```
10/23/23, 1:53 PM
   Name: Vidul Bhosale
   Div: A
   Roll No: 15
   Moodle ID: 20102152
   Machine Learning Experiment No: 4
   from google.colab import drive
   drive.mount('/content/drive')
         Mounted at /content/drive
   import numpy as np import
   pandas as pd import
   matplotlib.pyplot as ply
   %matplotlib inline
   import seaborn as sns import
   warnings
   warnings.filterwarnings('ignore'
   df = pd.read_csv('/content/BankDataset.csv')
   df.head()
                    Loan_ID Gender Married Dependents Education Self_Employed ApplicantIncome
          0 LP001002
                          Male
                                     No
                                                   0
                                                        Graduate
                                                                             No
                                                                                              5849
          1 LP001003
                          Male
                                    Yes
                                                        Graduate
                                                                             No
                                                                                              4583
          2 LP001005
                                                   0
                                                                                              3000
                          Male
                                    Yes
                                                        Graduate
                                                                             Yes
          3 LP001006
                          Male
                                    Yes
                                                             Not
                                                                                              2583
                                                        Graduate
        4
   df.info()
         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 614 entries, 0 to 613
         Data columns (total 13 columns):
                       Non-Null Count Dtype
         # Column
                                  -----
                            614 non-null
601 non-null
        0 Loan_ID
                                                 object
        1
            Gender
                                                  object
           Gender 601 non-null
Married 611 non-null
Dependents 599 non-null
Education 614 non-null
Self_Employed 582 non-null
ApplicantIncome 614 non-null
         2 Married
         3
                                                  object
         4
                                                  object
        5
                                                 object
         6
            CoapplicantIncome 614 non-null
         7
                                                 float64
            LoanAmount 592 non-null float64
Loan_Amount_Term 600 non-null float64
        8
           LoanAmount
         9
         10 Credit_History 564 non-null
                                                 float64
                                 614 non-null
                                                  object 12 Loan_Status
                                                                                    614 non-null
         11 Property_Area
             object dtypes: float64(4), int64(1), object(8) memory usage: 62.5+ KB
   df.shape
         (614, 13)
   df.isnull().sum()
         Loan TD
         Gender
                               13
         Married
         Dependents
                               15
         Education
                               0
         Self_Employed
                               32
         ApplicantIncome
                                0
```

CoapplicantIncome

Loan_Amount_Term

LoanAmount

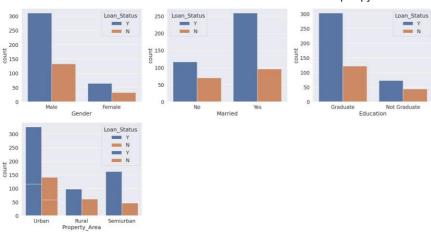
0

22

```
10/23/23, 1:53 PM
```

```
Credit_History
                         50
    Property_Area
                          0
     Loan_Status
     dtype: int64 df['LoanAmount']
     df['LoanAmount'].fillna(df['L
     oanAmount'].mean())
    df['Credit_History'] =
df['Credit_History'].fillna(d
     f['Credit_History'].median())
df.isnull().sum()
    Loan_ID
                          0
     Gender
                         13
    Married
                          3
    Dependents
                         15
     Education
                          0
     Self Employed
                       32
                       0
     ApplicantIncome
     CoapplicantIncome
     LoanAmount
     Loan Amount Term
                         14
    Credit_History
                          a
     Property_Area
                          0
     Loan_Status
                           0
     dtype: int64
df.dropna(inplace=True)
df.isnull().sum()
    Loan_ID
     Gender
     Married
                         0
    Dependents
                         0
    Education
     Self_Employed
                         0
                        0
    ApplicantIncome
    CoapplicantIncome 0
     LoanAmount
                         0
    Loan_Amount_Term
                         0
    Credit_History
                         0
     Property_Area
                         0
     Loan_Status
     dtype: int64
df.shape
     (542, 13)
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(20, 15))
sns.set(font_scale=1.5)
plt.subplot(3,3,1) sns.countplot(x=df['Gender'],
hue=df['Loan_Status'])
plt.subplot(3,3,2) sns.countplot(x=df['Married'],
hue=df['Loan_Status'])
plt.subplot(3,3,3) sns.countplot(x=df['Education'],
hue=df['Loan_Status'])
plt.subplot(3,3,4) sns.countplot(x=df['Self_Employed'],
hue=df['Loan_Status'])
plt.subplot(3,3,4) sns.countplot(x=df['Property_Area'],
hue=df['Loan_Status'])
plt.tight_layout()
plt.show()
```

ML Exp 4.ipynb - Colaboratory



```
df['Loan_Status'].replace('Y', 1, inplace=True)
df['Loan_Status'].replace('N', 0, inplace=True)
df['Loan_Status'].value_counts()
     1
          376
     0
         166
     Name: Loan_Status, dtype: int64
df.Gender=df.Gender.map({'Male':1, 'Female':0})
df['Gender'].value_counts()
     1
          444
          98
     Name: Gender, dtype: int64
df.Married=df.Married.map({'Yes':1, 'No':0})
df['Married'].value_counts()
     1
           355
         187
     Name: Married, dtype: int64
\label{lem:df.Dependents} $$ df.Dependents.map({'0':0, '1':1, '2':2, '3+':3})$ 
df['Dependents'].value_counts()
     0
           309
           94
     1
           94
     2
     3
           45
     Name: Dependents, dtype: int64
df.Education=df.Education.map({'Graduate':1, 'Not Graduate':0})
df['Education'].value_counts()
         425
     1
     0
         117
     Name: Education, dtype: int64
df.Self_Employed=df.Self_Employed.map({'Yes':1, 'No':0})
df['Self_Employed'].value_counts()
     0
           467
           75
     1
     Name: Self_Employed, dtype: int64
df.Property_Area=df.Property_Area.map({'Urban':2,'Rural':0,'Semiurban':1 })
df['Property_Area'].value_counts()
          209
     1
     2
          174
     0
          159
```

```
Name: Property_Area, dtype: int64
df['LoanAmount'].value_counts()
   146.412162
               19
    120.000000
               15
    100.000000
    110,000000
               13
    187.000000
               12
         280.000000
    240.000000
    214.000000
                1
    59.000000
                1
    253.000000
                1
    Name: LoanAmount, Length: 195, dtype: int64
df['Loan_Amount_Term'].value_counts()
    360.0
           464
    180.0
           38
    480.0
    300.0
            12
    84.0
            4
    120.0
            3
    240.0
    60.0
    36.0
            2
    12.0
    Name: Loan_Amount_Term, dtype: int64
df['Credit_History'].value_counts()
   1.0
         468
    0.0
         74
    Name: Credit_History, dtype: int64
df.head()
        Loan_ID Gender Married Dependents Education Self_Employed ApplicantIncome
    0 LP001002
                           0
                                    0
                                             1
                                                                     5849
    1 LP001003
                                                         0
                                                                     4583
                   1
                           1
    2 LP001005
                                                         1
                                                                     3000
    3 LP001006
                   1
                                    0
                                                         0
                                                                     2583
                           1
    4 LP001008
                           0
                                                                     6000
from sklearn.model_selection import train_test_split
from \ sklearn.linear\_model \ import \ Logistic Regression
from sklearn import metrics
X = df.iloc[:, 1:12] # Assuming columns 1 to 11 are your features y
= df['Loan_Status'] # Assuming 'Loan_Status' is your target column
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
model = LogisticRegression()
model.fit(X_train, y_train)
    {\bf \ref{logistic}} Regression
    LogisticRegression()
lr_prediction = model.predict(X_test)
print("y_predicted", lr_prediction)
print("y_test", y_test)
    1 1 0 1 1 1 1 0 1 1 1 1 1 1 1]
    y_test 433
```