PRACTICAL 8

Aim: To implement Adaboost learning algorithm.

Code:

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction import DictVectorizer
from sklearn.ensemble import AdaBoostClassifier

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])

scaler=StandardScaler()
scaler.fit(X_Test)

X_Train=scaler.transform(X_Train)
X_Test=scaler.transform(X_Test)

model=AdaBoostClassifier(random_state=1)
model.fit(X_Train, y_Train)
print(le.inverse_transform(model.predict(X_Test)))
```

Output:

```
[Running] python -u "c:\Users\athar\Documents\Practicals\AI Practical\P8\Adaboost.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.]]
['No']
```