```
from google.colab import files
upload=files.upload()
```

```
Choose Files Company_Data.csv
• Company_Data.csv(text/csv) - 17023 bytes, last modified: 2/25/2023 - 100% done
Saving Company_Data.csv to Company_Data.csv
```

```
import pandas as pd
import numpy as np
df=pd.read_csv("Company_Data.csv")
df.head()
```

	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

```
df.dtypes
```

```
Sales          float64
CompPrice        int64
Income           int64
Advertising      int64
Population       int64
Price            int64
ShelveLoc       object
Age              int64
Education        int64
Urban           object
US              object
dtype: object
```

```
mean=df["Sales"].mean()
for i in range(0,400):
  if df["Sales"][i] <= mean:
    df["Sales"][i]="No"
  else:
    df["Sales"][i]="Yes"
```

```
<ipython-input-82-56281154fe9b>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df["Sales"][i]="Yes"
/usr/local/lib/python3.8/dist-packages/pandas/core/indexing.py:1732: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  self._setitem_single_block(indexer, value, name)
```

```
df.dtypes
```

```
Sales          object
CompPrice       int64
Income          int64
Advertising     int64
Population      int64
Price           int64
ShelveLoc      object
Age             int64
Education       int64
Urban          object
US             object
dtype: object
```

```
df_cat=df.select_dtypes("object")
df_cat
```

```
df_con=df.select_dtypes("int")
df_con
```

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

400 rows × 7 columns

```
from sklearn.preprocessing import StandardScaler
SS = StandardScaler()
df_con_SS = SS.fit_transform(df.con)
df_con_SS=pd.DataFrame(df_con_SS)
df_con_SS
```

	0	1	2	3	4	5	6
0	0.850455	0.155361	0.657177	0.075819	0.177823	-0.699782	1.184449
1	-0.912484	-0.739060	1.409957	-0.032882	-1.386854	0.721723	-1.490113
2	-0.781896	-1.204159	0.506621	0.028262	-1.513719	0.350895	-0.725953
3	-0.520720	1.121336	-0.396715	1.366649	-0.794814	0.103677	0.038208
4	1.046337	-0.166631	-0.547271	0.510625	0.516132	-0.947000	-0.343872
...	...	...	...	...	...	...	...
395	0.850455	1.407551	1.560513	-0.420131	0.516132	-1.256023	0.038208
396	0.915749	-1.633482	-0.547271	-1.547909	0.177823	0.103677	-1.108033
397	2.417512	-1.526151	0.807733	0.700853	1.827078	-0.823391	1.566529
398	-1.630719	0.370022	0.054953	0.130170	-0.879391	-0.205346	-0.725953
399	0.589279	-1.132606	-0.998939	-1.615848	0.177823	-0.267150	0.802369

400 rows × 7 columns

```
from sklearn.preprocessing import LabelEncoder
LE = LabelEncoder()
for i in range(0,4):
  df_cat.iloc[:,i]=LE.fit_transform(df_cat.iloc[:,i])
df_cat
```

```
/usr/local/lib/python3.8/dist-packages/pandas/core/indexing.py:1773: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  self._setitem_single_column(ilocs[0], value, pi)
```

	Sales	ShelveLoc	Urban	US
0	1	0	1	1
1	1	1	1	1
2	1	2	1	1
3	0	2	1	1
4	0	0	1	0

```
df1 = pd.concat([df_cat,df_con_SS],axis=1)
df1.head()
```

	Sales	ShelveLoc	Urban	US	0	1	2	3	4	5	6
0	1	0	1	1	0.850455	0.155361	0.657177	0.075819	0.177823	-0.699782	1.184449
1	1	1	1	1	-0.912484	-0.739060	1.409957	-0.032882	-1.386854	0.721723	-1.490113
2	1	2	1	1	-0.781896	-1.204159	0.506621	0.028262	-1.513719	0.350895	-0.725953
3	0	2	1	1	-0.520720	1.121336	-0.396715	1.366649	-0.794814	0.103677	0.038208
4	0	0	1	0	1.046337	-0.166631	-0.547271	0.510625	0.516132	-0.947000	-0.343872

```
y=df1["Sales"]
x=df1.iloc[:,1:11]
x
```

	ShelveLoc	Urban	US	0	1	2	3	4	5	6
0	0	1	1	0.850455	0.155361	0.657177	0.075819	0.177823	-0.699782	1.184449
1	1	1	1	-0.912484	-0.739060	1.409957	-0.032882	-1.386854	0.721723	-1.490113
2	2	1	1	-0.781896	-1.204159	0.506621	0.028262	-1.513719	0.350895	-0.725953
3	2	1	1	-0.520720	1.121336	-0.396715	1.366649	-0.794814	0.103677	0.038208
4	0	1	0	1.046337	-0.166631	-0.547271	0.510625	0.516132	-0.947000	-0.343872
...	...	...	...	...	...	...	...	...	...	...
395	1	1	1	0.850455	1.407551	1.560513	-0.420131	0.516132	-1.256023	0.038208
396	2	0	1	0.915749	-1.633482	-0.547271	-1.547909	0.177823	0.103677	-1.108033
397	2	1	1	2.417512	-1.526151	0.807733	0.700853	1.827078	-0.823391	1.566529
398	0	1	1	-1.630719	0.370022	0.054953	0.130170	-0.879391	-0.205346	-0.725953
399	1	1	1	0.589279	-1.132606	-0.998939	-1.615848	0.177823	-0.267150	0.802369

400 rows × 10 columns

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(x,y,test_size=0.3)
```

```
from sklearn.tree import DecisionTreeRegressor
DT=DecisionTreeRegressor(max_depth=5)
DT.fit(X_train,Y_train)
Y_pred_train=DT.predict(X_train)
Y_pred_test=DT.predict(X_test)
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/utils/validation.py:1688: FutureWarning: Feature names only support names that are all st
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/utils/validation.py:1688: FutureWarning: Feature names only support names that are all st
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/utils/validation.py:1688: FutureWarning: Feature names only support names that are all st
  warnings.warn(
```

```
from sklearn.metrics import mean_squared_error
mse1 = mean_squared_error(Y_train,Y_pred_train)
RMSE1 = np.sqrt(mse1)
print("Training Error: ",RMSE1.round(2))

mse2 = mean_squared_error(Y_test,Y_pred_test)
RMSE2 = np.sqrt(mse2)
print("Test Error: ",RMSE2.round(2))
```

```
Training Error:  0.3
Test Error:  0.43
```

bagging

```
from sklearn.ensemble import BaggingRegressor
BG=BaggingRegressor(max_features=0.5)
BG.fit(X_train,Y_train)
Y_pred_train=BG.predict(X_train)
Y_pred_test=BG.predict(X_test)
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/utils/validation.py:1688: FutureWarning: Feature names only support names that are all st
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/utils/validation.py:1688: FutureWarning: Feature names only support names that are all st
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/utils/validation.py:1688: FutureWarning: Feature names only support names that are all st
  warnings.warn(
```

```
from sklearn.metrics import mean_squared_error
mse1 = mean_squared_error(Y_train,Y_pred_train)
RMSE1 = np.sqrt(mse1)
print("Training Error: ",RMSE1.round(2))

mse2 = mean_squared_error(Y_test,Y_pred_test)
RMSE2 = np.sqrt(mse2)
print("Test Error: ",RMSE2.round(2))
```

```
Training Error:  0.22
Test Error:  0.51
```