In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:

```
diabetes = pd.read_csv('diabetes.csv')
```

In [4]:

```
diabetes.head()
```

Out[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

In [5]:

```
from sklearn.model_selection import train_test_split
```

In [6]:

```
X_train, X_test, y_train, y_test = train_test_split(diabetes.drop('Outcome',ax
is=1), 
                                                    diabetes['Outcome'], test_
size=0.30, 
                                                    random_state=101)
```

In [7]:

```
from sklearn.linear_model import LogisticRegression
```

In [8]:

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

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logis
tic.py:818: 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 opti
ons:
    https://scikit-learn.org/stable/modules/linear_model.html#logi
stic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
```

Out[8]:

```
LogisticRegression()
```

In [9]:

```
predictions = logmodel.predict(X_test)
```

In [10]:

```
from sklearn.metrics import accuracy_score
score=accuracy_score(y_test,predictions)
score
```

Out[10]:

```
0.7922077922077922
```

In [ ]: