In [2]:

```
import pandas as pd
import numpy as np

df=pd.read_csv('DTC.csv')
df
df_up = df
```

In [3]:

```
x=df.iloc[:,1:-1]
```

In [4]:

```
x
```

Out[4]:

	Age	Income	Gender	Marital Status
0	<21	High	Male	Single
1	<21	High	Male	Married
2	21-35	High	Male	Single
3	>35	Medium	Male	Single
4	>35	Low	Female	Single
5	>35	Low	Female	Married
6	21-35	Low	Female	Married
7	<21	Medium	Male	Single
8	<21	Low	Female	Married
9	>35	Medium	Female	Single
10	<21	Medium	Female	Married
11	21-35	Medium	Male	Married
12	21-35	High	Female	Single
13	>35	Medium	Male	Married

In [5]:

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

x = x.apply(le.fit_transform)
x
```

Out[5]:

	Age	Income	Gender	Marital Status
0	1	0	1	1
1	1	0	1	0
2	0	0	1	1
3	2	2	1	1
4	2	1	0	1
5	2	1	0	0
6	0	1	0	0
7	1	2	1	1

	Age	Income	Gender	Marital Status
8	1	1	0	0
9	2	2	0	1
10	1	2	0	0
11	0	2	1	0
12	0	0	0	1
13	2	2	1	0

In [6]:

```
x['Gender'].values.reshape(-1,1)
```

Out[6]:

```
array([[1],
       [1],
       [1],
       [1],
       [0],
       [0],
       [0],
       [1],
       [0],
       [0],
       [0],
       [1],
       [0],
       [1]])
```

In [7]:

```
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder()

X_age = ohe.fit_transform(x['Age'].values.reshape(-1,1)).toarray().astype(int)
X_gen = ohe.fit_transform(x['Gender'].values.reshape(-1,1)).toarray().astype(int)
X_ms = ohe.fit_transform(x['Marital Status'].values.reshape(-1,1)).toarray().astype(int)

dfage = pd.DataFrame(X_age,columns=['Age<21','Age_btwn21-35','Age>35'])
df = pd.concat([x,dfage],axis=1)
df = df.drop(['Age'],axis=1)

dfOneHot = pd.DataFrame(X_gen, columns=['Female','Male'])
df = pd.concat([df, dfOneHot], axis=1)
df = df.drop(['Gender'], axis=1)

dfoh = pd.DataFrame(X_ms,columns=['Married','Single'])
df = pd.concat([df, dfoh], axis=1)
df = df.drop(['Marital Status'], axis=1)
df
```

Out[7]:

	Income	Age<21	Age_btwn21-35	Age>35	Female	Male	Married	Single
0	0	0	1	0	0	1	0	1
1	0	0	1	0	0	1	1	0
2	0	1	0	0	0	1	0	1
3	2	0	0	1	0	1	0	1
4	1	0	0	1	1	0	0	1
5	1	0	0	1	1	0	1	0
6	1	1	0	0	1	0	1	0
7	2	0	1	0	0	1	0	1
8	1	0	1	0	1	0	1	0

	Income	Age<21	Age_btwn21-35	Age>35	Female	Male	Married	Single
9	2	0	0	1	1	0	0	1
10	2	0	1	0	1	0	1	0
11	2	1	0	0	0	1	1	0
12	0	1	0	0	1	0	0	1
13	2	0	0	1	0	1	1	0

In [8]:

```
target = df_up.iloc[:,5]
```

In [9]:

```
target
```

Out[9]:

```
0      No
1      No
2     Yes
3     Yes
4     Yes
5      No
6     Yes
7      No
8     Yes
9     Yes
10    Yes
11    Yes
12    Yes
13     No
Name: Buys, dtype: object
```

In [10]:

```
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier(criterion="entropy")
dtc.fit(df,target)
y_pred = dtc.predict(df)
```

In [11]:

```
y_pred
```

Out[11]:

```
array(['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes',
       'Yes', 'Yes', 'Yes', 'No'], dtype=object)
```

In [12]:

```
# [Income = Low,Age < 21,Gender = Female, Marital Status = Married]
test_x=np.array([1,1,0,0,1,0,1,0])
pred_y=dtc.predict([test_x])
```

In [13]:

```
pred_y
```

Out[13]:

```
array(['Yes'], dtype=object)
```

In [14]:

```
from sklearn.metrics import confusion_matrix

confusion_matrix(target, y_pred)
```

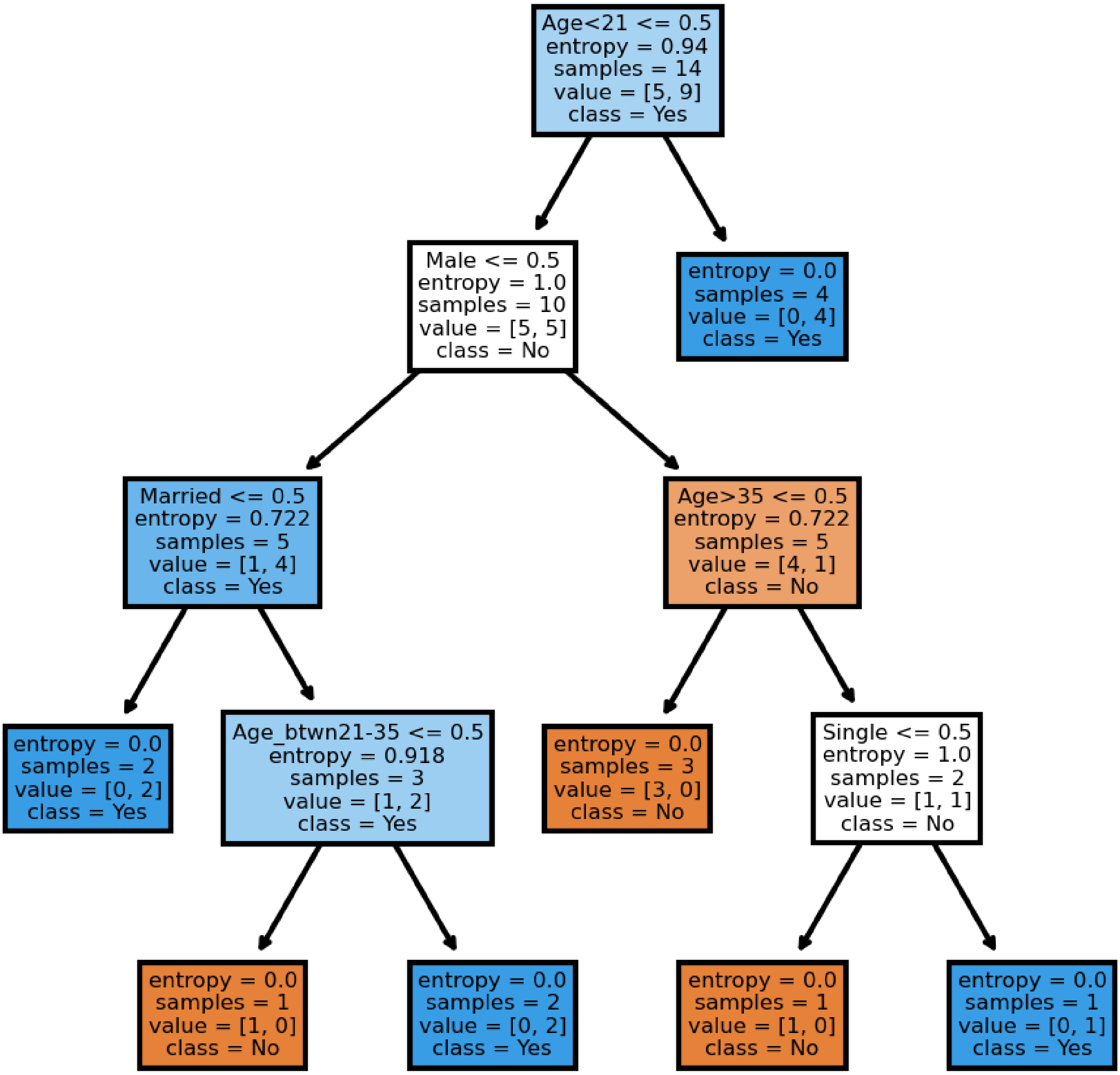
Out[14]:

```
array([[5, 0],
       [0, 9]])
```

In [15]:

```
from sklearn import tree
import matplotlib.pyplot as plt

fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (4,4), dpi=300)
tree.plot_tree(dtc, feature_names = df.columns, class_names=['No','Yes'], filled = True)
plt.show()
```



In [ ]: