```
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
```

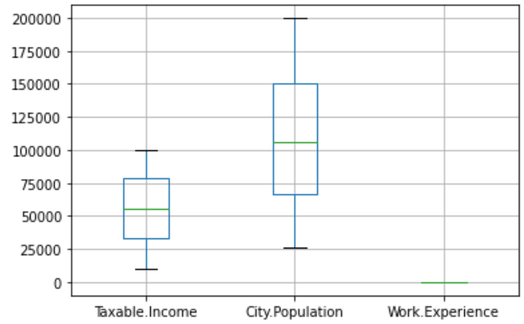
```
import pandas as pd
import numpy as np
```

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

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

```
df.shape
df.dtypes
df.isnull().sum()
df.boxplot(None)
```

```
<AxesSubplot:>
```



```
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])
```

```
    /usr/local/lib/python3.8/dist-packages/pandas/core/indexing.py:1951: 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.obj[selected_item_labels] = value
    /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)
```

```
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)
```

```
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.ensemble import RandomForestRegressor
RFC = RandomForestRegressor(max_depth=0.5,max_features=0.6)
RFC.fit(X_train,Y_train)
Y_pred_train=RFC.predict(X_train)
Y_pred_test=RFC.predict(X_test)
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:  0.5
    Test Error:  0.5
    /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(
```

✓ 0s completed at 11:15 PM ● ×