PRACTICAL 5

Aim: To implement Decision-Tree for restaurant wait problem.

Code:

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction import DictVectorizer
from sklearn import tree
import warnings

warnings.filterwarnings(action='ignore',category=DeprecationWarning)

data=pd.read_csv('data.csv')
cols_to_retain=['Alt','Bar','Fri','Hun','Pat','Price','Rain','Res','Type','Est
']

X_feature=data[cols_to_retain]
X_dict=X_feature.T.to_dict().values()

vect=DictVectorizer(sparse=False)
X_vector=vect.fit_transform(X_dict)
print(X_vector)

X_Train=X_vector[:-1]
X_Test=X_vector[-1:]
print('Train Set')
print(X_Train)
print('test set')
print(X_Test)

le=LabelEncoder()
y_Train=le.fit_transform(data['Goal'][:-1])

clf=tree.DecisionTreeClassifier(criterion='entropy')
clf=clf.fit(X_Train,y_Train)
print('test data')
print(le.inverse_transform(clf.predict(X_Test)))
```

Output:

```
[Running] python -u "c:\Users\athar\Documents\Practicals\AI Practical\P5\DecisionTree.py"
[[  0.   1.   1.   0.   1.   0.   0.   1.   0.   0.   1.   0.   1. 100.
    1.   0.   0.   1.   0.   1.   0.   0.]
 [  0.   1.   1.   0.   0.   1.   0.   1.   0.   0.   1.   1.   0.  10.
    1.   0.   1.   0.   0.   0.   0.   1.]
 [  1.   0.   0.   1.   1.   0.   0.   1.   0.   1.   0.   0.   1.  10.
    1.   0.   1.   0.   1.   0.   0.   0.]
 [  0.   1.   1.   0.   0.   0.   1.   0.   1.   1.   0.   1.   0. 100.
    1.   0.   0.   1.   0.   1.   0.   0.]
 [  1.   0.   0.   1.   1.   0.   0.   1.   0.   0.   1.   0.   1.  75.
    0.   1.   0.   1.   0.   0.   1.   0.]
 [  1.   0.   0.   1.   1.   0.   0.   1.   0.   1.   0.   0.   1.  10.
    0.   1.   1.   0.   1.   0.   0.   0.]
 [  1.   0.   1.   0.   1.   0.   0.   1.   0.   0.   1.   0.   1.  75.
    0.   1.   0.   1.   0.   0.   0.   1.]]
Train Set
[[  0.   1.   1.   0.   1.   0.   0.   1.   0.   0.   1.   0.   1. 100.
    1.   0.   0.   1.   0.   1.   0.   0.]
 [  0.   1.   1.   0.   0.   1.   0.   1.   0.   0.   1.   1.   0.  10.
    1.   0.   1.   0.   0.   0.   0.   1.]
 [  1.   0.   0.   1.   1.   0.   0.   1.   0.   1.   0.   0.   1.  10.
    1.   0.   1.   0.   1.   0.   0.   0.]
 [  0.   1.   1.   0.   0.   0.   1.   0.   1.   1.   0.   1.   0. 100.
    1.   0.   0.   1.   0.   1.   0.   0.]
 [  1.   0.   0.   1.   1.   0.   0.   1.   0.   0.   1.   0.   1.  75.
    0.   1.   0.   1.   0.   0.   1.   0.]
 [  1.   0.   0.   1.   1.   0.   0.   1.   0.   1.   0.   0.   1.  10.
    0.   1.   1.   0.   1.   0.   0.   0.]]
test set
[[ 1.  0.  1.  0.  1.  0.  0.  1.  0.  0.  1.  0.  1. 75.  0.  1.  0.  1.
   0.  0.  0.  1.]]
test data
['Yes']
```