

# Assignment 2 – Logistic Regression

## Problem Statement:

You work in XYZ Company as a Python Data Scientist. The company officials have collected some data on health parameters based on diabetes and wish for you to create a model from it. Dataset: diabetes.csv

## Tasks To Be Performed:

1. Load the dataset using pandas
2. Extract data from outcome column is a variable named Y
3. Extract data from every column except outcome column in a variable named X
4. Divide the dataset into two parts for training and testing in 70% and 30% proportion
5. Create and train Logistic Regression Model on training set
6. Make predictions based on the testing set using the trained model
7. Check the performance by calculating the confusion matrix and accuracy score of the model

In [1]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import *
```

In [2]:

```
df = pd.read_csv(r"csv files/diabetes.csv")
df
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

In [3]:

```
X = df.drop(columns=['Outcome'])
y = df['Outcome']
```

In [4]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.30, random_state=42)
```

In [5]:

```
lr = LogisticRegression()
lr.fit(X_train, y_train)
```

C:\Users\Roy\anaconda3\lib\site-packages\sklearn\linear\_model\\_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max\_iter) or scale the data as shown in:  
<https://scikit-learn.org/stable/modules/preprocessing.html>  
Please also refer to the documentation for alternative solver options:  
[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)  
n\_iter\_i = \_check\_optimize\_result(

Out[5]:

▼ LogisticRegression

LogisticRegression()

In [6]:

```
y_pred = lr.predict(X_test)
```

In [7]:

```
confusion_matrix(y_test, y_pred)
```

Out[7]:

```
array([[121,  30],
       [ 30,  50]], dtype=int64)
```

In [8]:

```
accuracy_score(y_test, y_pred)
```

Out[8]: 0.7402597402597403

In [ ]: