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

Choose Files Fraud_check.csv

- **Fraud_check.csv**(text/csv) - 21837 bytes, last modified: 2/25/2023 - 100% done

```
Saving Fraud_check.csv to Fraud_check.csv
```

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

df=pd.read_csv("Fraud_check.csv")
df.head()
```

	Undergrad	Marital.Status	Taxable.Income	City.Population	Work.Experience	Urban
0	NO	Single	68833	50047	10	YES
1	YES	Divorced	33700	134075	18	YES
2	NO	Married	36925	160205	30	YES
3	YES	Single	50190	193264	15	YES
4	NO	Married	81002	27533	28	NO

```
df.dtypes
```

```
Undergrad          object
Marital.Status     object
Taxable.Income      int64
City.Population     int64
Work.Experience     int64
Urban              object
dtype: object
```

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

	Undergrad	Marital.Status	Urban
0	NO	Single	YES
1	YES	Divorced	YES
2	NO	Married	YES
3	YES	Single	YES
4	NO	Married	NO
...	...	...	...
595	YES	Divorced	YES
596	YES	Divorced	YES
597	NO	Divorced	YES
598	YES	Married	NO
599	NO	Divorced	NO

600 rows × 3 columns

```
from sklearn.preprocessing import LabelEncoder
LE=LabelEncoder()
for i in range(0,3):
  df_cat.iloc[:,i]=LE.fit_transform(df_cat.iloc[:,i])
```

```
df_cat
```

	Undergrad	Marital.Status	Urban
0	0	2	1
1	1	0	1
2	0	1	1
3	1	2	1
4	0	1	0
...	...	...	...
595	1	0	1
596	1	0	1
597	0	0	1
598	1	1	0

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

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

	0	1	2	Undergrad	Marital.Status	Urban
0	0.520362	-1.178521	-0.629143	0	2	1
1	-0.821464	0.508500	0.276370	1	0	1
2	-0.698292	1.033109	1.634639	0	1	1
3	-0.191666	1.696831	-0.063197	1	2	1
4	0.985129	-1.630532	1.408261	0	1	0
...	...	...	...	...	...	...
595	0.807075	-1.390432	-0.968710	1	0	1
596	0.563672	-1.071672	-1.534655	1	0	1
597	-0.300744	0.909696	-1.761033	0	0	1
598	1.656940	1.432197	0.163181	1	1	0
599	1.577766	0.991590	0.049992	0	0	0

600 rows × 6 columns

```
x=df1.iloc[:,0:5]
y=df1["Urban"]
```

```
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:  0.44
Test Error:  0.56
```

################## 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:  0.33
Test Error:  0.53
```

✓ 0s completed at 9:33 PM