Desh Iyer | 500081889 | Experiment - 1

April 2, 2023

1 Experiment 1 - Train a Machine Learning Model to Detect Loan Fraud

1.1 Import Libraries

[ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
```

1.2 Extract Data

[ ]:
```
train = pd.read_csv('data/train.csv')
train.head()
```

[ ]:
```
   Loan_ID Gender Married Dependents     Education Self_Employed  \
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

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0             5849                0.0         NaN             360.0
1             4583             1508.0       128.0             360.0
2             3000                0.0        66.0             360.0
3             2583             2358.0       120.0             360.0
4             6000                0.0       141.0             360.0

   Credit_History Property_Area Loan_Status
0             1.0         Urban           Y
1             1.0         Rural           N
2             1.0         Urban           Y
3             1.0         Urban           Y
4             1.0         Urban           Y
```

[ ]:
```
test = pd.read_csv('data/test.csv')
test.head()
```

[ ]:
```
    Loan_ID Gender Married Dependents     Education Self_Employed  \
0  LP001015   Male     Yes          0      Graduate            No
1  LP001022   Male     Yes          1      Graduate            No
2  LP001031   Male     Yes          2      Graduate            No
3  LP001035   Male     Yes          2      Graduate            No
4  LP001051   Male      No          0  Not Graduate            No

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0             5720                  0       110.0             360.0
1             3076               1500       126.0             360.0
2             5000               1800       208.0             360.0
3             2340               2546       100.0             360.0
4             3276                  0        78.0             360.0

   Credit_History Property_Area
0             1.0         Urban
1             1.0         Urban
2             1.0         Urban
3             NaN         Urban
4             1.0         Urban
```

1.3 Explore the Dataset

[ ]:
```
train.shape
```

[ ]:
```
(614, 13)
```

[ ]:
```
test.shape
```

[ ]:
```
(367, 12)
```

[ ]:
```
list(train.columns)
```

[ ]:
```
['Loan_ID',
 'Gender',
 'Married',
 'Dependents',
 'Education',
 'Self_Employed',
 'ApplicantIncome',
 'CoapplicantIncome',
 'LoanAmount',
 'Loan_Amount_Term',
 'Credit_History',
```

