

1. Write a Python Program to implement Decision tree based classification
Consider 70% of the training data and 30% as test data.

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix

data = load_iris()
x, y = data.data, data.target

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
random_state=42) # defining the test size

clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

confusion = confusion_matrix(y_test, y_pred)

print("Confusion Matrix:")
print(confusion)
```

Confusion Matrix:

```
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

PS C:\Users\Nutakki\Desktop\Semester 5\10. DWDM Lab>

```

import matplotlib.pyplot as plt

class_names = ["Class 0", "Class 1", "Class 2"]
fig, ax = plt.subplots()
cax = ax.matshow(confusion, cmap=plt.cm.Blues)
plt.colorbar(cax)
ax.set_xticklabels([''] + class_names)
ax.set_yticklabels([''] + class_names)
plt.xlabel('Predicted')
plt.ylabel('True')
for i in range(len(class_names)):
    for j in range(len(class_names)):
        plt.text(j, i, str(confusion[i, j]), va='center', ha='center')

plt.show()

```

