Intro to Data Assignment – 2

Kirti Katiyar

Ques No – 2

Explanation of your rule and the intuition you used to design it

The rule I designed for labeling data rows is based on specific conditions related to medical factors like age, gender, chest pain, blood pressure, cholesterol, electrocardiogram results, exercise-induced angina, ST depression, and the number of affected vessels. The intuition behind these conditions is that they might be indicative of certain heart-related conditions or risks.

I applied this rule to predict outcomes ('yes' or 'no') for each row in the dataset. The training accuracy, which measures how well the rule performs on the entire dataset, is calculated to be a certain percentage.

Additionally, the test accuracy is calculated on a separate portion (1/10th) of the dataset that was set aside for testing. This accuracy represents how well the rule generalizes to unseen data.

In summary, the rule is designed to make predictions based on medical conditions, and the accuracy metrics evaluate its performance on both the training and test datasets.

```python
#               Intro to Data Assignment No - 2

#Question - 1 & 3

# The rule you designed to label the data rows
# The accuracy of your rule calculated on testdata.csv separated from
the main file – traindata.csv

import pandas as pd
import numpy as np

def logic(row):
    if row['Age'] > 60 and row['Gender'] == 1 and row['Chest Pain'] ==
1:
        return 'yes'
    elif row['Blood Pr'] > 140 and row['Cholesterol'] > 240:
        return 'yes'
    elif row['ElectroCardio'] == 2 and row['Max Heart Rate'] < 150:
        return 'yes'
    elif row['Exercise Induced Angina'] == 1 and row['ST depression
ind. By exercise'] > 1.0:
        return 'yes'
    elif row['Chest Pain'] == 4 and row['# of vessels'] >= 2:
        return 'yes'
    else:
        return 'no'

# Load the dataset
df = pd.read_csv('Train_data_IDS.csv')

df['Predicted_Result'] = df.apply(lambda row: logic(row), axis=1)

# Comparing predicted results & actual results
Training_accuracy = (df['Result'] == df['Predicted_Result']).mean()
print(f"Training Accuracy: {Training_accuracy:.2%}")

test_size = len(df) // 10

# Split the dataset
train_set = df.iloc[:-test_size].copy()
test_set = df.iloc[-test_size:].copy()

test_set['Predicted_Result'] = test_set.apply(lambda row: logic(row),
axis=1)

# accuracy of the test set
test_accuracy = (test_set['Result'] ==
test_set['Predicted_Result']).mean()
print(f"Accuracy on the test set: {test_accuracy:.2%}")
print(df)
```

```
Training Accuracy: 74.88%
Accuracy on the test set: 80.00%
     Age  Gender  Chest Pain  Blood Pr  Cholesterol  Blood Sugar  \
0     63       1           1       145          233            1
1     67       1           4       160          286            0
2     67       1           4       120          229            0
3     37       1           3       130          250            0
4     41       0           2       130          204            0
..   ...     ...         ...       ...          ...          ...
198   50       0           2       120          244            0
199   59       1           1       160          273            0
200   50       0           4       110          254            0
201   64       0           4       180          325            0
202   57       1           3       150          126            1

     ElectroCardio  Max Heart Rate  Exercise Induced Angina  \
0                2             150                        0
1                2             108                        1
2                2             129                        1
3                0             187                        0
4                2             172                        0
..             ...             ...                      ...
198              0             162                        0
199              2             125                        0
200              2             159                        0
201              0             154                        1
202              0             173                        0

     ST depression ind. By exercise  Slope of the peak exercise   \
0                               2.3                            3
1                               1.5                            2
2                               2.6                            2
3                               3.5                            3
4                               1.4                            1
..                              ...                          ...
198                             1.1                            1
199                             0.0                            1
200                             0.0                            1
201                             0.0                            1
202                             0.2                            1

     # of vessels  defect Result Predicted_Result
0               0       6     no              yes
1               3       3    yes              yes
2               2       7    yes              yes
3               0       3     no               no
4               0       3     no               no
..            ...     ...    ...              ...
198             0       3     no               no
199             0       3    yes              yes
```

```
200              0       3     no                 no
201              0       3     no                yes
202              1       7     no                 no

[203 rows x 15 columns]
```