INTRODUCTION TO MACHINE LEARNING

LAB ASSIGNMENT – 4

NAME : PRATHAPANI SATWIKA

REG.NO.:20BCD7160

DECISION TREE CLASSIFICATION :

CODE :

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import AdaBoostClassifier

from sklearn.metrics import classification_report, confusion_matrix
company=pd.read_csv("Carseats.csv")
company.head()
company.dtypes
company['High'] = company.Sales.map(lambda x: 1 if x>8 else 0)
company['ShelveLoc']=company['ShelveLoc'].astype('category')
company['Urban']=company['Urban'].astype('category')
company['US']=company['US'].astype('category')
company.dtypes
company.head()
company['ShelveLoc']=company['ShelveLoc'].cat.codes
company['Urban']=company['Urban'].cat.codes
company['US']=company['US'].cat.codes
company.tail()
feature_cols=['CompPrice','Income','Advertising','Population','Price','She
lveLoc','Age','Education','Urban','US']
x = company[feature_cols]
y = company.High
print(x)
print(y)
x_train,x_test,y_train,y_test= train_test_split(x,y, test_size=0.2,random_
state=0)
dcmodel =  BaggingClassifier(DecisionTreeClassifier(max_depth = 6), random
_state=0)
dcmodel =  AdaBoostClassifier(DecisionTreeClassifier(max_depth = 6), rando
m_state=0)
```

```
dcmodel = dcmodel.fit(x_train,y_train)
y_predict = dcmodel.predict(x_test)
print("Accuracy : ", accuracy_score(y_test,y_predict)*100 )
print(confusion_matrix(y_test,y_predict))
print(classification_report(y_test,y_predict))
```

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import AdaBoostClassifier

from sklearn.metrics import classification_report, confusion_matrix
company=pd.read_csv("Carseats.csv")
company.head()
company.dtypes
company['High'] = company.Sales.map(lambda x: 1 if x>8 else 0)
company['ShelveLoc']=company['ShelveLoc'].astype('category')
company['Urban']=company['Urban'].astype('category')
company['US']=company['US'].astype('category')
company.dtypes
company.head()
company['ShelveLoc']=company['ShelveLoc'].cat.codes
company['Urban']=company['Urban'].cat.codes
company['US']=company['US'].cat.codes
company.tail()
feature_cols=['CompPrice','Income','Advertising','Population','Price','ShelveLoc','Age','Education','Urban','US']
x = company[feature_cols]
y = company.High
print(x)
print(y)
x_train,x_test,y_train,y_test= train_test_split(x,y, test_size=0.2,random_state=0)
dcmodel =  BaggingClassifier(DecisionTreeClassifier(max_depth = 6), random_state=0)
dcmodel =  AdaBoostClassifier(DecisionTreeClassifier(max_depth = 6), random_state=0)

dcmodel = dcmodel.fit(x_train,y_train)
y_predict = dcmodel.predict(x_test)
print("Accuracy : ", accuracy_score(y_test,y_predict)*100 )
print(confusion_matrix(y_test,y_predict))
print(classification_report(y_test,y_predict))
```

OUTPUT :

```
     CompPrice  Income  Advertising  Population  Price  ShelveLoc  Age  \
0          138      73           11         276    120          0   42
1          111      48           16         260     83          1   65
2          113      35           10         269     80          2   59
3          117     100            4         466     97          2   55
4          141      64            3         340    128          0   38
..         ...     ...          ...         ...    ...        ...  ...
395        138     108           17         203    128          1   33
396        139      23            3          37    120          2   55
397        162      26           12         368    159          2   40
398        100      79            7         284     95          0   50
399        134      37            0          27    120          1   49

     Education  Urban  US
0           17      1   1
1           10      1   1
2           12      1   1
3           14      1   1
4           13      1   0
..         ...    ...  ..
395         14      1   1
396         11      0   1
397         18      1   1
398         12      1   1
399         16      1   1

[400 rows x 10 columns]
0      1
1      1
2      1
3      0
4      0
      ..
395    1
396    0
397    0
398    0
399    1
Name: High, Length: 400, dtype: int64
Accuracy :  72.5
[[34  9]
 [13 24]]
              precision    recall  f1-score   support

           0       0.72      0.79      0.76        43
           1       0.73      0.65      0.69        37

    accuracy                           0.73        80
   macro avg       0.73      0.72      0.72        80
weighted avg       0.73      0.72      0.72        80
```

