

8. Develop a program to implement Decision Tree model and analyze the model using confusion matrix.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
import seaborn as sns
```

```
pima = pd.read_csv("diabetes.csv")
```

```
pima
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
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	
..	
763	10	101	76	48	180	32.9	
764	2	122	70	27	0	36.8	
765	5	121	72	23	112	26.2	
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1
..
763	0.171	63	0
764	0.340	27	0
765	0.245	30	0
766	0.349	47	1
767	0.315	23	0

```
[768 rows x 9 columns]
```

```
X = pima.iloc[:, [0,7]].values
y= pima.iloc[:, 8].values
```

```
# Split dataset into training set and test set
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0) # 70% training and 30% test
```

```

# Create Decision Tree classifier object
clf = DecisionTreeClassifier()
# Train Decision Tree Classifier
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("\nConfusion Matrix is:", metrics.confusion_matrix(y_test, y_pred))

Accuracy: 0.6190476190476191

Confusion Matrix is: [[124  33]
 [ 55  19]]

cnf_matrix=pd.crosstab(y_test, y_pred, rownames=['actual'], colnames=['Predicted'])
sns.heatmap(cnf_matrix, annot=True)

<AxesSubplot:xlabel='Predicted', ylabel='actual'>

```

