Muhammad Umer Adeeb

Question 4: Kaggle Dataset

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('/content/Heart.csv')
df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 303,\n  \"fields\": [\
n    {\n      \"column\": \"Unnamed: 0\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 87,\n        \"min\": 1,\n
\"max\": 303,\n        \"num_unique_values\": 303,\n
\"samples\": [\n          180,\n          229,\n          112\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 9,\n        \"min\":
29,\n        \"max\": 77,\n        \"num_unique_values\": 41,\n
\"samples\": [\n          61,\n          64,\n          44\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Sex\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"ChestPain\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"asymptomatic\",\n          \"nontypical\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"RestBP\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 17,\n
\"min\": 94,\n        \"max\": 200,\n        \"num_unique_values\":
50,\n        \"samples\": [\n          124,\n          192\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Chol\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 51,\n        \"min\": 126,\n        \"max\": 564,\n
\"num_unique_values\": 152,\n        \"samples\": [\n          321,\n
187\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Fbs\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

```
\"RestECG\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 0,\n        \"min\": 0,\n        \"max\": 2,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          2,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"MaxHR\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 22,\n        \"min\": 71,\n        \"max\": 202,\n
\"num_unique_values\": 91,\n        \"samples\": [\n          170,\n
114\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"ExAng\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Oldpeak\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 1.161075022068634,\n        \"min\": 0.0,\n
\"max\": 6.2,\n        \"num_unique_values\": 40,\n
\"samples\": [\n          2.4,\n          0.2\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Slope\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 0,\n        \"min\":
1,\n        \"max\": 3,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          3,\n          2\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Ca\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.9374383177242157,\n
\"min\": 0.0,\n        \"max\": 3.0,\n        \"num_unique_values\":
4,\n        \"samples\": [\n          3.0,\n          1.0\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Thal\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
3,\n        \"samples\": [\n          \"fixed\",\n
\"normal\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"AHD\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          \"Yes\",\
n          \"No\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  303 non-null    int64
 1   Age         303 non-null    int64
```

```
 2   Sex          303 non-null    int64
 3   ChestPain    303 non-null    object
 4   RestBP       303 non-null    int64
 5   Chol         303 non-null    int64
 6   Fbs          303 non-null    int64
 7   RestECG      303 non-null    int64
 8   MaxHR        303 non-null    int64
 9   ExAng        303 non-null    int64
 10  Oldpeak      303 non-null    float64
 11  Slope        303 non-null    int64
 12  Ca           299 non-null    float64
 13  Thal         301 non-null    object
 14  AHD          303 non-null    object
dtypes: float64(2), int64(10), object(3)
memory usage: 35.6+ KB
```

```
df.shape
```

```
(303, 15)
```

```
df.isnull().sum()
```

```
Unnamed: 0    0
Age           0
Sex           0
ChestPain     0
RestBP        0
Chol          0
Fbs           0
RestECG       0
MaxHR         0
ExAng         0
Oldpeak       0
Slope         0
Ca            4
Thal          2
AHD           0
dtype: int64
```

```
df.dtypes
```

```
Unnamed: 0      int64
Age             int64
Sex             int64
ChestPain      object
RestBP          int64
Chol            int64
Fbs             int64
RestECG         int64
MaxHR           int64
ExAng           int64
```

```
Oldpeak       float64
Slope           int64
Ca            float64
Thal           object
AHD            object
dtype: object
```

```
for col in df.columns:
    zero_count = (df[col] == 0).sum()
    print(f"Number of zeros in {col}: {zero_count}")
```

```
Number of zeros in Unnamed: 0: 0
Number of zeros in Age: 0
Number of zeros in Sex: 97
Number of zeros in ChestPain: 0
Number of zeros in RestBP: 0
Number of zeros in Chol: 0
Number of zeros in Fbs: 258
Number of zeros in RestECG: 151
Number of zeros in MaxHR: 0
Number of zeros in ExAng: 204
Number of zeros in Oldpeak: 99
Number of zeros in Slope: 0
Number of zeros in Ca: 176
Number of zeros in Thal: 0
Number of zeros in AHD: 0
```

```
df['Age'].mean()
```

```
54.43894389438944
```

```
df.columns
```

```
Index(['Unnamed: 0', 'Age', 'Sex', 'ChestPain', 'RestBP', 'Chol',
'Fbs',
       'RestECG', 'MaxHR', 'ExAng', 'Oldpeak', 'Slope', 'Ca', 'Thal',
'AHD'],
      dtype='object')
```

```
#Now extract only Age, Sex, ChestPain, RestBP, Chol.
df.drop(columns=['Unnamed: 0', 'Fbs','RestECG', 'MaxHR', 'ExAng',
'Oldpeak', 'Slope', 'Ca', 'Thal'], inplace=True)
df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 303,\n  \"fields\": [\
n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 9,\n        \"min\": 29,\n
\"max\": 77,\n        \"num_unique_values\": 41,\n        \"samples\":
[\n          61,\n          64,\n          44\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Sex\",\n      \"properties\": {\n
```

\"dtype\": \"number\",\n   \"std\": 0,\n   \"min\": 0,\n   \"max\": 1,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   0,\n   1\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"ChestPain\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   \"asymptomatic\",\n   \"nontypical\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"RestBP\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 17,\n   \"min\": 94,\n   \"max\": 200,\n   \"num_unique_values\": 50,\n   \"samples\": [\n   124,\n   192\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Chol\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 51,\n   \"min\": 126,\n   \"max\": 564,\n   \"num_unique_values\": 152,\n   \"samples\": [\n   321,\n   187\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"AHD\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 2,\n   \"samples\": [\n   \"Yes\",\n   \"No\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
x = df.drop(columns=['AHD'])
y= df['AHD']
display(x.head())
display(y.head())
```

{"summary":"{\n \"name\": \"display(y\",\n \"rows\": 5,\n \"fields\": [\n   {\n   \"column\": \"Age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 14,\n   \"min\": 37,\n   \"max\": 67,\n   \"num_unique_values\": 4,\n   \"samples\": [\n   67,\n   41,\n   63\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Sex\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0,\n   \"min\": 0,\n   \"max\": 1,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   0,\n   1\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"ChestPain\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   \"asymptomatic\",\n   \"nontypical\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"RestBP\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 15,\n   \"min\": 120,\n   \"max\": 160,\n   \"num_unique_values\": 4,\n   \"samples\": [\n   160,\n   130\n   ],\

n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Chol\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 30,\n   \"min\": 204,\n   \"max\": 286,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   286,\n   204\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
0     No
1    Yes
2    Yes
3     No
4     No
Name: AHD, dtype: object
```

```
y.replace({'Yes': 1, 'No': 0}, inplace=True)
display(y.head())
```

```
<ipython-input-12-a8acbb79871c>:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
  y.replace({'Yes': 1, 'No': 0}, inplace=True)
```

```
0    0
1    1
2    1
3    0
4    0
Name: AHD, dtype: int64
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,
test_size=0.25, random_state=90)
print(x_train.shape)
display(x_train.head())
print(x_test.shape)
display(x_test.head())
print(y_train.shape)
display(y_train.head())
print(y_test.shape)
display(y_test.head)
```

```
(227, 5)
```

{"summary":"{\n \"name\": \"display(y_test\",\n \"rows\": 5,\n \"fields\": [\n   {\n   \"column\": \"Age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 8,\n   \"min\": 40,\n   \"max\": 62,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   59,\n

```
55,\n          57\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Sex\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"ChestPain\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"asymptomatic\",\n          \"typical\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"RestBP\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 19,\n
\"min\": 132,\n        \"max\": 178,\n        \"num_unique_values\":
4,\n        \"samples\": [\n          178,\n          140\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Chol\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 57,\n
\"min\": 199,\n        \"max\": 342,\n        \"num_unique_values\":
5,\n        \"samples\": [\n          270,\n          342\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```


```
(76, 5)
```


```
{"summary":"{\n  \"name\": \"display(y_test\",\n  \"rows\": 5,\n
\"fields\": [\n    {\n      \"column\": \"Age\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
3,\n        \"min\": 53,\n        \"max\": 61,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          53,\n
59,\n          56\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Sex\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 1,\n        \"max\": 1,\n
\"num_unique_values\": 1,\n        \"samples\": [\n          1\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"ChestPain\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n
\"asymptomatic\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"RestBP\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 11,\n        \"min\": 110,\n        \"max\": 138,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          123\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Chol\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 46,\n
\"min\": 166,\n        \"max\": 282,\n        \"num_unique_values\":
5,\n        \"samples\": [\n          282\n        ],\n
```

```
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```

```
(227,)
```

```
209    1
183    0
165    0
41     0
291    0
Name: AHD, dtype: int64
```

```
(76,)
```

```
<bound method NDFrame.head of 288    0
223    1
264    1
96     1
219    0
      ..
55     1
79     1
217    0
287    0
188    1
Name: AHD, Length: 76, dtype: int64>
```

```
x_train['ChestPain'].unique()
```

```
array(['asymptomatic', 'typical', 'nontypical', 'nonanginal'],
      dtype=object)
```

```
from sklearn.preprocessing import OrdinalEncoder

ordinal_encoder = OrdinalEncoder(categories=[['typical',
'asymptomatic', 'nonanginal', 'nontypical']])
x_train['ChestPain'] =
ordinal_encoder.fit_transform(x_train[['ChestPain']])
x_test['ChestPain'] = ordinal_encoder.transform(x_test[['ChestPain']])
print(x_train.head())
print(x_test.head())
```

```
     Age  Sex  ChestPain  RestBP  Chol
209   62    0        1.0     150   244
183   59    1        0.0     178   270
165   57    1        1.0     132   207
41    40    1        0.0     140   199
291   55    0        3.0     132   342
     Age  Sex  ChestPain  RestBP  Chol
288   56    1        3.0     130   221
223   53    1        1.0     123   282
```

```
264   61    1        1.0     138   166
96    59    1        1.0     110   239
219   59    1        1.0     138   271
```

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
x_train[['Age', 'RestBP', 'Chol']] =
scaler.fit_transform(x_train[['Age', 'RestBP', 'Chol']])
x_test[['Age', 'RestBP', 'Chol']] = scaler.transform(x_test[['Age',
'RestBP', 'Chol']])

print(x_train.head())
print(x_test.head())
```

```
          Age  Sex  ChestPain    RestBP      Chol
209  0.834084    0        1.0  1.014683 -0.025148
183  0.508203    1        0.0  2.627015  0.469594
165  0.290948    1        1.0 -0.021816 -0.729203
41  -1.555713    1        0.0  0.438850 -0.881431
291  0.073694    0        3.0 -0.021816  1.839647
          Age  Sex  ChestPain    RestBP      Chol
288  0.182321    1        3.0 -0.136982 -0.462804
223 -0.143560    1        1.0 -0.540065  0.697936
264  0.725457    1        1.0  0.323684 -1.509372
96   0.508203    1        1.0 -1.288648 -0.120290
219  0.508203    1        1.0  0.323684  0.488622
```

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(x_train, y_train)
```

```
LogisticRegression()
```

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

```
array([0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1,
0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
1,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0,
0,
       1, 0, 0, 0, 1, 1, 1, 0, 0, 0])
```

```
pd.DataFrame({'y_test':y_test, 'y_predict':y_pred})
```


```
{"summary":"{\n  \"name\": \"pd\",\n  \"rows\": 76,\n  \"fields\": [\n
{\n      \"column\": \"y_test\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
```

```
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"y_predict\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe"}
```

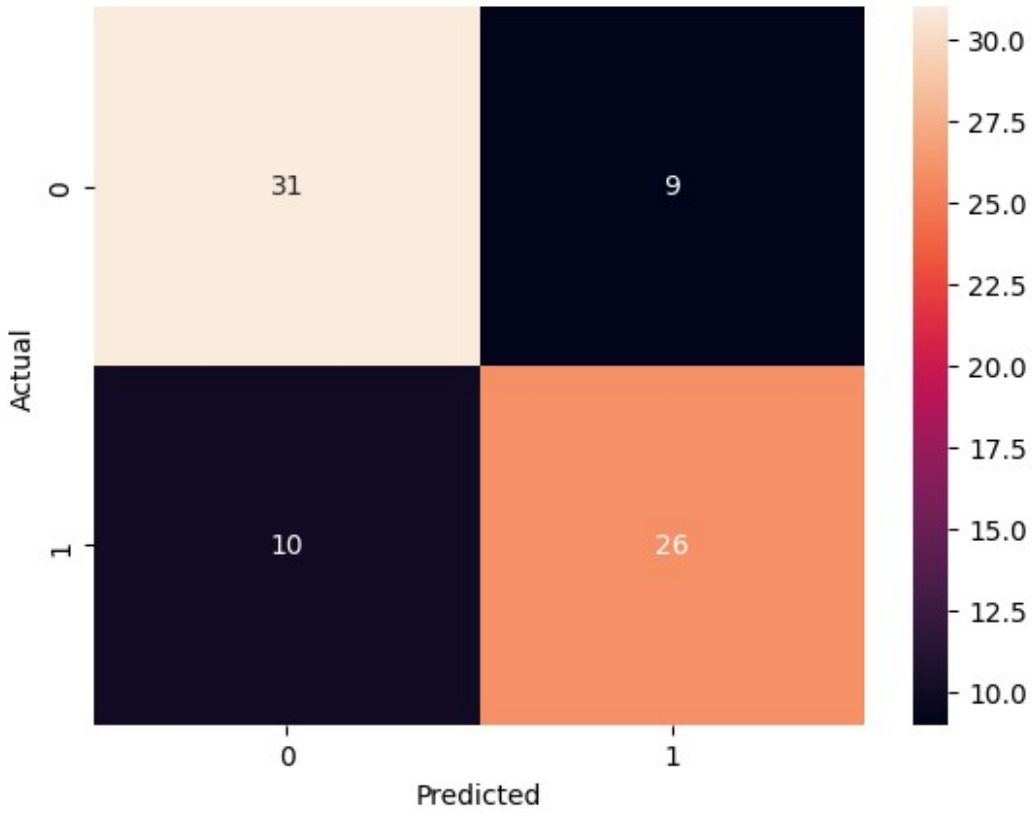
```
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix, classification_report
print('Accuracy Score: ', accuracy_score(y_test, y_pred))
print('Precision Score: ', precision_score(y_test, y_pred))
print('Recall Score: ', recall_score(y_test, y_pred))
print('F1 Score: ', f1_score(y_test, y_pred))
print('Confusion Matrix:\n ', confusion_matrix(y_test, y_pred))
print('Classification Report:\n ', classification_report(y_test,
y_pred))
```

```
Accuracy Score:  0.75
Precision Score:  0.7428571428571429
Recall Score:  0.7222222222222222
F1 Score:  0.7323943661971831
Confusion Matrix:
  [[31  9]
 [10 26]]
Classification Report:
                precision    recall  f1-score   support

           0       0.76      0.78      0.77        40
           1       0.74      0.72      0.73        36

    accuracy                           0.75        76
   macro avg       0.75      0.75      0.75        76
weighted avg       0.75      0.75      0.75        76
```

```
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```


31
9
10
26
0
1
Actual
0
1
Predicted
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0