In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


df=pd.read_csv('C:\\Users\\sawan\\Downloads\\diabetes.csv')
```

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 |

In [3]:
```
x=df.drop(['Outcome'], axis=1)
```

Out[3]:
```
(768, 8)
```

In [4]:
```
#target variable
y=df.Outcome
```

Out[4]:
```
(768,)
```

In [5]:
```
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
```

In [6]:
```
# Create Decision Tree classifer object
model = DecisionTreeClassifier()

# Train Decision Tree Classifer
model = model.fit(x_train,y_train)

#Predict the response for test dataset
```

In [7]:
```
#Evaluation using Accuracy score
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
```

```
Accuracy: 66.20370370370371
```

In [8]:
```
#Evaluation using Confusion matrix
from sklearn.metrics import confusion_matrix
```

Out[8]:
```
array([[99, 38],
       [35, 44]], dtype=int64)
```

In [9]:
```
#Evaluation using Classification report
from sklearn.metrics import classification_report
```

```
              precision    recall  f1-score   support

           0       0.74      0.72      0.73       137
           1       0.54      0.56      0.55        79

    accuracy                           0.66       216
   macro avg       0.64      0.64      0.64       216
weighted avg       0.66      0.66      0.66       216
```