In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import  DecisionTreeClassifier
from sklearn import tree
from sklearn.metrics import classification_report
```

In [3]:
```
df=pd.read_csv("/users\SAURABH/Saurabh patil/DATA SCIENCE/DEsicion TRee/Company_Data.csv")
```

In [4]:
```
df
```

Out[4]:

	Sales	CompPrice	Income	Advertising	Population	Price	ShelveLoc	Age	Education	Urban	US
0	9.50	138	73	11	276	120	Bad	42	17	Yes	Yes
1	11.22	111	48	16	260	83	Good	65	10	Yes	Yes
2	10.06	113	35	10	269	80	Medium	59	12	Yes	Yes
3	7.40	117	100	4	466	97	Medium	55	14	Yes	Yes
4	4.15	141	64	3	340	128	Bad	38	13	Yes	No
...	...	...	...	...	...	...	...	...	...	...	...
395	12.57	138	108	17	203	128	Good	33	14	Yes	Yes
396	6.14	139	23	3	37	120	Medium	55	11	No	Yes
397	7.41	162	26	12	368	159	Medium	40	18	Yes	Yes
398	5.94	100	79	7	284	95	Bad	50	12	Yes	Yes
399	9.71	134	37	0	27	120	Good	49	16	Yes	Yes

400 rows × 11 columns

In [6]:
```
df.Sales.describe()
```

Out[6]:
```
count    400.000000
mean       7.496325
std        2.824115
min        0.000000
25%        5.390000
50%        7.490000
75%        9.320000
max       16.270000
Name: Sales, dtype: float64
```

In [7]:
```
#Changing the categorical variables into dummies.
df1 = pd.get_dummies(df)

#Converting the Target variable i.e. Sales into Categorical
df1['Category'] = pd.cut(df1['Sales'],
                        bins=[0,10, np.inf],
                        labels=['Low/Mid','High'],
                        include_lowest=True)
```

In [8]:
```
df1
```

Out[8]:

	Sales	CompPrice	Income	Advertising	Population	Price	Age	Education	ShelveLoc_Bad	ShelveLoc_Good	ShelveLoc_Medium	Urban_No	Urban_Yes	US_No	US_Yes	Category
0	9.50	138	73	11	276	120	42	17	1	0	0	0	1	0	1	Low/Mid
1	11.22	111	48	16	260	83	65	10	0	1	0	0	1	0	1	High
2	10.06	113	35	10	269	80	59	12	0	0	1	0	1	0	1	High
3	7.40	117	100	4	466	97	55	14	0	0	1	0	1	0	1	Low/Mid
4	4.15	141	64	3	340	128	38	13	1	0	0	0	1	1	0	Low/Mid
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	12.57	138	108	17	203	128	33	14	0	1	0	0	1	0	1	High
396	6.14	139	23	3	37	120	55	11	0	0	1	1	0	0	1	Low/Mid
397	7.41	162	26	12	368	159	40	18	0	0	1	0	1	0	1	Low/Mid
398	5.94	100	79	7	284	95	50	12	1	0	0	0	1	0	1	Low/Mid
399	9.71	134	37	0	27	120	49	16	0	1	0	0	1	0	1	Low/Mid

400 rows × 16 columns

In [9]:
```
x1 = df1.iloc[:,1:15]
y1 = df1.iloc[:,15]
```

In [10]:
```
x1
```

Out[10]:

	CompPrice	Income	Advertising	Population	Price	Age	Education	ShelveLoc_Bad	ShelveLoc_Good	ShelveLoc_Medium	Urban_No	Urban_Yes	US_No	US_Yes
0	138	73	11	276	120	42	17	1	0	0	0	1	0	1
1	111	48	16	260	83	65	10	0	1	0	0	1	0	1
2	113	35	10	269	80	59	12	0	0	1	0	1	0	1
3	117	100	4	466	97	55	14	0	0	1	0	1	0	1
4	141	64	3	340	128	38	13	1	0	0	0	1	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	138	108	17	203	128	33	14	0	1	0	0	1	0	1
396	139	23	3	37	120	55	11	0	0	1	1	0	0	1
397	162	26	12	368	159	40	18	0	0	1	0	1	0	1
398	100	79	7	284	95	50	12	1	0	0	0	1	0	1
399	134	37	0	27	120	49	16	0	1	0	0	1	0	1

400 rows × 14 columns

In [11]:
```
y1
```

Out[11]:
```
0      Low/Mid
1         High
2         High
3      Low/Mid
4      Low/Mid
        ...
395       High
396    Low/Mid
397    Low/Mid
398    Low/Mid
399    Low/Mid
Name: Category, Length: 400, dtype: category
Categories (2, object): ['Low/Mid' < 'High']
```

In [12]:
```
# Splitting data into training and testing data set
x_train, x_test,y_train,y_test = train_test_split(x1,y1, test_size=0.25,random_state=40)
```

Building Decision Tree Classifier using Entropy Criteria

Iteration-1: Max Depth = 2

In [13]:
```
model1 = DecisionTreeClassifier(criterion = 'entropy',max_depth=2)
model1.fit(x_train,y_train)
preds1 = model1.predict(x_test) # predicting on test data set

print('Model leaves:',model1.get_n_leaves(),'\n','\n',
      pd.Series(preds1).value_counts(),'\n','\n',
      'Model Accuracy is:',np.mean(preds1==y_test))
```

```
Model leaves: 4

 Low/Mid    81
High       19
dtype: int64

 Model Accuracy is: 0.82
```

Iteration-2: Max Depth = 3

In [14]:
```
model2 = DecisionTreeClassifier(criterion = 'entropy',max_depth=3)
model2.fit(x_train,y_train)
preds2 = model2.predict(x_test) # predicting on test data set

print('Model leaves:',model2.get_n_leaves(),'\n','\n',
      pd.Series(preds2).value_counts(),'\n','\n',
      'Model Accuracy is:',np.mean(preds2==y_test))
```

```
Model leaves: 7

 Low/Mid    81
High       19
dtype: int64

 Model Accuracy is: 0.76
```

Iteration-3: Max Depth = 4

In [15]:
```
model3 = DecisionTreeClassifier(criterion = 'entropy',max_depth=4)
model3.fit(x_train,y_train)
preds3 = model3.predict(x_test) # predicting on test data set

print('Model leaves:',model3.get_n_leaves(),'\n','\n',
      pd.Series(preds3).value_counts(),'\n','\n',
      'Model Accuracy is:',np.mean(preds3==y_test))
```

```
Model leaves: 12

 Low/Mid    81
High       19
dtype: int64

 Model Accuracy is: 0.8
```

Iteration-4: Max Depth = 5

In [16]:
```
model4 = DecisionTreeClassifier(criterion = 'entropy',max_depth=5)
model4.fit(x_train,y_train)
preds4 = model4.predict(x_test) # predicting on test data set

print('Model leaves:',model4.get_n_leaves(),'\n','\n',
      pd.Series(preds4).value_counts(),'\n','\n',
      'Model Accuracy is:',np.mean(preds4==y_test))
```

```
Model leaves: 18

 Low/Mid    78
High       22
dtype: int64

 Model Accuracy is: 0.81
```

Iteration-5: Max Depth = 6

In [17]:
```
model5 = DecisionTreeClassifier(criterion = 'entropy',max_depth=6)
model5.fit(x_train,y_train)
preds5 = model5.predict(x_test) # predicting on test data set

print('Model leaves:',model5.get_n_leaves(),'\n','\n',
      pd.Series(preds5).value_counts(),'\n','\n',
      'Model Accuracy is:',np.mean(preds5==y_test))
```

```
Model leaves: 24

 Low/Mid    83
High       17
dtype: int64

 Model Accuracy is: 0.84
```

Iteration-6: Max Depth = 7

In [18]:
```
model6 = DecisionTreeClassifier(criterion = 'entropy',max_depth=7)
model6.fit(x_train,y_train)
preds6 = model6.predict(x_test) # predicting on test data set

print('Model leaves:',model6.get_n_leaves(),'\n','\n',
      pd.Series(preds6).value_counts(),'\n','\n',
      'Model Accuracy is:',np.mean(preds6==y_test))
```

```
Model leaves: 28

 Low/Mid    83
High       17
dtype: int64

 Model Accuracy is: 0.82
```

Hence, the classifier model at the end of iteration 5 has the max accuracy i.e. 83%

In [19]:
```
print(classification_report(preds5,y_test))
```

```
              precision    recall  f1-score   support

        High       0.53      0.59      0.56        17
     Low/Mid       0.91      0.89      0.90        83

    accuracy                           0.84       100
   macro avg       0.72      0.74      0.73       100
weighted avg       0.85      0.84      0.84       100
```

Building Decision Tree Classifier (CART) using Gini Criteria

In [20]:
```
from sklearn.tree import DecisionTreeClassifier
model_gini = DecisionTreeClassifier(criterion='gini', max_depth=6)
```

In [21]:
```
model_gini.fit(x_train, y_train)
```

Out[21]:
```
DecisionTreeClassifier(max_depth=6)
```

In [22]:
```
#Prediction and computing the accuracy
predG=model_gini.predict(x_test)
print('Model Accuracy is:',np.mean(predG==y_test))
```

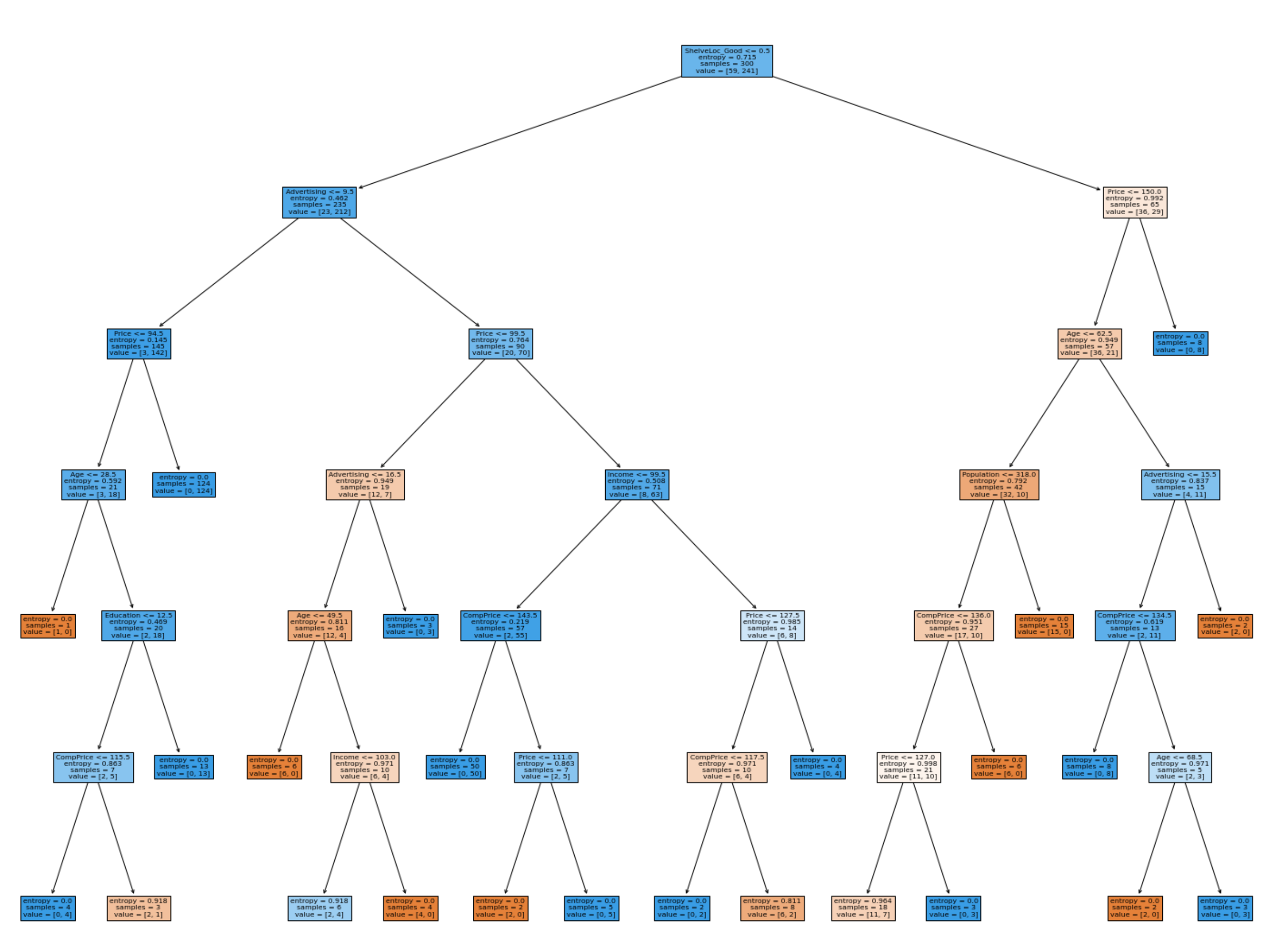
```
Model Accuracy is: 0.83
```

Same Accuracy is achieved using CART as well

Let's Visualize both the Decision Trees

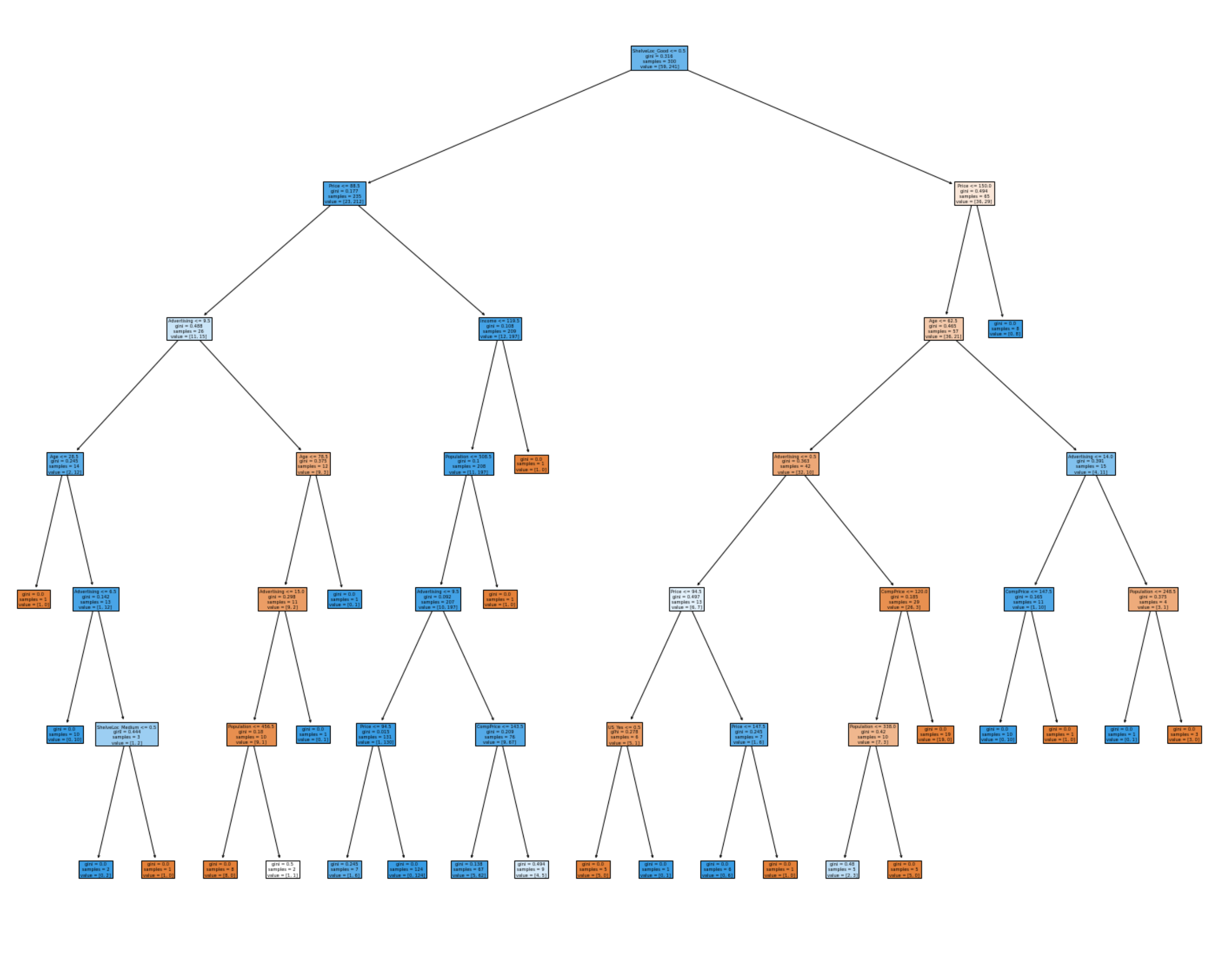
In [23]:
```
fig = plt.figure(figsize=(25,20))
fig = tree.plot_tree(model5,
                    feature_names= ['CompPrice','Income','Advertising','Population','Price','Age','Education',
                    'ShelveLoc_Bad','ShelveLoc_Good','ShelveLoc_Medium','Urban_No','Urban_Yes','US_No',
                    'US_Yes'], filled=True)
plt.title('Decision tree using Entropy',fontsize=22)
plt.savefig('DT_Entropy.png')
```

Decision tree using Entropy



In [24]:
```
fig = plt.figure(figsize=(25,20))
fig = tree.plot_tree(model_gini,
                    feature_names= ['CompPrice','Income','Advertising','Population','Price','Age','Education',
                    'ShelveLoc_Bad','ShelveLoc_Good','ShelveLoc_Medium','Urban_No','Urban_Yes','US_No',
                    'US_Yes'], filled=True)
plt.title('Decision tree using CART',fontsize=22)
plt.savefig('DT_Cart.png')
```

Decision tree using CART



After combining the insights from both the plots, we can say that following are the 3 top factors affecting the sales:

1. Shelf Location at stores
2. Pricing
3. Advertising

In [ ]: