▾ Name : Vasu Kalariya

Roll : PE29

IMLA Lab Assi 5

```python
import numpy as np
import pandas as pd
dataset = pd.read_csv("dt1.csv")
dataset.head()
```

	age	income	Student	credit_rating	buy_comp
0	<30	high	no	fair	no
1	<30	high	no	excellent	no
2	30-40	high	no	fair	yes
3	>40	medium	no	fair	yes
4	>40	low	yes	fair	yes

```python
X = dataset.iloc[:,:-1]
Y = dataset.iloc[:,4:5].values
```

```python
from sklearn.preprocessing import LabelEncoder
labelencoder_X = LabelEncoder()
X = X.apply(LabelEncoder().fit_transform)
print(X)
```

```
    age  income  Student  credit_rating
0     1       0        0              1
1     1       0        0              0
2     0       0        0              1
3     2       2        0              1
4     2       1        1              1
5     2       1        1              0
6     0       1        1              0
7     1       2        0              1
8     1       1        1              1
9     2       2        1              1
10    1       2        1              0
11    0       2        0              0
12    0       0        1              1
13    2       2        0              0
```

```python
from sklearn.tree import DecisionTreeClassifier
```

```
classifier = DecisionTreeClassifier()
classifier = classifier.fit(X, Y)

X_in = np.array([1,1,0,0])
pred_y = classifier.predict([X_in])
print("Prediction: ", pred_y)
```

```
    Prediction:  ['no']
```

```
labelencoder_Y = LabelEncoder()
Y = labelencoder_Y.fit_transform(Y)
print(Y)
```

```
    [0 0 1 1 1 0 1 0 1 1 1 1 1 0]
    /usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_label.py:251: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_
      y = column_or_1d(y, warn=True)
```

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import cross_val_score

regressor = DecisionTreeRegressor(random_state=0)
regressor.fit(X,Y)
```

```
    DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=None,
                          max_features=None, max_leaf_nodes=None,
                          min_impurity_decrease=0.0, min_impurity_split=None,
                          min_samples_leaf=1, min_samples_split=2,
                          min_weight_fraction_leaf=0.0, presort='deprecated',
                          random_state=0, splitter='best')
```

```
y_pred = regressor.predict([X_in])
print("Prediction: ", y_pred)
```

```
    Prediction:  [0.]
```

✓ 0s   completed at 8:30 AM