Pgm 4

```
import numpy as np
import pandas as pd

data = pd.read_csv('tennis.csv')

def entropy(target):
    val,counts = np.unique(target,return_counts = True)
    ent = 0
    for i in range(len(val)):
        c = counts[i]/sum(counts)
        ent += -c*np.log2(c)
    return ent

def infoGain(data,features,target):
    te = entropy(data[target])
    val,counts = np.unique(data[features],return_counts = True)
    eg = 0
    for i in range(len(val)):
        c = counts[i]/sum(counts)
        eg += c*entropy(data[data[features] == val[i]][target])
    InfoGain = te-eg
    return InfoGain

def id3(data, features, target, pnode):

    t = np.unique(data[target])

    if len(t) == 1:
        return t[0]

    if len(features) == 0:
        return pnode

    pnode = t[np.argmax(t[1])]

    IG = [infoGain(data,f,target) for f in features]
    index = np.argmax(IG)

    col = features[index]
    tree = {col:{}}

    features = [f for f in features if f!=col]

    for val in np.unique(data[col]):
        sub_data = data[data[col]==val].dropna()
        subtree = id3(sub_data,features,target,pnode)
        tree[col][val] = subtree
    return tree

testData = data.sample(frac = 0.1)
data.drop(testData.index,inplace = True)

target = 'PlayTennis'
features = data.columns[data.columns!=target]

tree = id3(data,features,target,None)
print (tree, end='\n\n')
test = testData.to_dict('records')[0]
print(test, '=>', id3(test,features,target,None))
```

Pgm 6

```
import pandas as pd
from sklearn import tree
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB

# Load Data from CSV
data = pd.read_csv('tennis2.csv')
print("The first 5 Values of data is :\n", data.head())
X = data.iloc[:, :-1]
print("\nThe First 5 values of the train data is\n", X.head())
y = data.iloc[:, -1]
print("\nThe First 5 values of train output is\n", y.head())
le_outlook = LabelEncoder()
X.Outlook = le_outlook.fit_transform(X.Outlook)

le_Temperature = LabelEncoder()
X.Temperature = le_Temperature.fit_transform(X.Temperature)

le_Humidity = LabelEncoder()
X.Humidity = le_Humidity.fit_transform(X.Humidity)

le_Windy = LabelEncoder()
X.Windy = le_Windy.fit_transform(X.Windy)

print("\nNow the Train output is\n", X.head())

le_PlayTennis = LabelEncoder()
y = le_PlayTennis.fit_transform(y)
print("\nNow the Train output is\n",y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.20)

classifier = GaussianNB()
classifier.fit(X_train, y_train)

from sklearn.metrics import accuracy_score
print("Accuracy is:", accuracy_score(classifier.predict(X_test), y_test))
```