CODE :

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.externals.six import StringIO
import pydotplus as pdd
from IPython.display import Image
from sklearn.tree import export_graphviz

data=pd.read_csv("sales.csv")
data
data.describe()

print(data['Buys'].value_counts())
le=LabelEncoder();
#data=data.apply(le.fit_transform)
x=data.iloc[:,:-1] #-1 means don't take last column
x=x.apply(le.fit_transform)
#Store labels in Y
y=data.iloc[:,-1]
classifier=DecisionTreeClassifier(criterion='entropy')
classifier.fit(x,y)
#Predict value for the given Expression
#[Age < 21, Income = Low,Gender = Female, Marital Status =
Married]
test_x=np.array([1,1,0,0])
pred_y=classifier.predict([test_x])
print("Predicted class for input [Age < 21, Income =
Low,Gender = Female, Marital Status = Married]\n", test_x," is
",pred_y[0])
#method to generate graph
dot_dat=export_graphviz(classifier,out_file=None,feature_names
=x.columns,class_names=["No","Yes"])
graph=pdd.graph_from_dot_data(dot_dat)
graph.write_png("tree.png")
Image(graph.create_jpg())
```

Input for the code –

```
#sales.csv

ID,Age,Income,Gender,MaritialStatus,Buys
1,<21,High,Male,Single,No
2,<21,High,Male,Married,No
3,21-35,High,Male,Single,Yes
4,>35,Medium,Male,Single,Yes
5,>35,Low,Female,Single,Yes
6,>35,Low,Female,Married,No
7,21-35,Low,Female,Married,Yes
8,<21,Medium,Male,Single,No
```

9,<21,Low,Female,Married,Yes
10,>35,Medium,Female,Single,Yes
11,<21,Medium,Female,Married,Yes
12,21-35,Medium,Male,Married,Yes
13,21-35,High,Female,Single,Yes
14,>35,Medium,Male,Married,No

OUTPUT :

Yes 9
No 5

Name: Buys, dtype: int64

Predicted class for input [Age < 21, Income = Low,Gender = Female, Marital Status = Married]

[1100] is Yes