dtype: int64
```

In [6]:

```
df['LoanAmount'] = df['LoanAmount'].fillna(df['LoanAmount'].mean())
```

In [7]:

```
df['Credit_History'] = df['Credit_History'].fillna(df['Credit_History'].median())
```

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
Loan_ID          0
Gender           13
Married          3
Dependents       15
Education        0
Self_Employed    32
ApplicantIncome  0
CoapplicantIncome 0
LoanAmount       0
Loan_Amount_Term 14
Credit_History   0
Property_Area    0
Loan_Status      0
dtype: int64
```

In [9]:

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

In [10]:

```
df.isnull().sum()
```

Out[10]:

Loan_ID	0
Gender	0
Married	0
Dependents	0
Education	0
Self_Employed	0
ApplicantIncome	0
CoapplicantIncome	0
LoanAmount	0
Loan_Amount_Term	0
Credit_History	0
Property_Area	0
Loan_Status	0
dtype:	int64

In [11]:

```
df.shape
```

Out[11]:

```
(542, 13)
```

In [12]:

```
plt.figure(figsize = (100, 50))
sns.set(font_scale = 5)
plt.subplot(331)
sns.countplot(df[ 'Gender' ],hue=df[ 'Loan_Status' ])

plt.subplot(332)
sns.countplot(df[ 'Married' ],hue=df[ 'Loan_Status' ])

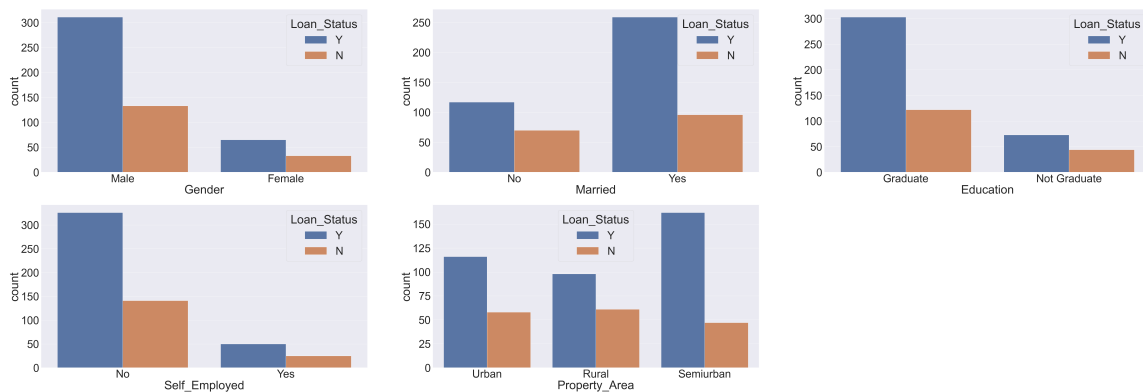
plt.subplot(333)
sns.countplot(df[ 'Education' ],hue=df[ 'Loan_Status' ])

plt.subplot(334)
sns.countplot(df[ 'Self_Employed' ],hue=df[ 'Loan_Status' ])

plt.subplot(335)
sns.countplot(df[ 'Property_Area' ],hue=df[ 'Loan_Status' ])
```

Out[12]:

<AxesSubplot:xlabel='Property_Area', ylabel='count'>



In [13]:

```
df[ 'Loan_Status' ].replace('Y',1,inplace=True)
df[ 'Loan_Status' ].replace('N',0,inplace=True)
```

In [14]:

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

Out[14]:

```
1    376
0    166
Name: Loan_Status, dtype: int64
```

In [15]:

```
df.Gender=df.Gender.map({'Male':1,'Female':0})
df[ 'Gender' ].value_counts()
```

Out[15]:

```
1    444
0     98
Name: Gender, dtype: int64
```

In [16]:

```
df.Married=df.Married.map({'Yes':1,'No':0})
df['Married'].value_counts()
```

Out[16]:

```
1    355
0    187
Name: Married, dtype: int64
```

In [17]:

```
df.Dependents=df.Dependents.map({'0':0,'1':1,'2':2,'3+':3})
df['Dependents'].value_counts()
```

Out[17]:

```
0    309
1     94
2     94
3     45
Name: Dependents, dtype: int64
```

In [18]:

```
df.Education=df.Education.map({'Graduate':1,'Not Graduate':0})
df['Education'].value_counts()
```

Out[18]:

```
1    425
0    117
Name: Education, dtype: int64
```

In [19]:

```
df.Self_Employed=df.Self_Employed.map({'Yes':1,'No':0})
df['Self_Employed'].value_counts()
```

Out[19]:

```
0    467
1     75
Name: Self_Employed, dtype: int64
```

In [20]:

```
df.Property_Area=df.Property_Area.map({'Urban':2,'Rural':0,'Semiurban':1})
df['Property_Area'].value_counts()
```

Out[20]:

```
1    209
2    174
0    159
Name: Property_Area, dtype: int64
```

In [21]:

```
df['LoanAmount'].value_counts()
```

Out[21]:

```
146.412162    19
120.000000    15
100.000000    14
110.000000    13
187.000000    12
..
280.000000     1
240.000000     1
214.000000     1
59.000000      1
253.000000     1
Name: LoanAmount, Length: 195, dtype: int64
```

In [22]:

```
df['Loan_Amount_Term'].value_counts()
```

Out[22]:

```
360.0    464
180.0     38
480.0     13
300.0     12
84.0       4
120.0       3
240.0       3
60.0        2
36.0         2
12.0         1
Name: Loan_Amount_Term, dtype: int64
```

In [23]:

```
df['Credit_History'].value_counts()
```

Out[23]:

```
1.0    468
0.0     74
Name: Credit_History, dtype: int64
```

In [24]:

```
df.head()
```

Out[24]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
0	LP001002	1	0	0	1	0	5849	
1	LP001003	1	1	1	1	0	4583	
2	LP001005	1	1	0	1	1	3000	
3	LP001006	1	1	0	0	0	2583	
4	LP001008	1	0	0	1	0	6000	

In [46]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
lr=LogisticRegression()
```

In [58]:

```
x = df.iloc[1:542,1:12].values
y = df.iloc[1:542,12].values
```

In [65]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,random_state=40)
```

In [66]:

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[66]:

```
((432, 11), (109, 11), (432,), (109,))
```

In [67]:

```
model=lr.fit(x_test,y_test)
lr.score(x_test,y_test)*100
```

Out[67]:

```
80.73394495412845
```

In []:

In []:

In []:

In []:

In []: