```
import pandas as pd
```

```
purchase = pd.read_csv('https://github.com/YBIFoundation/Dataset/raw/main/Customer%20Purchase
```

```
purchase.head()
```

	Customer ID	Age	Gender	Education	Review	Purchased
0	1021	30	Female	School	Average	No
1	1022	68	Female	UG	Poor	No
2	1023	70	Female	PG	Good	No
3	1024	72	Female	PG	Good	No
4	1025	16	Female	UG	Average	No

```
purchase.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Customer ID  50 non-null     int64 
 1   Age          50 non-null     int64 
 2   Gender       50 non-null     object
 3   Education    50 non-null     object
 4   Review       50 non-null     object
 5   Purchased    50 non-null     object
dtypes: int64(2), object(4)
memory usage: 2.5+ KB
```

```
purchase.describe()
```

	Customer ID	Age

```
purchase.columns
```

```
Index(['Customer ID', 'Age', 'Gender', 'Education', 'Review', 'Purchased'],
dtype='object')
```

```
y = purchase['Purchased']
```

```
X = purchase.drop(['Purchased','Customer ID'],axis=1)
```

75%	1057.75000	74.000000

```
X.replace({'Review':{'Poor':0,'Average':1,'Good':2}},inplace=True)
X.replace({'Education':{'School':0,'UG':1,'PG':2}},inplace=True)
X.replace({'Gender':{'Male': 0,'Female':1}},inplace=True)
```

```
X.head()
```

	Age	Gender	Education	Review
0	30	1	0	1
1	68	1	1	0
2	70	1	2	2
3	72	1	2	2
4	16	1	1	1

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, train_size=0.8, random_state=2529)
```

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((40, 4), (10, 4), (40,), (10,))
```

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
```

```
model.fit(X_train,y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
y_pred = model.predict(X_test)
```

```
y_pred
```

```
array(['No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'No', 'Yes'],
      dtype=object)
```

```
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
```

```
confusion_matrix(y_test,y_pred)
```

```
array([[2, 1],
       [3, 4]])
```

```
accuracy_score(y_test,y_pred)
```

```
0.6
```

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

          No       0.40      0.67      0.50         3
         Yes       0.80      0.57      0.67         7

    accuracy                           0.60        10
   macro avg       0.60      0.62      0.58        10
weighted avg       0.68      0.60      0.62        10
```

+ Code + Text

✓ 0s completed at 6:18 PM