Import Necessary Libraries

In [162…
```
!pip install pydotplus
```

```
Requirement already satisfied: pydotplus in c:\users\91998\anaconda3\lib\site-packages (2.0.2)
Requirement already satisfied: pyparsing>=2.0.1 in c:\users\91998\anaconda3\lib\site-packages (from pydotplus) (2.4.7)
```

In [163…
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from io import StringIO
import pydotplus
import seaborn as sns
import matplotlib.image as mpimg
from sklearn.tree import export_graphviz
from sklearn.tree import DecisionTreeClassifier as DT
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn import tree
from IPython.display import Image
```

Data Collection

In [164…
```
com_data=pd.read_csv('Company_Data.csv',sep=',')
com_data
```

Out[164…

	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

Data Understanding

Initial Analysis

In [165…
```
com_data.shape
```

Out[165… (400, 11)

In [166…
```
com_data.isna().sum()
```

Out[166…
```
Sales          0
CompPrice      0
Income         0
Advertising    0
Population     0
Price          0
ShelveLoc      0
Age            0
Education      0
Urban          0
US             0
dtype: int64
```

In [167…
```
com_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Sales        400 non-null    float64
 1   CompPrice    400 non-null    int64
 2   Income       400 non-null    int64
 3   Advertising  400 non-null    int64
 4   Population   400 non-null    int64
 5   Price        400 non-null    int64
 6   ShelveLoc    400 non-null    object
 7   Age          400 non-null    int64
 8   Education    400 non-null    int64
 9   Urban        400 non-null    object
 10  US           400 non-null    object
dtypes: float64(1), int64(7), object(3)
memory usage: 34.5+ KB
```

In [168…
```
com_data.head()
```

Out[168…

	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

In [169…
```
com_data.describe().T
```

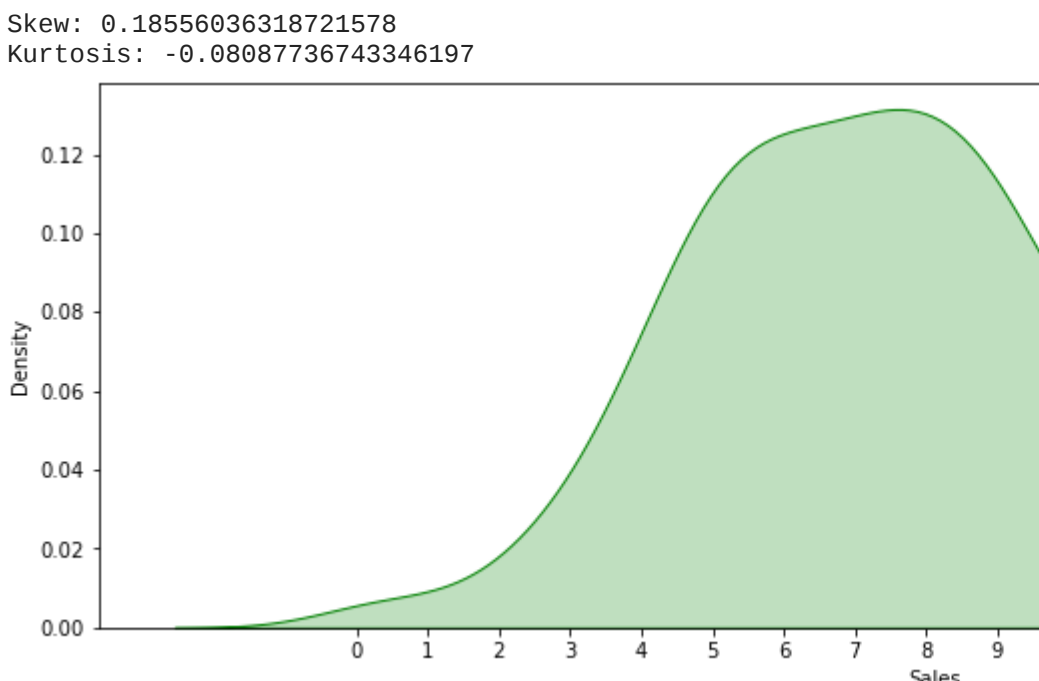
Out[169…

	count	mean	std	min	25%	50%	75%	max
Sales	400.0	7.496325	2.824115	0.0	5.39	7.49	9.32	16.27
CompPrice	400.0	124.975000	15.334512	77.0	115.00	125.00	135.00	175.00
Income	400.0	68.657500	27.986037	21.0	42.75	69.00	91.00	120.00
Advertising	400.0	6.635000	6.650364	0.0	0.00	5.00	12.00	29.00
Population	400.0	264.840000	147.376436	10.0	139.00	272.00	398.50	509.00
Price	400.0	115.795000	23.676664	24.0	100.00	117.00	131.00	191.00
Age	400.0	53.322500	16.200297	25.0	39.75	54.50	66.00	80.00
Education	400.0	13.900000	2.620528	10.0	12.00	14.00	16.00	18.00

Outliers Check

In [170…
```
import warnings
warnings.filterwarnings('ignore')
```

In [171…
```
data1=sns.boxplot(com_data['Sales'])
```

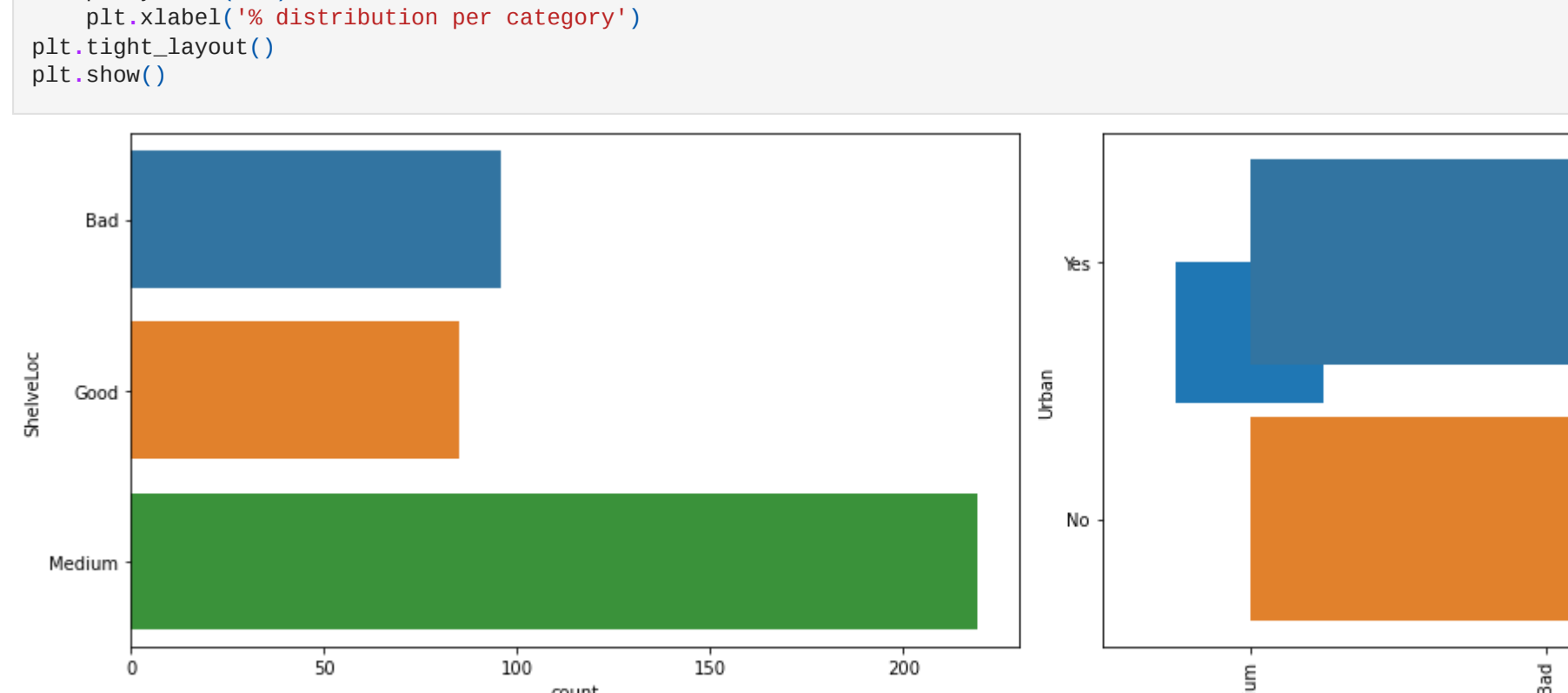


The data has 2 outliers

In [172…
```
plt.rcParams['figure.figsize']=9,4
```

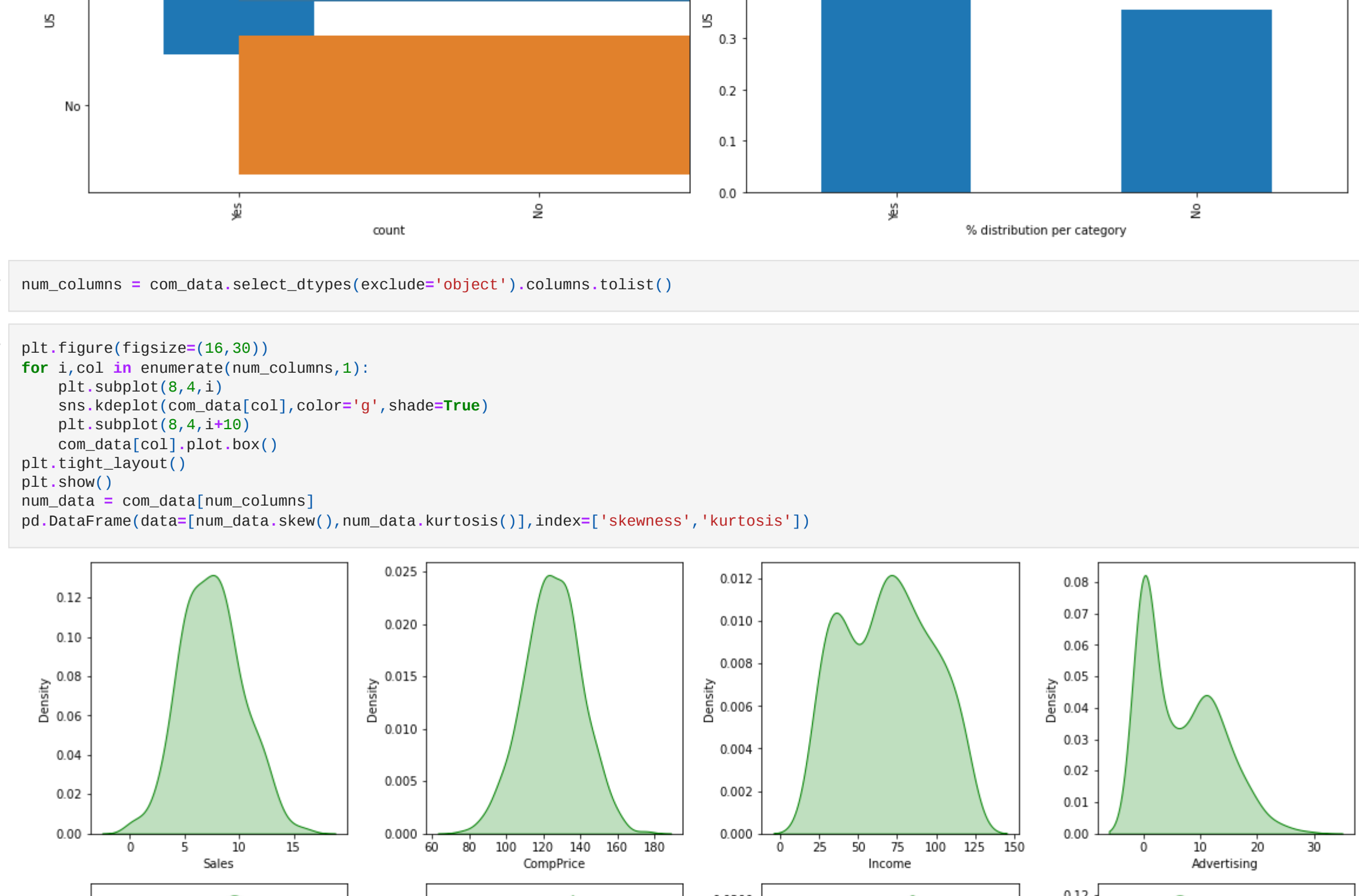
In [173…
```
plt.figure(figsize=(15,5))
print("Skew: {}".format(com_data['Sales'].skew()))
print("Kurtosis: {}".format(com_data['Sales'].kurtosis()))
data1 = sns.kdeplot(com_data['Sales'],shade=True,color='g')
plt.xticks([i for i in range(0,20,1)])
plt.show()
```

```
Skew: 0.18556036318721578
Kurtosis: -0.08087736743346197
```



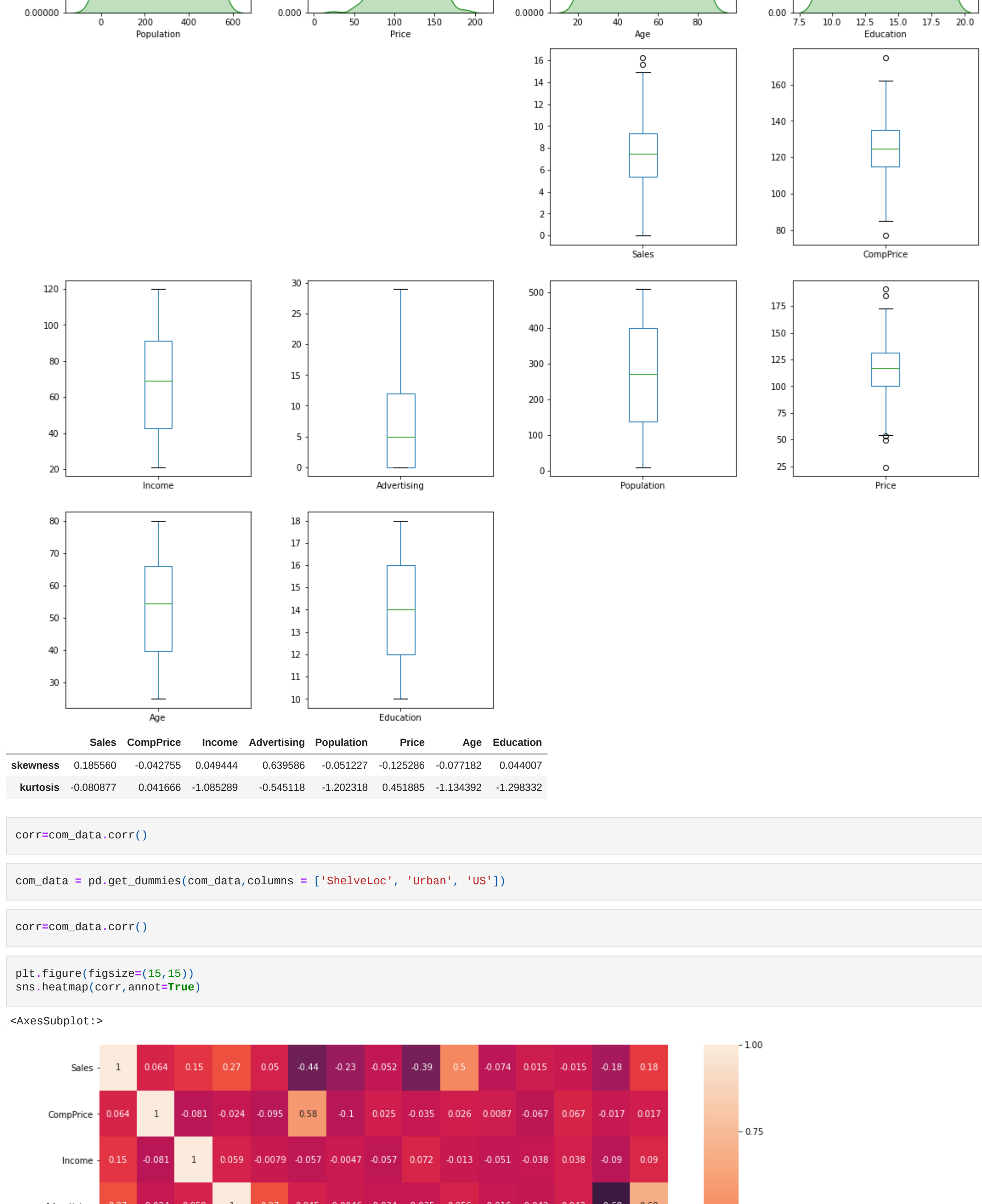
In [174…
```
obj_colum = com_data.select_dtypes(include='object').columns.tolist()
```

In [175…
```
plt.figure(figsize=(16,10))
for i,col in enumerate(obj_colum,1):
    plt.subplot(2,2,i)
    sns.countplot(data=com_data,y=col)
    plt.subplot(2,2,i+1)
    com_data[col].value_counts(normalize=True).plot.bar()
    plt.ylabel(col)
    plt.xlabel('% distribution per category')
plt.tight_layout()
plt.show()
```



In [176…
```
num_columns = com_data.select_dtypes(exclude='object').columns.tolist()
```

In [177…
```
plt.figure(figsize=(16,30))
for i,col in enumerate(num_columns,1):
    plt.subplot(8,4,i)
    sns.kdeplot(com_data[col],color='g',shade=True)
    plt.subplot(8,4,i+10)
    com_data[col].plot.box()
plt.tight_layout()
plt.show()
num_data = com_data[num_columns]
pd.DataFrame(data=[num_data.skew(),num_data.kurtosis()],index=['skewness','kurtosis'])
```



Out[177…

	Sales	CompPrice	Income	Advertising	Population	Price	Age	Education
skewness	0.185560	-0.042755	0.049444	0.639586	-0.051227	-0.125286	-0.077182	0.044007
kurtosis	-0.080877	0.041666	-1.085289	-0.545118	-1.202318	0.451885	-1.134392	-1.298332

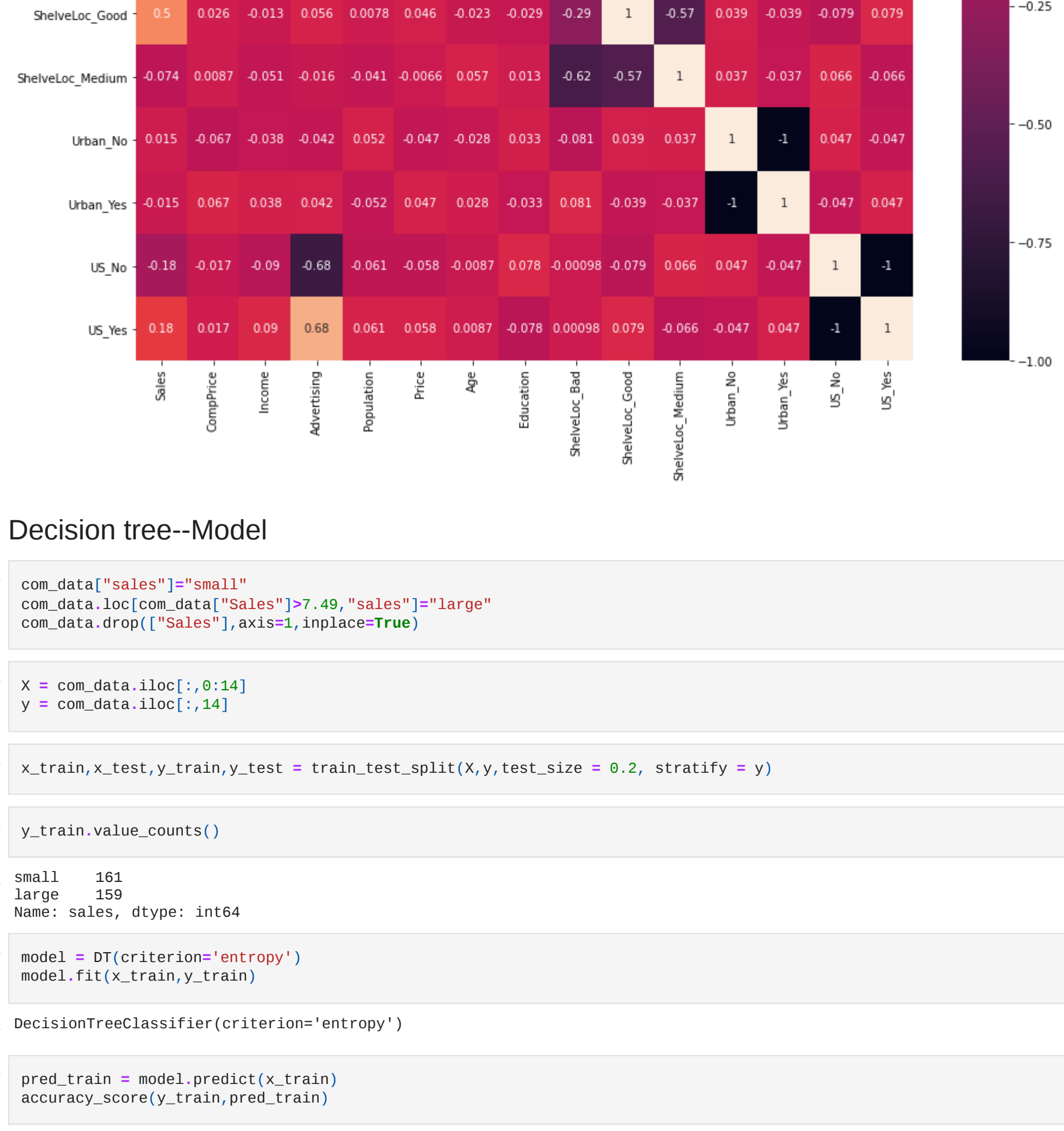
In [178…
```
corr=com_data.corr()
```

In [179…
```
com_data = pd.get_dummies(com_data,columns = ['ShelveLoc', 'Urban', 'US'])
```

In [180…
```
corr=com_data.corr()
```

In [181…
```
plt.figure(figsize=(15,15))
sns.heatmap(corr,annot=True)
```

Out[181… <AxesSubplot:>



Decision tree--Model

In [182…
```
com_data["sales"]="small"
com_data.loc[com_data["Sales"]>7.49,"sales"]="large"
com_data.drop(["Sales"],axis=1,inplace=True)
```

In [183…
```
X = com_data.iloc[:,0:14]
y = com_data.iloc[:,14]
```

In [184…
```
x_train,x_test,y_train,y_test = train_test_split(X,y,test_size = 0.2, stratify = y)
```

In [185…
```
y_train.value_counts()
```

Out[185…
```
small    161
large    159
Name: sales, dtype: int64
```

In [186…
```
model = DT(criterion='entropy')
model.fit(x_train,y_train)
```

Out[186… DecisionTreeClassifier(criterion='entropy')

In [187…
```
pred_train = model.predict(x_train)
accuracy_score(y_train,pred_train)
```

Out[187… 1.0

In [188…
```
confusion_matrix(y_train,pred_train)
```

Out[188…
```
array([[159,   0],
       [  0, 161]], dtype=int64)
```

In [189…
```
pred_test = model.predict(x_test)
```

In [190…
```
accuracy_score(y_test,pred_test)
```

Out[190… 0.775

In [191…
```
confusion_matrix(y_test,pred_test)
```

Out[191…
```
array([[32,  8],
       [10, 30]], dtype=int64)
```

In [192…
```
com_data_t=pd.DataFrame({'Actual':y_test, 'Predicted':pred_test})
```

In [193…
```
com_data_t
```

Out[193…

	Actual	Predicted
9	small	small
80	large	large
288	small	small
265	small	small
89	large	large
...	...	...
28	small	small
36	large	large
196	small	small
348	large	large
248	small	small

80 rows × 2 columns

In [194…
```
cols = list(com_data.columns)
```

In [195…
```
predictors = cols[0:14]
target = cols[14]
```

In [196…
```
dot_data = StringIO()
```

In [197…
```