REGRESSION :

CODE :

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
dataset = pd.read_csv('Company_data.csv')
X = dataset['Temperature'].values
y = dataset['Revenue'].values
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.05
)

# Fitting Decision Tree Regression to the dataset
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
regressor.fit(X_train.reshape(-1,1), y_train.reshape(-1,1))
y_pred = regressor.predict(X_test.reshape(-1,1))
y_pred
df = pd.DataFrame({'Real Values':y_test.reshape(-
1), 'Predicted Values':y_pred.reshape(-1)})
df
# Visualising the Decision Tree Regression Results
X_grid = np.arange(min(X), max(X), 0.01)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X_test, y_test, color = 'red')
plt.scatter(X_test, y_pred, color = 'green')
plt.title('Decision Tree Regression')
plt.xlabel('Temperature')
plt.ylabel('Revenue')
plt.show()

plt.plot(X_grid, regressor.predict(X_grid), color = 'black')
plt.title('Decision Tree Regression')
plt.xlabel('Temperature')
plt.ylabel('Revenue')
plt.show()
```

```
[ ] import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd

    dataset = pd.read_csv('Company_data.csv')
    X = dataset['Temperature'].values
    y = dataset['Revenue'].values
```

```
[ ] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.05)

    # Fitting Decision Tree Regression to the dataset
    from sklearn.tree import DecisionTreeRegressor
    regressor = DecisionTreeRegressor()
    regressor.fit(X_train.reshape(-1,1), y_train.reshape(-1,1))
```

```
DecisionTreeRegressor()
```

```
y_pred = regressor.predict(X_test.reshape(-1,1))
y_pred
```

```
array([800.2024937, 625.1901215, 612.1539491, 546.6938576, 774.1080813,
       145.6253019, 118.8121496, 641.0253891, 807.5412872, 696.7166402,
       665.6726764, 586.150568 , 534.7990284, 321.7500343, 402.3984607,
       646.2669458, 454.1892673, 246.7871609, 653.9867356, 827.6848313,
       682.7528689, 493.1154676, 594.8048712, 905.4776043, 537.6648006])
```

```
[ ] df = pd.DataFrame({'Real Values':y_test.reshape(-1), 'Predicted Values':y_pred.reshape(-1)})
    df
```

```
# Visualising the Decision Tree Regression Results
X_grid = np.arange(min(X), max(X), 0.01)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X_test, y_test, color = 'red')
plt.scatter(X_test, y_pred, color = 'green')
plt.title('Decision Tree Regression')
plt.xlabel('Temperature')
plt.ylabel('Revenue')
plt.show()

plt.plot(X_grid, regressor.predict(X_grid), color = 'black')
plt.title('Decision Tree Regression')
plt.xlabel('Temperature')
plt.ylabel('Revenue')
plt.show()
```

OUTPUT :

	Real Values	Predicted Values
0	828.296077	800.202494
1	570.577875	625.190122
2	625.846421	612.153949
3	524.236115	546.693858
4	750.444733	774.108081
5	242.509855	145.625302
6	131.657017	118.812150
7	662.558990	641.025389
8	809.672053	807.541287
9	706.364904	696.716640
10	628.453211	665.672676
11	588.527551	586.150568
12	531.742485	534.799028
13	219.303993	321.750034
14	467.446707	402.398461
15	574.423310	646.266946
16	473.604335	454.189267
17	242.236208	246.787161
18	654.129377	653.986736
19	824.954357	827.684831
20	618.235765	682.752869
21	489.569090	493.115468
22	540.977511	594.804871
23	898.805423	905.477604
24	545.903929	537.664801