```
 'Property_Area',
 'Loan_Status']
```

[ ]:
```
list(test.columns)
```

[ ]:
```
['Loan_ID',
 'Gender',
 'Married',
 'Dependents',
 'Education',
 'Self_Employed',
 'ApplicantIncome',
 'CoapplicantIncome',
 'LoanAmount',
 'Loan_Amount_Term',
 'Credit_History',
 'Property_Area']
```

[ ]:
```
train.dtypes
```

[ ]:
```
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
```

[ ]:
```
train['Loan_Status'].value_counts()
```

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

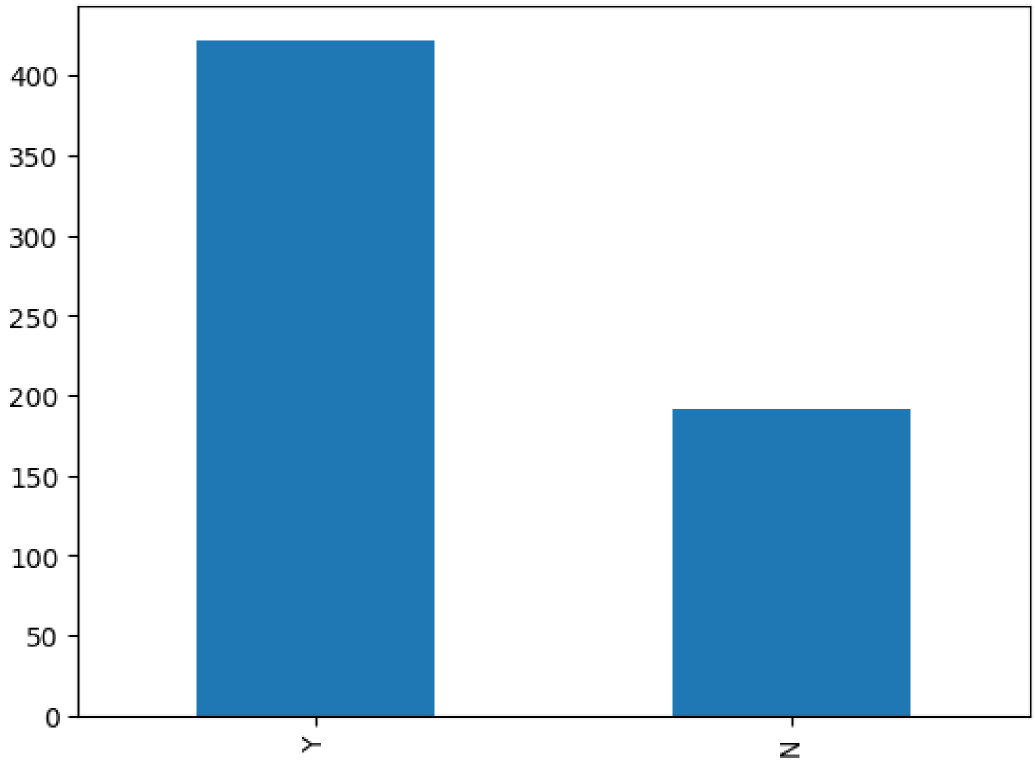
[ ]:
```
train['Loan_Status'].value_counts(normalize=True)
```

[ ]:
```
Y    0.687296
N    0.312704
Name: Loan_Status, dtype: float64
```

1.3.1 Check whether the dataset is balanced or not

[ ]:
```
train['Loan_Status'].value_counts().plot.bar()
```

[ ]: <AxesSubplot: >



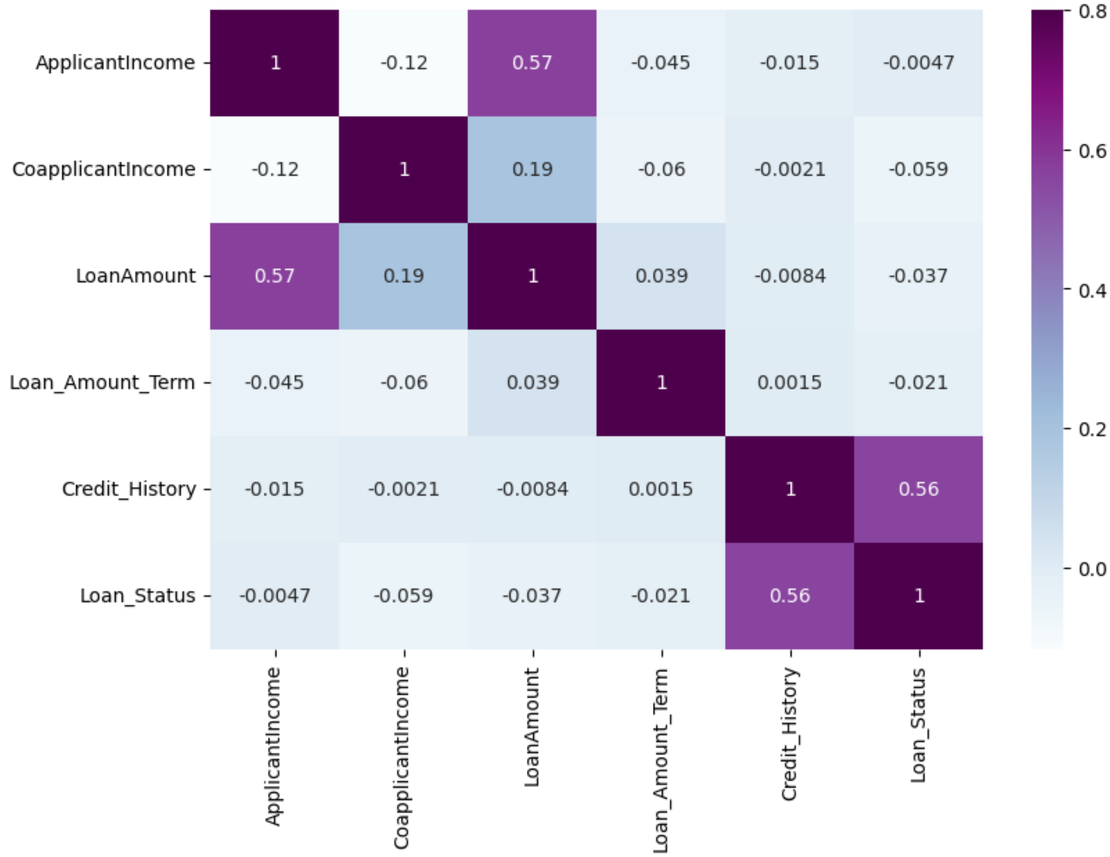
[ ]:
```
train['Dependents'].replace('3+', 3,inplace=True)
test['Dependents'].replace('3+', 3,inplace=True)

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

1.3.2 Plotting a correlation heatmap between the features of the dataset

[ ]:
```
matrix = train.corr()
f, ax = plt.subplots(figsize=(9,6))
sns.heatmap(matrix, vmax=.8, cmap="BuPu", annot = True)
```

```
/tmp/ipykernel_19879/130472854.py:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  matrix = train.corr()
```

[ ]: <AxesSubplot: >



1.3.3 Replacing `null` values with the modes of the respective feature

[ ]:
```
train['Gender'].fillna(train['Gender'].mode()[0], inplace=True)
train['Married'].fillna(train['Married'].mode()[0], inplace=True)
train['Dependents'].fillna(train['Dependents'].mode()[0], inplace=True)
train['Self_Employed'].fillna(train['Self_Employed'].mode()[0], inplace=True)
train['Credit_History'].fillna(train['Credit_History'].mode()[0], inplace=True)

train['Loan_Amount_Term'].fillna(train['Loan_Amount_Term'].mode()[0],␣
↪inplace=True)

train['LoanAmount'].fillna(train['LoanAmount'].median(), inplace=True)
```

[ ]:
```
train.isnull().sum()
```

[ ]:
```
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
```

[ ]:
```
test['Gender'].fillna(train['Gender'].mode()[0], inplace=True)
test['Married'].fillna(train['Married'].mode()[0], inplace=True)
test['Dependents'].fillna(train['Dependents'].mode()[0], inplace=True)
test['Self_Employed'].fillna(train['Self_Employed'].mode()[0], inplace=True)
test['Credit_History'].fillna(train['Credit_History'].mode()[0], inplace=True)
test['Loan_Amount_Term'].fillna(train['Loan_Amount_Term'].mode()[0],␣
 ↪inplace=True)
test['LoanAmount'].fillna(train['LoanAmount'].median(), inplace=True)
```

[ ]:
```
test.isnull().sum()
```

[ ]:
```
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
dtype: int64
```

[ ]:
```
loanID = train['Loan_ID']

train = train.drop('Loan_ID',axis=1)
test = test.drop('Loan_ID',axis=1)
```

[ ]:
```
X = train.drop('Loan_Status', 1)
y = train.Loan_Status
```

```
/tmp/ipykernel_19879/1314552169.py:1: FutureWarning: In a future version of
```

```
pandas all arguments of DataFrame.drop except for the argument 'labels' will be
keyword-only.
  X = train.drop('Loan_Status', 1)
```

1.4 Train-test Split

[ ]:
```python
X = pd.get_dummies(X)
train = pd.get_dummies(train)
test = pd.get_dummies(test)
```

[ ]:
```python
from sklearn.model_selection import train_test_split
x_train, x_cv, y_train, y_cv = train_test_split(X,y, test_size=0.3)
```

1.5 Fitting the Data to a Logistic Regression Model

[ ]:
```python
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

model = LogisticRegression()
model.fit(x_train, y_train)
```

[ ]:
```
LogisticRegression()
```

[ ]:
```python
x_cv
```

[ ]:
```
     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
351             8750             4167.0       308.0             360.0
399             1500             1800.0       103.0             360.0
337             2500             4600.0       176.0             360.0
456             4301                0.0       118.0             360.0
576             3087             2210.0       136.0             360.0
..               …                  …          …                 …
380             3333             2500.0       128.0             360.0
92              3273             1820.0        81.0             360.0
317             2058             2134.0        88.0             360.0
240             5819             5000.0       120.0             360.0
60              2500             3796.0       120.0             360.0

     Credit_History  Gender_Female  Gender_Male  Married_No  Married_Yes  \
351             1.0              0            1           1            0
399             0.0              1            0           1            0
337             1.0              0            1           0            1
456             1.0              0            1           0            1
576             0.0              0            1           0            1
..              …             …          …         …          …
380             1.0              0            1           0            1
92              1.0              0            1           0            1
```

```
317           1.0           0          1          0          1
240           1.0           0          1          0          1
60            1.0           0          1          0          1

     Dependents_3  Dependents_0  Dependents_1  Dependents_2  \
351             0             1             0             0
399             0             1             0             0
337             0             0             0             1
456             0             1             0             0
576             0             1             0             0
..            …            …            …            …
380             0             1             0             0
92              0             0             0             1
317             0             1             0             0
240             0             0             0             1
60              0             1             0             0

     Education_Graduate  Education_Not Graduate  Self_Employed_No  \
351                   1                       0                 1
399                   1                       0                 1
337                   1                       0                 0
456                   1                       0                 1
576                   1                       0                 1
..                  …                      …                …
380                   1                       0                 1
92                    0                       1                 1
317                   1                       0                 1
240                   1                       0                 1
60                    1                       0                 1

     Self_Employed_Yes  Property_Area_Rural  Property_Area_Semiurban  \
351                  0                    1                        0
399                  0                    0                        1
337                  1                    1                        0
456                  0                    0                        0
576                  0                    0                        1
..                 …                   …                       …
380                  0                    0                        1
92                   0                    0                        0
317                  0                    0                        0
240                  0                    1                        0
60                   0                    0                        0

     Property_Area_Urban
351                    0
399                    0
337                    0
```

```
456                    1
576                    0
..                   ...
380                    0
92                     1
317                    1
240                    0
60                     1

[185 rows x 20 columns]
```

1.6 Accuracy of our logistic regression model

[ ]:
```
pred_cv = model.predict(x_cv)

accuracy_score(y_cv, pred_cv)
```

[ ]:
```
0.654054054054054
```

[ ]:
```
test
```

[ ]:
```
     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0               5720                  0       110.0             360.0
1               3076               1500       126.0             360.0
2               5000               1800       208.0             360.0
3               2340               2546       100.0             360.0
4               3276                  0        78.0             360.0
..               ...                ...         ...               ...
362             4009               1777       113.0             360.0
363             4158                709       115.0             360.0
364             3250               1993       126.0             360.0
365             5000               2393       158.0             360.0
366             9200                  0        98.0             180.0

     Credit_History  Gender_Female  Gender_Male  Married_No  Married_Yes  \
0               1.0              0            1           0            1
1               1.0              0            1           0            1
2               1.0              0            1           0            1
3               1.0              0            1           0            1
4               1.0              0            1           1            0
..              ...            ...          ...         ...          ...
362             1.0              0            1           0            1
363             1.0              0            1           0            1
364             1.0              0            1           1            0
365             1.0              0            1           0            1
366             1.0              0            1           1            0
```

```
     Dependents_3  Dependents_0  Dependents_1  Dependents_2  \
0               0             1             0             0
1               0             0             1             0
2               0             0             0             1
3               0             0             0             1
4               0             1             0             0
..            …             …             …             …
362             1             0             0             0
363             0             1             0             0
364             0             1             0             0
365             0             1             0             0
366             0             1             0             0

     Education_Graduate  Education_Not Graduate  Self_Employed_No  \
0                     1                       0                 1
1                     1                       0                 1
2                     1                       0                 1
3                     1                       0                 1
4                     0                       1                 1
..                  …                       …                 …
362                   0                       1                 0
363                   1                       0                 1
364                   1                       0                 1
365                   1                       0                 1
366                   1                       0                 0

     Self_Employed_Yes  Property_Area_Rural  Property_Area_Semiurban  \
0                    0                    0                        0
1                    0                    0                        0
2                    0                    0                        0
3                    0                    0                        0
4                    0                    0                        0
..                 …                    …                        …
362                  1                    0                        0
363                  0                    0                        0
364                  0                    0                        1
365                  0                    1                        0
366                  1                    1                        0

     Property_Area_Urban
0                      1
1                      1
2                      1
3                      1
4                      1
..                   …
362                    1
```

```
363                       1
364                       0
365                       0
366                       0

[367 rows x 20 columns]
```

[ ]:
```
pred_test = model.predict(test)
```

[ ]:
```
prediction = pd.read_csv('data/predicted.csv')
prediction.head()
```

[ ]:
```
    Loan_ID Loan_Status
0  LP001015           N
1  LP001022           N
2  LP001031           N
3  LP001035           N
4  LP001051           N
```

1.6.1 Creating a dataframe for the predicted values from our model

[ ]:
```
prediction['Loan_Status'] = pred_test

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

predictions = pd.DataFrame(prediction, columns=['Loan_ID','Loan_Status'])
predictions
```

[ ]:
```
      Loan_ID Loan_Status
0    LP001015           Y
1    LP001022           Y
2    LP001031           Y
3    LP001035           Y
4    LP001051           Y
..        …          …
362  LP002971           Y
363  LP002975           Y
364  LP002980           Y
365  LP002986           Y
366  LP002989           Y

[367 rows x 2 columns]
```

1.6.2 Writing that prediction dataframe to a .csv file

[ ]:
```
predictions.to_csv('data/logistic.csv')
```