Assignment 1

Name: Navaneeth Shaji

Roll Number: 21CS30032

In [ ]:

```
# import all the necessary libraries here
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score
from sklearn.preprocessing import StandardScaler
```

In [ ]:

```
df = pd.read_csv('../../dataset/cross-validation.csv')
# replacing null values with mode
df = df.fillna(df.mode().iloc[0])
X = df.iloc[:, 1:-1].values
y = df.iloc[:, -1].values
display(df)
m= X.shape[0]
n= X.shape[1]
print(X.shape)

label_encoder = LabelEncoder()
list = [0,1,2,3,4,10]
for i in list:
    x= X[:,i]
    X[:,i] = label_encoder.fit_transform(x)


y = label_encoder.fit_transform(y)
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	120.0	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y
...	...	...	...	...	...	...	...	...	...	...	...	...	...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	360.0	1.0	Rural	Y
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	180.0	1.0	Rural	Y
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	360.0	1.0	Urban	Y
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	360.0	1.0	Urban	Y
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	360.0	0.0	Semiurban	N

614 rows × 13 columns

```
(614, 11)
```

In [ ]:

```
# splitting the dataset into 5 folds

X1, X2, y1, y2 = train_test_split(X, y, test_size=0.2, random_state=0)
X2, X3, y2, y3 = train_test_split(X2, y2, test_size=0.25, random_state=0)
X3, X4, y3, y4 = train_test_split(X3, y3, test_size=0.33, random_state=0)
X4, X5, y4, y5 = train_test_split(X4, y4, test_size=0.5, random_state=0)

# storing the folds in a list
X_list = [X1, X2, X3, X4, X5]
y_list = [y1, y2, y3, y4, y5]
```

In [ ]:

```
# training the model for each fold

# storing the accuracy, precision and recall for each fold
accuracy_list = np.zeros(5)
precision_list = np.zeros(5)
recall_list = np.zeros(5)

for i in range(5):
    X_test = X_list[i]
    y_test = y_list[i]

    X_train = np.array([])
    y_train = np.array([])

    for j in range(5):
        if j != i:
            if X_train.size == 0:
                X_train = X_list[j]
                y_train = y_list[j]
            else:
                X_train = np.concatenate((X_train, X_list[j]))
                y_train = np.concatenate((y_train, y_list[j]))


    # training the model
    classifier = LogisticRegression(random_state=0,solver='saga',max_iter=10000)
    classifier.fit(X_train, y_train)

    # predicting the test set results
    y_pred = classifier.predict(X_test)

    # making the confusion matrix
    from sklearn.metrics import confusion_matrix
    cm = confusion_matrix(y_test, y_pred)

    acc = accuracy_score(y_test, y_pred)
    accuracy_list[i] = acc

    precision = precision_score(y_test, y_pred)
    precision_list[i] = precision

    recall = recall_score(y_test, y_pred)
    recall_list[i] = recall
```

In [ ]:

```
print("Mean Accuracy: ", accuracy_list.mean())
print("Mean Precision: ", precision_list.mean())
print("Mean Recall: ", recall_list.mean())
```

```
Mean Accuracy:  0.7298718970453673
Mean Precision:  0.7384370568993837
Mean Recall:  0.9876328844790928
```

```
In [ ]:
```