export_graphviz(model,out_file = dot_data, filled =True, rounded = True, feature_names =predictors,class_names = target, impurity = False )
```

In [198…
```
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
```

In [199…
```
graph.write_png('company_full.png')
```

Out[199… True

In [200…
```
img = mpimg.imread('company_full.png')
```

In [201…
```
plt.imshow(img)
```

Out[201… <matplotlib.image.AxesImage at 0x2cefb3fe760>

In [202…
```
model.feature_importances_
```

Out[202…
```
array([0.15154678, 0.07011195, 0.14092177, 0.08652149, 0.3171172 ,
       0.09284615, 0.02048544, 0.        , 0.08658577, 0.01236959,
       0.        , 0.01288459, 0.00860927, 0.        ])
```

In [203…
```
fi = pd.DataFrame({'feature': list(x_train.columns),
                   'importance': model.feature_importances_}).\
                    sort_values('importance', ascending = False)
fi
```

Out[203…

	feature	importance
4	Price	0.317117
0	CompPrice	0.151547
2	Advertising	0.140922
5	Age	0.092846
8	ShelveLoc_Good	0.086586
3	Population	0.086521
1	Income	0.070112
6	Education	0.020485
11	Urban_Yes	0.012885
9	ShelveLoc_Medium	0.012370
12	US_No	0.008609
7	ShelveLoc_Bad	0.000000
10	Urban_No	0.000000
13	US_Yes	0.000000

In [ ]:

In [ ]:

In [ ]:

In [ ]: