ML S1 Q1 Project Report: Heart Disease Predictive Model

Team Members:
Adarsh B, Aryan G

10/21/2024

Table of Contents

Part 1 – Statement/Project Goal

According to the CDC, every year about 695,000 people in the country die from heart disease, equivalent to 1 out of 5 deaths in the United States. Out of all of those deaths, approximately 1 out of 3 deaths could have been prevented with earlier diagnosis. This dataset contains heart disease data from the 5 most popular independent heart disease datasets.

Hungarian
Cleveland
Switzerland
Long Beach VA
Statlog (Heart) Data Set

Our project aims to use the combined heart disease dataset to predict whether an individual has heart disease based on key medical attributes, potentially improving early diagnosis rates. By classifying heart disease using machine learning techniques, our project hopes to contribute to more effective prevention strategies and healthcare interventions, addressing a significant public health issue.

Part 2 – Description of Dataset

The dataset contains information about heart disease, with various features used for predicting heart conditions. This dataset is curated by combining 5 popular heart disease datasets about individuals related to heart disease.

Dataset Summary:
- File Name: heart.csv
- Number of Instances (Rows): 1190
- Number of Non-Class Attributes (Columns): 11
- Missing Values: None
- Class Attribute: "target" represents whether the individual has heart disease (1) or not (0).

Attributes (Columns):
1. age: Age of the patient
2. sex: Gender of the patient (1 = Male, 0 = Female)
3. chest pain type: Chest pain type (4 values)
4. resting bp s: Resting blood pressure (in mmHg)
5. cholesterol: Serum cholesterol in mg/dl
6. fasting blood sugar: Fasting blood sugar > 120 mg/dl (1 = True, 0 = False)

7. resting ecg: Resting electrocardiographic results (values 0, 1, 2)
8. max heart rate: Maximum heart rate achieved
9. exercise angina: Exercise induced angina (1 = Yes, 0 = No)
10. oldpeak: ST depression induced by exercise relative to rest
11. ST slope: The slope of the peak exercise ST segment

Class Attribute (Target):
Target Distribution: The dataset contains two classes:
- Class 0: No heart disease (47.1% of the instances)
- Class 1: Heart disease (52.9% of the instances)

Data Characteristics:
- Missing Values: No missing values.
- Data Distribution: The dataset has a slightly imbalanced class distribution, with a skew towards individuals diagnosed with heart disease (52.9%).

Part 3 – Pre-Processing

1. Normalization
To ensure that our models don't overestimate the impact of one specific attribute, we intend to perform min-max normalization to transpose all values on a scale from 0 to 1. This process would have to be done for the attributes: Age, resting bp s, cholesterol, max heart rate, oldpeak, and st slope.

2. Missing and Disguised Values
While auditing the dataset we discovered that the attribute "cholesterol" had the following distribution. There were a large number of values that were zero, 172 instances out of 1190, while there were 221 other distinct values for that attribute. The likelihood of over 10% of the population having the same value for a numeric attribute is very low, especially when that attribute is cholesterol level, which is not possible. Due to the high number of outlier values, we decided to change all the values that were 0 to the median value for the attribute, which is 210. We then also had to deal with a couple of disguised missing entries in the "resting bp s" attribute. For this attribute, some values were zero, which is again not possible in a living person. Since there were only five disguised entries for this value, we replaced all values that were zero with the mean for the attribute, 132. Finally, we performed min-max normalization for all attributes that were not between 0 and 1.

3. Train-test split
To split the data into training and testing sets we will randomly select 85% of instances to use for the training set and then use the remaining 15% for the testing set. It is appropriate to randomly split because the distribution of the classes is close to even, so there is highly unlikely to be a major difference in the class distribution of the test set and the training set. This split will result in 1012 instances in the training set and 178 instances in the testing set. We used python to split the dataset and test dataset for all 5 methods. Our code is shown on the following page:

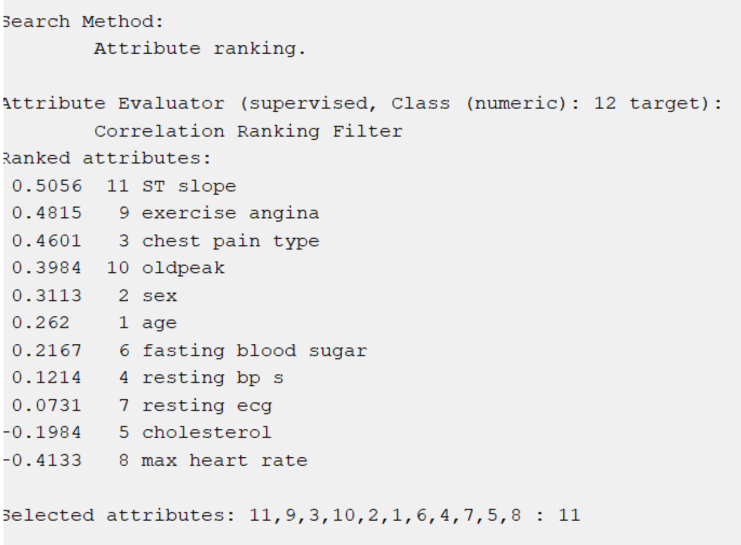
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

df = pd.read_csv('heart_full.csv')

print(df.columns)
df1 = df[['ST', 'angina', 'chest', 'oldpeak', 'sex', 'hr', 'target']]
df2 = df[['age', 'sex', 'chest', 'cholesterol', 'hr', 'angina', 'oldpeak',
'ST', 'target']]
df3 = df[['chest', 'ST', 'ecg', 'bs', 'sex', 'cholesterol', 'target']]
df4 = df[['ST', 'angina', 'chest', 'oldpeak', 'sex', 'hr', 'cholesterol',
'target']]
df5 = df[['ecg', 'age', 'bps', 'cholesterol', 'hr', 'target']]

df1train, df1test = train_test_split(df1, test_size=0.15)
df1train.to_csv('df1train.csv', index=False)
df1test.to_csv('df1test.csv', index=False)

df2train, df2test = train_test_split(df2, test_size=0.15)
df2train.to_csv('df2train.csv', index=False)
df2test.to_csv('df2test.csv', index=False)

df3train, df3test = train_test_split(df3, test_size=0.15)
df3train.to_csv('df3train.csv', index=False)
df3test.to_csv('df3test.csv', index=False)

df4train, df4test = train_test_split(df4, test_size=0.15)
df4train.to_csv('df4train.csv', index=False)
df4test.to_csv('df4test.csv', index=False)

df5train, df5test = train_test_split(df5, test_size=0.15)
df5train.to_csv('df5rain.csv', index=False)
df5test.to_csv('df5test.csv', index=False)
```

Part 4 – Attribute Selection Algorithms & Model Classifiers Used

4.1 – Attribute Selection Algorithms

Method 1: Correlation Analysis

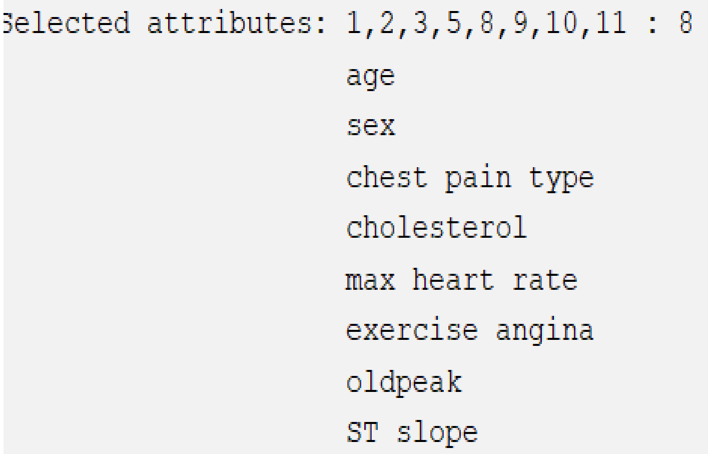
We did correlation analysis in weka and chose the attributes that had an absolute correlation higher than 0.3. Hence the attributes selected were ST slope, exercise angina, chest pain type, oldpeak, sex, and max heart rate.



```
Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (numeric): 12 target):
        Correlation Ranking Filter
Ranked attributes:
 0.5056   11 ST slope
 0.4815    9 exercise angina
 0.4601    3 chest pain type
 0.3984   10 oldpeak
 0.3113    2 sex
 0.262     1 age
 0.2167    6 fasting blood sugar
 0.1214    4 resting bp s
 0.0731    7 resting ecg
-0.1984    5 cholesterol
-0.4133    8 max heart rate

Selected attributes: 11,9,3,10,2,1,6,4,7,5,8 : 11
```

Method 2: Cfs subset eval

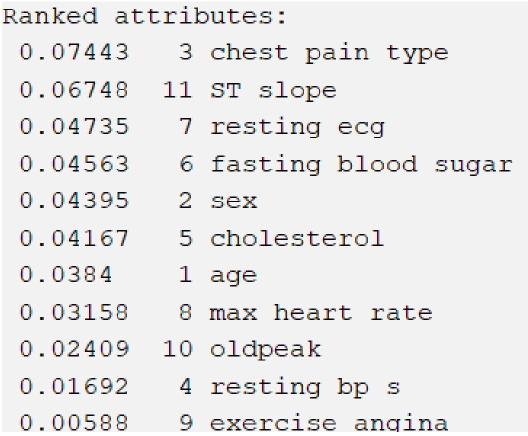
For this method we picked the attributes that were selected by the cfs subset evaluator, which evaluates the worth of a subset of attributes by taking into account the prediction accuracy of each individual attribute while minimizing their redundancy by checking against their intercorrelation. The attributes that we selected were age, sex, chest pain type, cholesterol, max heart rate, exercise angina, oldpeak, ST slope.



```
Selected attributes: 1,2,3,5,8,9,10,11 : 8
                     age
                     sex
                     chest pain type
                     cholesterol
                     max heart rate
                     exercise angina
                     oldpeak
                     ST slope
```

Method 3: Relief Attribute Evaluator

This method evaluates the worth of each attribute by repeatedly sampling an instance and then considers the value of each attribute for the nearest

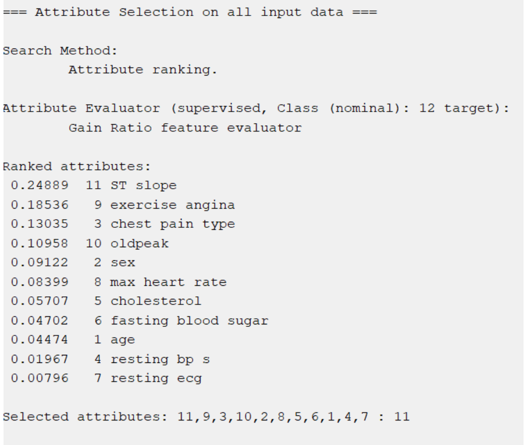


```
Ranked attributes:
 0.07443    3 chest pain type
 0.06748   11 ST slope
 0.04735    7 resting ecg
 0.04563    6 fasting blood sugar
 0.04395    2 sex
 0.04167    5 cholesterol
 0.0384     1 age
 0.03158    8 max heart rate
 0.02409   10 oldpeak
 0.01692    4 resting bp s
 0.00588    9 exercise angina
```

instance of the same and different class. The advantage of this method is that it can work for both discrete and continuous data. We set our cutoff value to .04 and selected the attributes; chest pain type, ST slope, resting ecg, fasting blood sugar, sex, and cholesterol.

Method 4: Gain Ratio

This method is an optimization to the info gain evaluator, which uses a decision tree to partition attributes and picking the best one by taking into account predictive ability, by reducing the entropy produced in splitting a set of a attributes and picking the best one out of them. We set our cutoff value as .09 and selected the attributes ST slope, exercise angina, chest pain type, oldpeak, and sex.



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 12 target):
        Gain Ratio feature evaluator

Ranked attributes:
 0.24889  11 ST slope
 0.18536   9 exercise angina
 0.13035   3 chest pain type
 0.10958  10 oldpeak
 0.09122   2 sex
 0.08399   8 max heart rate
 0.05707   5 cholesterol
 0.04702   6 fasting blood sugar
 0.04474   1 age
 0.01967   4 resting bp s
 0.00796   7 resting ecg

Selected attributes: 11,9,3,10,2,8,5,6,1,4,7 : 11
```

Method 5: Self Selection

For our 5th dataset we chose attributes that we felt were underrepresented in the other 4 datasets, even though that would possibly indicate that these attributes did not have great predictive power, we still wanted to investigate the performance of these attributes on predicting the class. The attributes that we selected were resting ecg, age, resting bp s, cholesterol, and max heart rate.

4.2 Classifiers Used

OneR: We used OneR as since the model is a relatively simple algorithm, we thought that it would serve as a good baseline for the rest of our classifiers so that we could test the real effectiveness of the more complex algorithms and check for a significant difference in performance.

Random Forest: We used Random Forest since it is a very popular classifier and is extremely versatile in its applications. Thus, we expected Random Forest to be kind of the gold standard for our classifiers and perform highly in all of the performance metrics.

Naive Bayes: Our interest in using Naive Bayes was our curiosity in the possible differences in performance between deterministic and probabilistic networks. Since the rest of our classifiers are deterministic, we wanted to investigate the performance of Naive Bayes, which is probabilistic, on the data.

J48: Although one of our classifiers is already a tree algorithm, the Random Forest classifier algorithm differs from decision trees in that it is not strictly deterministic, which is why we wanted to assess the difference between the J48 classifier and Random Forest

Part 5 – Results

5.1 Performance Metrics

Accuracy: We used accuracy in order to get a comprehensive view of our models performance, which accuracy is great for because it pools all correct predictions instead of dividing them by class, which gives us that comprehensive review.

TP Rate: In order to evaluate our models' true performance, we used TP rate in order to see how well our models can predict both the class value of False (No heart disease) and True (deart disease). This helps us determine our models success in its main purpose, correctly identifying heart disease.

FP Rate: We chose this metric for similar reasons as the TP rate as using this, we can detect what percentage of people will be misdiagnosed as having no heart disease when they actually do.

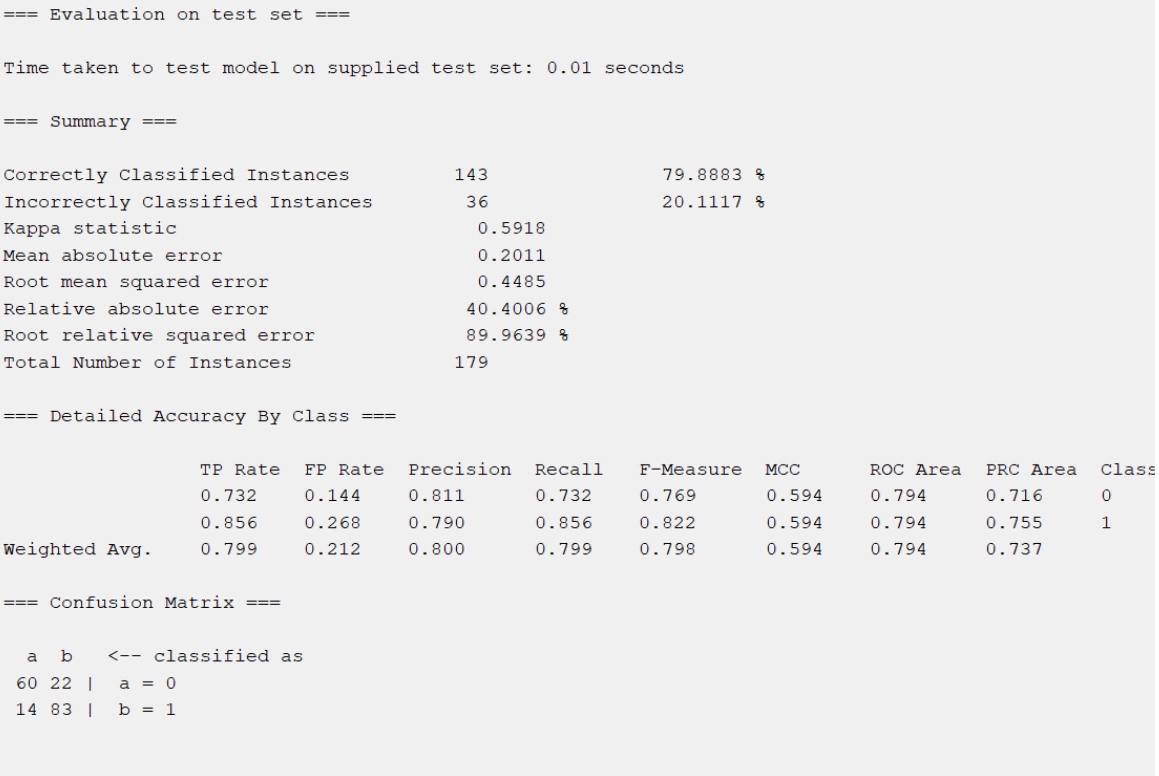
AUC: As our dataset's class distribution is fairly balanced, it is safe to use AUC as a performance metric. AUC provides a good intersection between precision and recall, and allows us to do a more simple evaluation, being that models with higher AUC are usually better in balanced datasets.

F-Measure: To evaluate our models' performances, we used F-measure in order to see the balance between precision and recall at the class value level. This way we get a deeper understanding of our dataset that we might have missed if we just used AUC.

5.2 Results

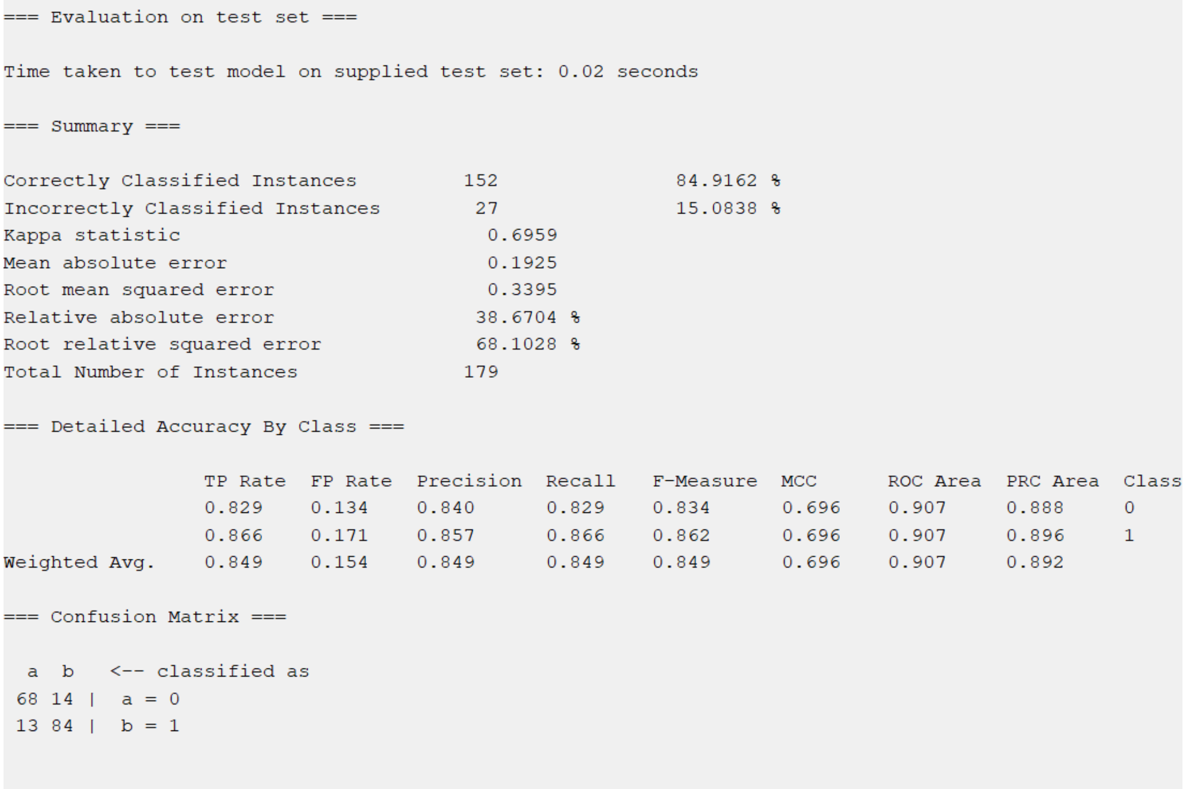
Dataset 1:

Here are the results for our first dataset, which contained the attributes: ST slope, exercise angina, chest pain type, oldpeak, sex, and max heart rate



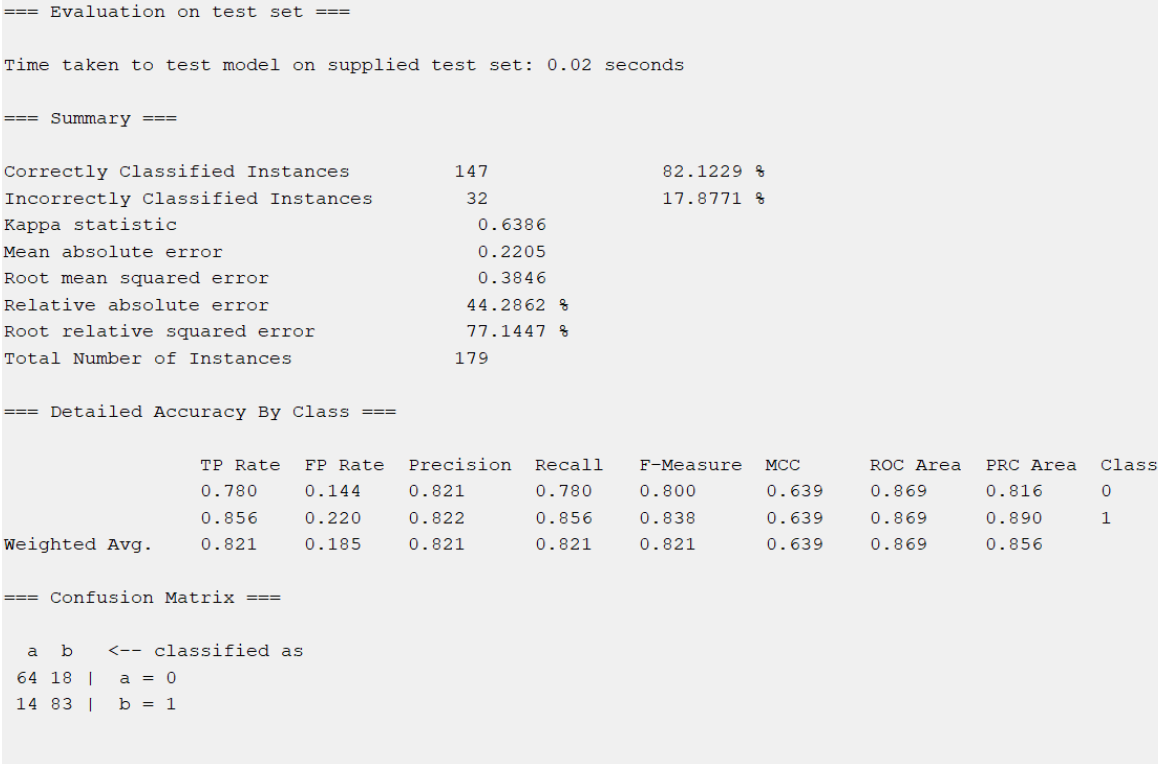
```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         143               79.8883 %
Incorrectly Classified Instances        36               20.1117 %
Kappa statistic                          0.5918
Mean absolute error                      0.2011
Root mean squared error                  0.4485
Relative absolute error                 40.4006 %
Root relative squared error             89.9639 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.732    0.144    0.811      0.732    0.769      0.594    0.794     0.716     0
                 0.856    0.268    0.790      0.856    0.822      0.594    0.794     0.755     1
Weighted Avg.    0.799    0.212    0.800      0.799    0.798      0.594    0.794     0.737

=== Confusion Matrix ===

  a  b   <-- classified as
 60 22 |  a = 0
 14 83 |  b = 1
```

OneR



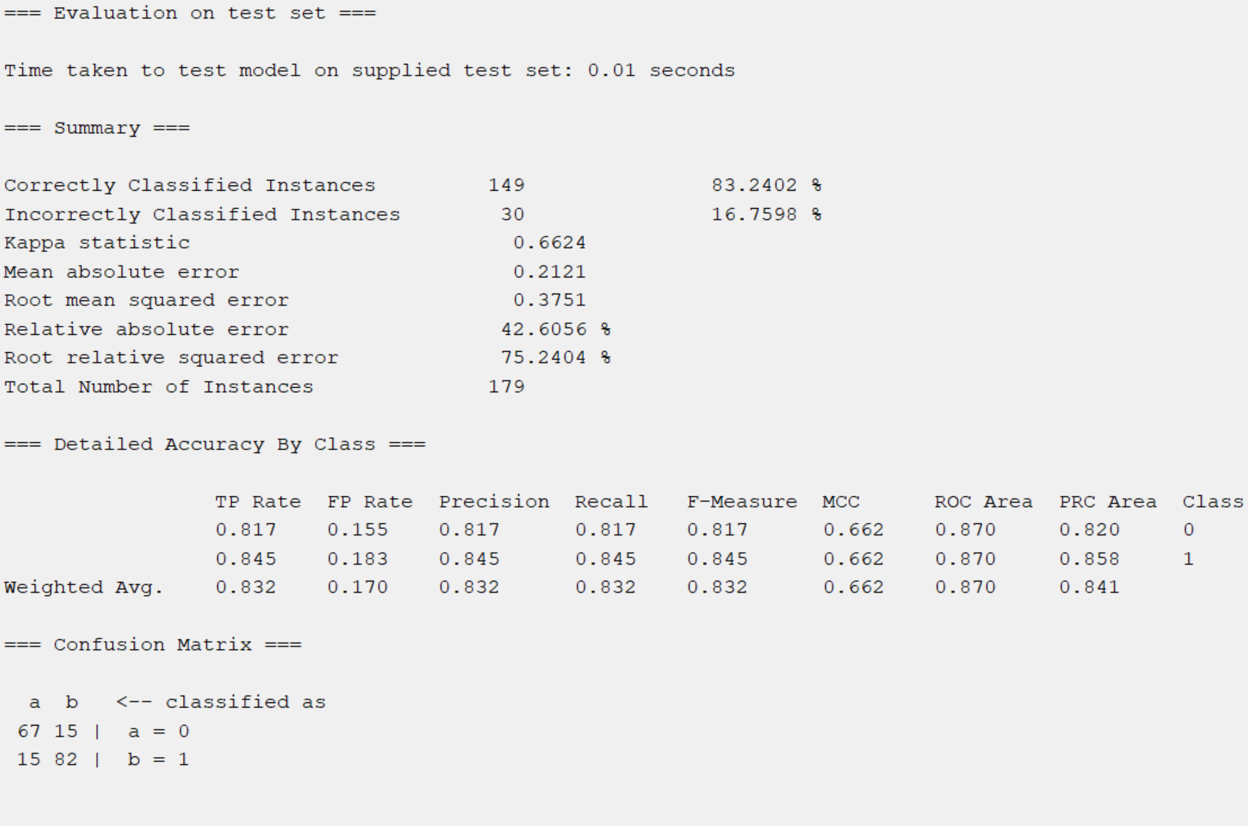
```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.02 seconds

=== Summary ===

Correctly Classified Instances         152               84.9162 %
Incorrectly Classified Instances        27               15.0838 %
Kappa statistic                          0.6959
Mean absolute error                      0.1925
Root mean squared error                  0.3395
Relative absolute error                 38.6704 %
Root relative squared error             68.1028 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.829    0.134    0.840      0.829    0.834      0.696    0.907     0.888     0
                 0.866    0.171    0.857      0.866    0.862      0.696    0.907     0.896     1
Weighted Avg.    0.849    0.154    0.849      0.849    0.849      0.696    0.907     0.892

=== Confusion Matrix ===

  a  b   <-- classified as
 68 14 |  a = 0
 13 84 |  b = 1
```

Random Forest



```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.02 seconds

=== Summary ===

Correctly Classified Instances         147               82.1229 %
Incorrectly Classified Instances        32               17.8771 %
Kappa statistic                          0.6386
Mean absolute error                      0.2205
Root mean squared error                  0.3846
Relative absolute error                 44.2862 %
Root relative squared error             77.1447 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.780    0.144    0.821      0.780    0.800      0.639    0.869     0.816     0
                 0.856    0.220    0.822      0.856    0.838      0.639    0.869     0.890     1
Weighted Avg.    0.821    0.185    0.821      0.821    0.821      0.639    0.869     0.856

=== Confusion Matrix ===

  a  b   <-- classified as
 64 18 |  a = 0
 14 83 |  b = 1
```

Naive Bayes



```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         149               83.2402 %
Incorrectly Classified Instances        30               16.7598 %
Kappa statistic                          0.6624
Mean absolute error                      0.2121
Root mean squared error                  0.3751
Relative absolute error                 42.6056 %
Root relative squared error             75.2404 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.817    0.155    0.817      0.817    0.817      0.662    0.870     0.820     0
                 0.845    0.183    0.845      0.845    0.845      0.662    0.870     0.858     1
Weighted Avg.    0.832    0.170    0.832      0.832    0.832      0.662    0.870     0.841

=== Confusion Matrix ===

  a  b   <-- classified as
 67 15 |  a = 0
 15 82 |  b = 1
```

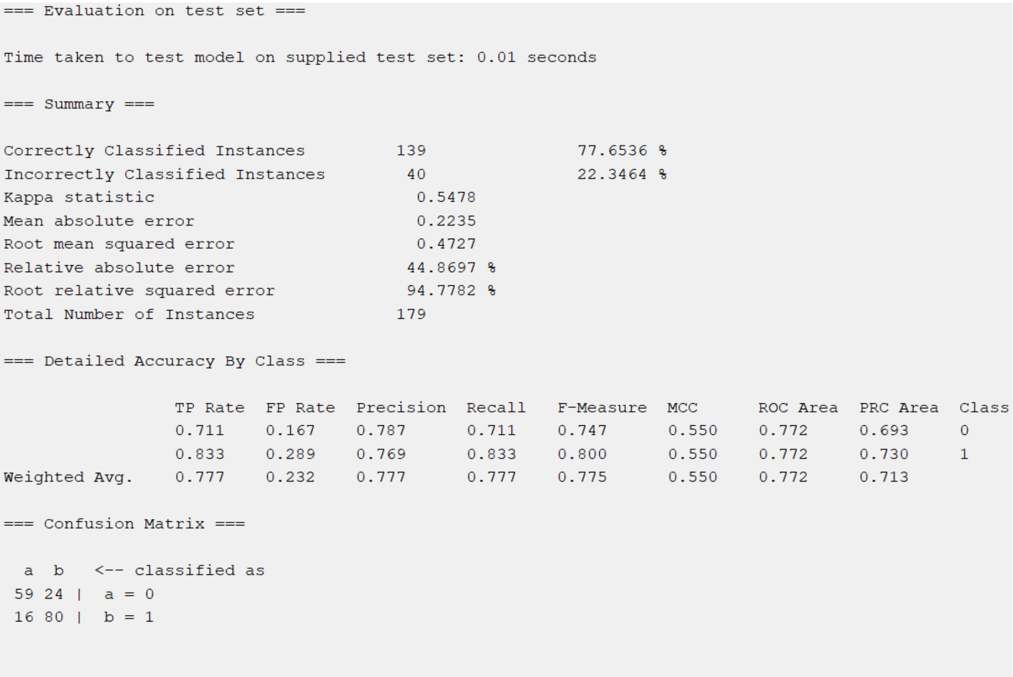
J48

Summary Table for Dataset 1:

	Performance Metric					
		Accuracy	TP Rate (False/True)	FP Rate (False/True)	AUC	F-Measure (False/True)
Classifier Used	OneR	.799	.732 .856	.144 .268	.794	.769 .822
	Random Forest	.849	.829 .866	.134 .171	.907	.834 .862
	Naive Bayes	.821	.780 .856	.144 .220	.869	.800 .838
	J48	.832	.817 .845	.155 .183	.870	.817 .845

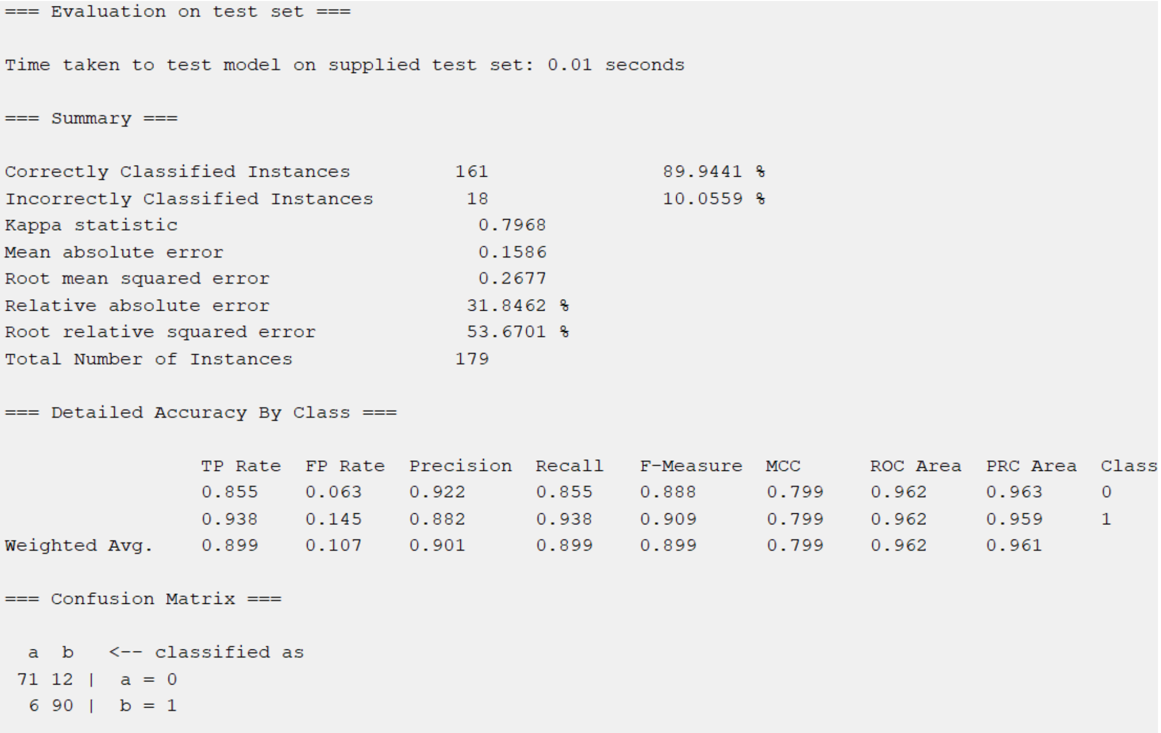
Dataset 2:

Here are the results for our second dataset, which contained the attributes: age, sex, chest pain type, cholesterol, max heart rate, exercise angina, oldpeak, and ST slope.



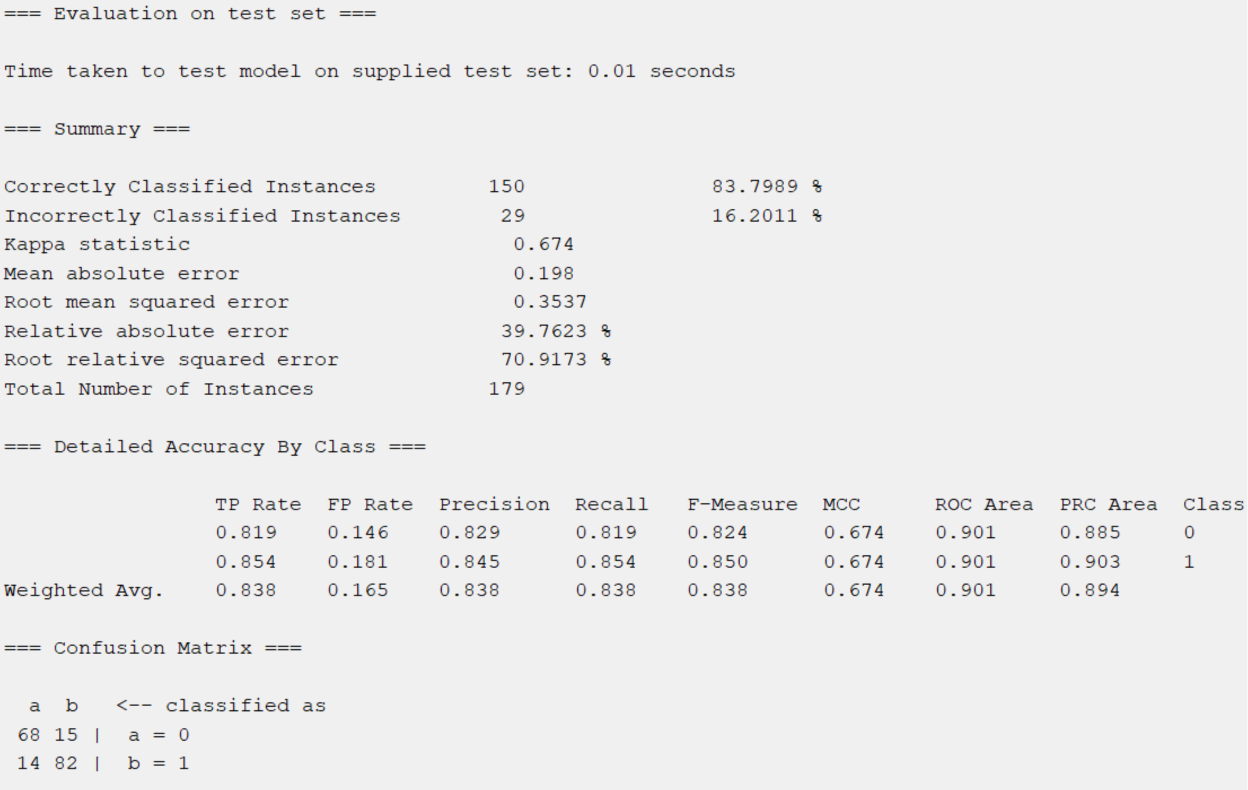
```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         139               77.6536 %
Incorrectly Classified Instances        40               22.3464 %
Kappa statistic                          0.5478
Mean absolute error                      0.2235
Root mean squared error                  0.4727
Relative absolute error                 44.8697 %
Root relative squared error             94.7782 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.711    0.167    0.787      0.711    0.747      0.550    0.772     0.693     0
                 0.833    0.289    0.769      0.833    0.800      0.550    0.772     0.730     1
Weighted Avg.    0.777    0.232    0.777      0.777    0.775      0.550    0.772     0.713

=== Confusion Matrix ===

  a  b   <-- classified as
 59 24 |  a = 0
 16 80 |  b = 1
```

OneR



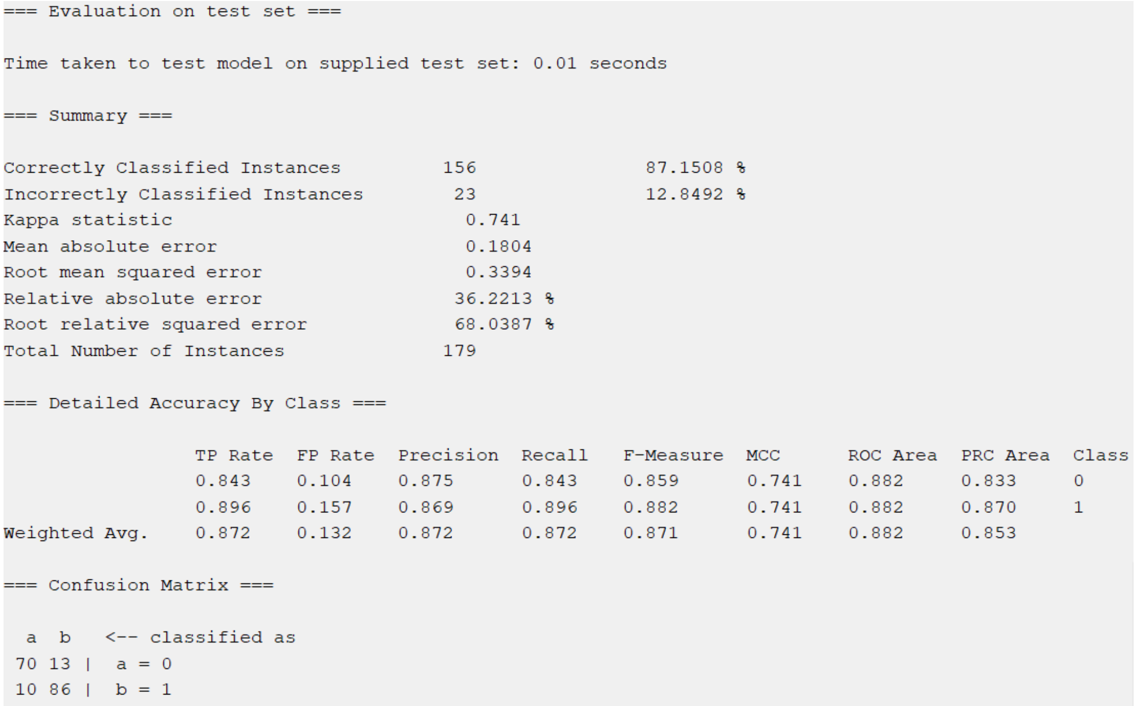
```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         161               89.9441 %
Incorrectly Classified Instances        18               10.0559 %
Kappa statistic                          0.7968
Mean absolute error                      0.1586
Root mean squared error                  0.2677
Relative absolute error                 31.8462 %
Root relative squared error             53.6701 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.855    0.063    0.922      0.855    0.888      0.799    0.962     0.963     0
                 0.938    0.145    0.882      0.938    0.909      0.799    0.962     0.959     1
Weighted Avg.    0.899    0.107    0.901      0.899    0.899      0.799    0.962     0.961

=== Confusion Matrix ===

  a  b   <-- classified as
 71 12 |  a = 0
  6 90 |  b = 1
```

Random Forest



```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         150               83.7989 %
Incorrectly Classified Instances        29               16.2011 %
Kappa statistic                          0.674
Mean absolute error                      0.198
Root mean squared error                  0.3537
Relative absolute error                 39.7623 %
Root relative squared error             70.9173 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.819    0.146    0.829      0.819    0.824      0.674    0.901     0.885     0
                 0.854    0.181    0.845      0.854    0.850      0.674    0.901     0.903     1
Weighted Avg.    0.838    0.165    0.838      0.838    0.838      0.674    0.901     0.894

=== Confusion Matrix ===

  a  b   <-- classified as
 68 15 |  a = 0
 14 82 |  b = 1
```

Naive Bayes



```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         156               87.1508 %
Incorrectly Classified Instances        23               12.8492 %
Kappa statistic                          0.741
Mean absolute error                      0.1804
Root mean squared error                  0.3394
Relative absolute error                 36.2213 %
Root relative squared error             68.0387 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.843    0.104    0.875      0.843    0.859      0.741    0.882     0.833     0
                 0.896    0.157    0.869      0.896    0.882      0.741    0.882     0.870     1
Weighted Avg.    0.872    0.132    0.872      0.872    0.871      0.741    0.882     0.853

=== Confusion Matrix ===

  a  b   <-- classified as
 70 13 |  a = 0
 10 86 |  b = 1
```

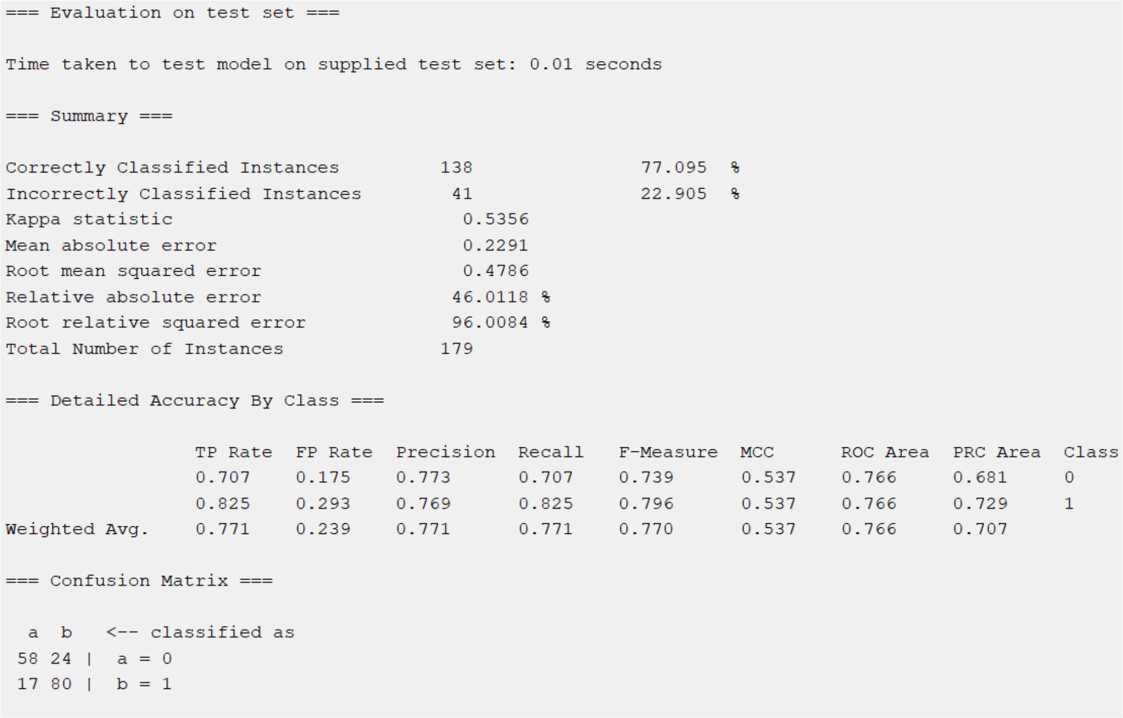
J48

Summary Table for Dataset 2:

	Performance Metric					
Classifier Used		Accuracy	TP Rate (False/True)	FP Rate (False/True	AUC	F-Measure (False/True)
	OneR	.777	.711 .833	.167 .289	.772	.747 .800
	Random Forest	.899	.855 .938	.063 .145	.962	.888 .909
	Naive Bayes	.838	.819 .854	.146 .181	.901	.824 .850
	J48	.872	.843 .896	.104 .157	.882	.859 .882

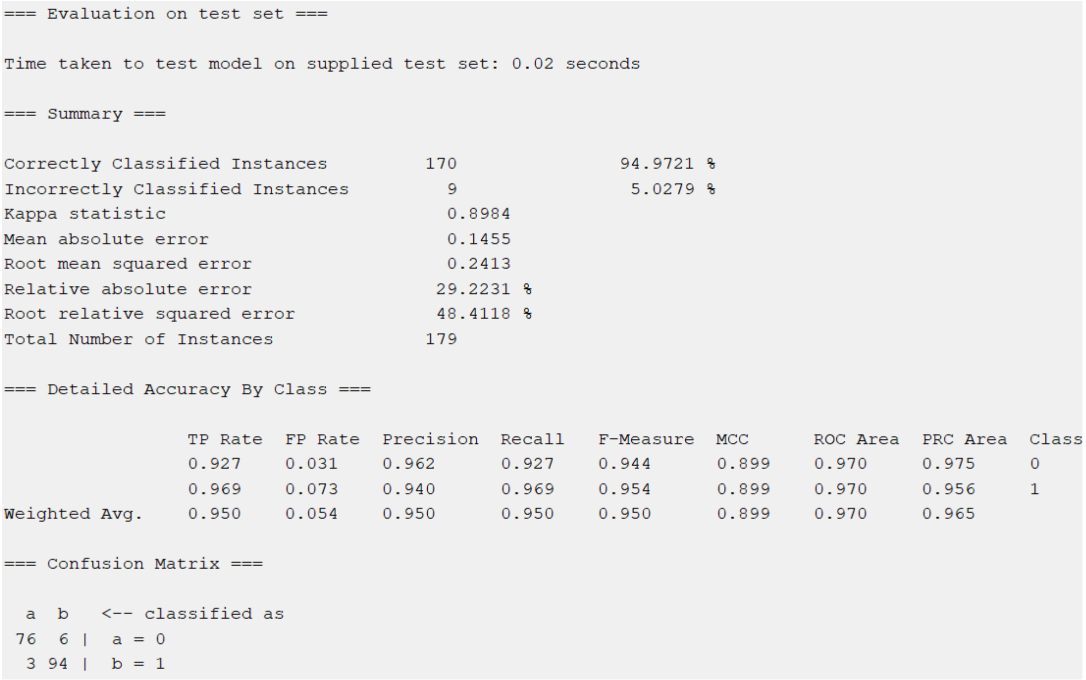
Dataset 3:

Here are the results for our third dataset, which contained the attributes: chest pain type, ST slope, resting ecg, fasting blood sugar, sex, and cholesterol.



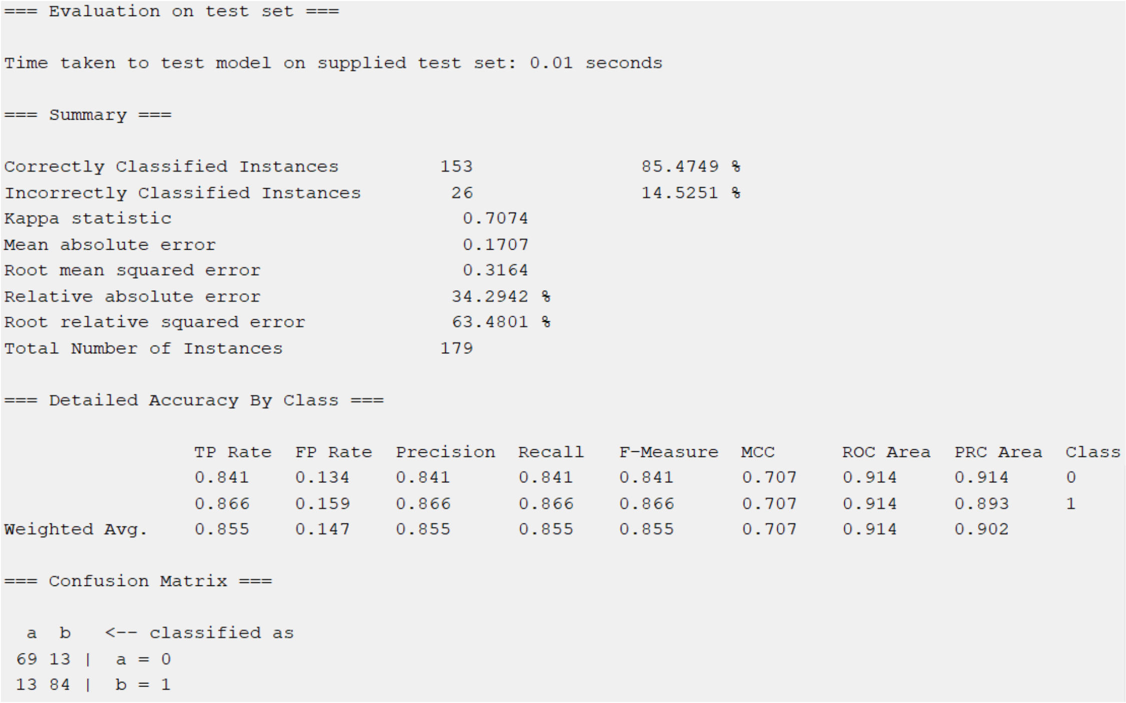
```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         138               77.095  %
Incorrectly Classified Instances        41               22.905  %
Kappa statistic                          0.5356
Mean absolute error                      0.2291
Root mean squared error                  0.4786
Relative absolute error                 46.0118 %
Root relative squared error             96.0084 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.707    0.175    0.773      0.707    0.739      0.537    0.766     0.681     0
                 0.825    0.293    0.769      0.825    0.796      0.537    0.766     0.729     1
Weighted Avg.    0.771    0.239    0.771      0.771    0.770      0.537    0.766     0.707

=== Confusion Matrix ===

  a  b   <-- classified as
 58 24 |  a = 0
 17 80 |  b = 1
```

OneR



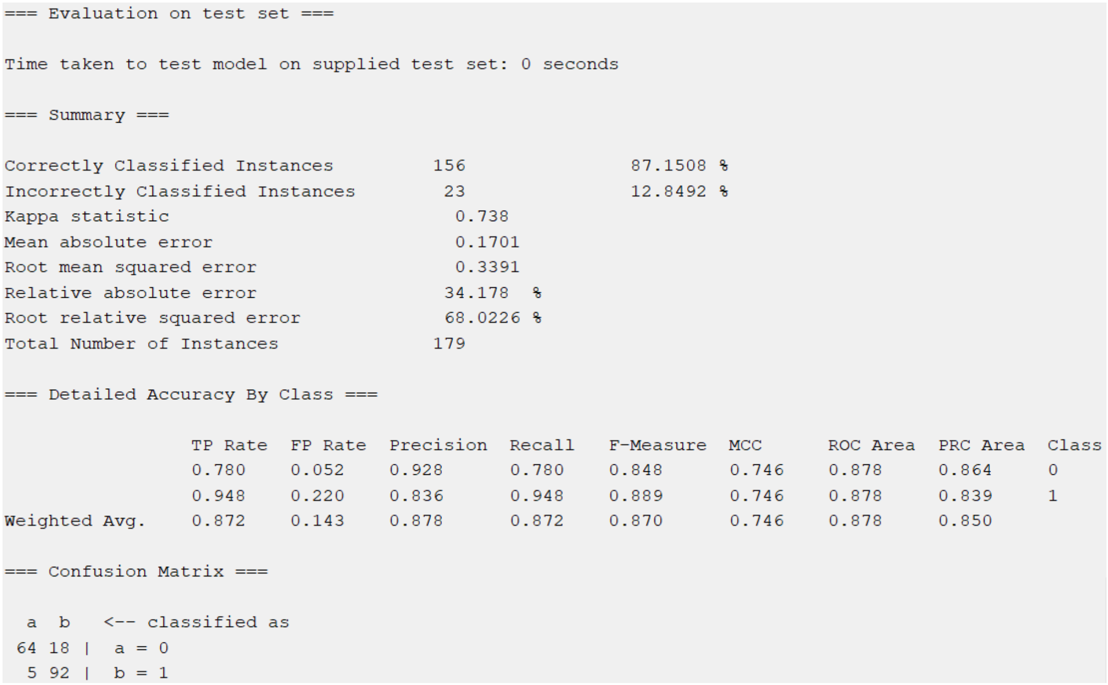
```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.02 seconds

=== Summary ===

Correctly Classified Instances         170               94.9721 %
Incorrectly Classified Instances         9                5.0279 %
Kappa statistic                          0.8984
Mean absolute error                      0.1455
Root mean squared error                  0.2413
Relative absolute error                 29.2231 %
Root relative squared error             48.4118 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.927    0.031    0.962      0.927    0.944      0.899    0.970     0.975     0
                 0.969    0.073    0.940      0.969    0.954      0.899    0.970     0.956     1
Weighted Avg.    0.950    0.054    0.950      0.950    0.950      0.899    0.970     0.965

=== Confusion Matrix ===

  a  b   <-- classified as
 76  6 |  a = 0
  3 94 |  b = 1
```

Random Forest



```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         153               85.4749 %
Incorrectly Classified Instances        26               14.5251 %
Kappa statistic                          0.7074
Mean absolute error                      0.1707
Root mean squared error                  0.3164
Relative absolute error                 34.2942 %
Root relative squared error             63.4801 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.841    0.134    0.841      0.841    0.841      0.707    0.914     0.914     0
                 0.866    0.159    0.866      0.866    0.866      0.707    0.914     0.893     1
Weighted Avg.    0.855    0.147    0.855      0.855    0.855      0.707    0.914     0.902

=== Confusion Matrix ===

  a  b   <-- classified as
 69 13 |  a = 0
 13 84 |  b = 1
```

Naive Bayes



```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         156               87.1508 %
Incorrectly Classified Instances        23               12.8492 %
Kappa statistic                          0.738
Mean absolute error                      0.1701
Root mean squared error                  0.3391
Relative absolute error                 34.178  %
Root relative squared error             68.0226 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.780    0.052    0.928      0.780    0.848      0.746    0.878     0.864     0
                 0.948    0.220    0.836      0.948    0.889      0.746    0.878     0.839     1
Weighted Avg.    0.872    0.143    0.878      0.872    0.870      0.746    0.878     0.850

=== Confusion Matrix ===

  a  b   <-- classified as
 64 18 |  a = 0
  5 92 |  b = 1
```

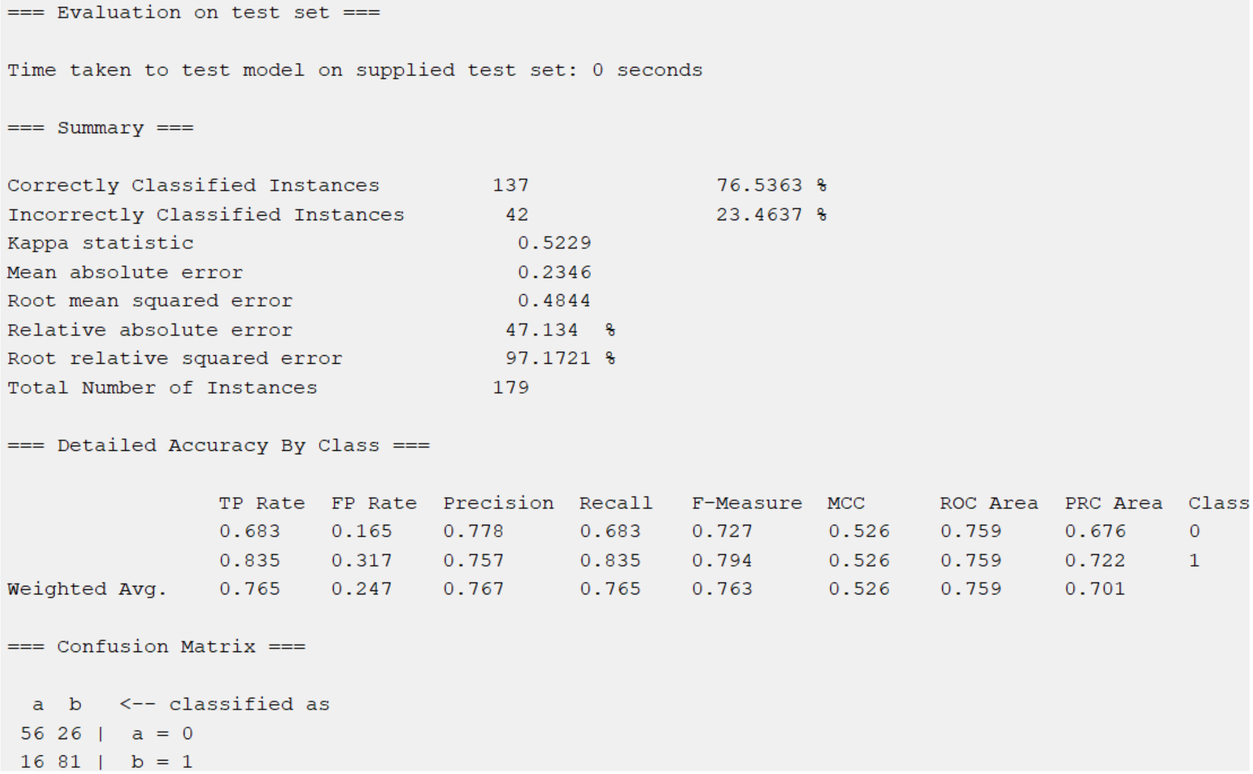
J48

Summary Table for Dataset 3:

		Performance Metric				
Classifier Used		Accuracy	TP Rate (False/True)	FP Rate (False/True	AUC	F-Measure (False/True)
	OneR	.771	.707 .825	.175 .293	.766	.739 .796
	Random Forest	.950	.927 .969	.031 .073	.970	.944 .954
	Naive Bayes	.855	.841 .866	.134 .159	.914	.841 .866
	J48	.872	.780 .948	.052 .220	.878	.848 .889

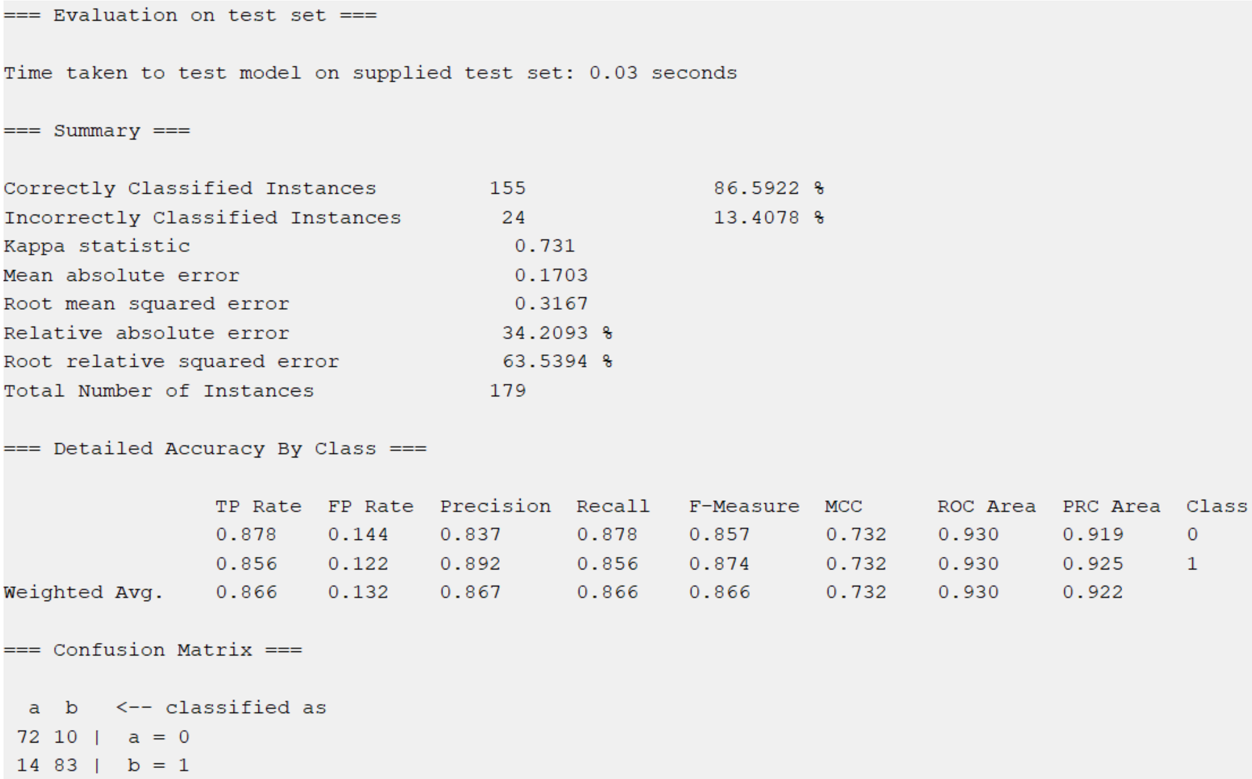
Dataset 4:

Here are the results for our fourth dataset, which contained the attributes: ST slope, exercise angina, chest pain type, oldpeak, and sex.



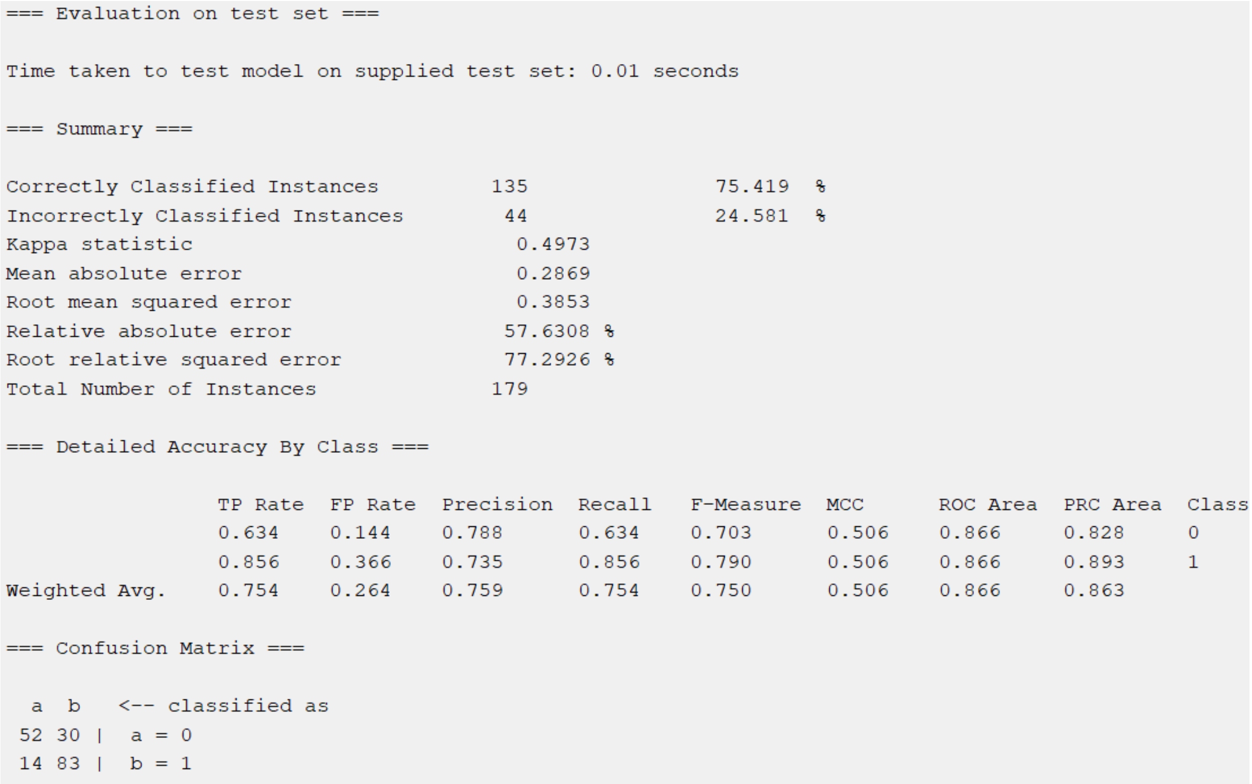
```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         137               76.5363 %
Incorrectly Classified Instances        42               23.4637 %
Kappa statistic                          0.5229
Mean absolute error                      0.2346
Root mean squared error                  0.4844
Relative absolute error                 47.134  %
Root relative squared error             97.1721 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.683    0.165    0.778      0.683    0.727      0.526    0.759     0.676     0
                 0.835    0.317    0.757      0.835    0.794      0.526    0.759     0.722     1
Weighted Avg.    0.765    0.247    0.767      0.765    0.763      0.526    0.759     0.701

=== Confusion Matrix ===

  a  b   <-- classified as
 56 26 |  a = 0
 16 81 |  b = 1
```

OneR



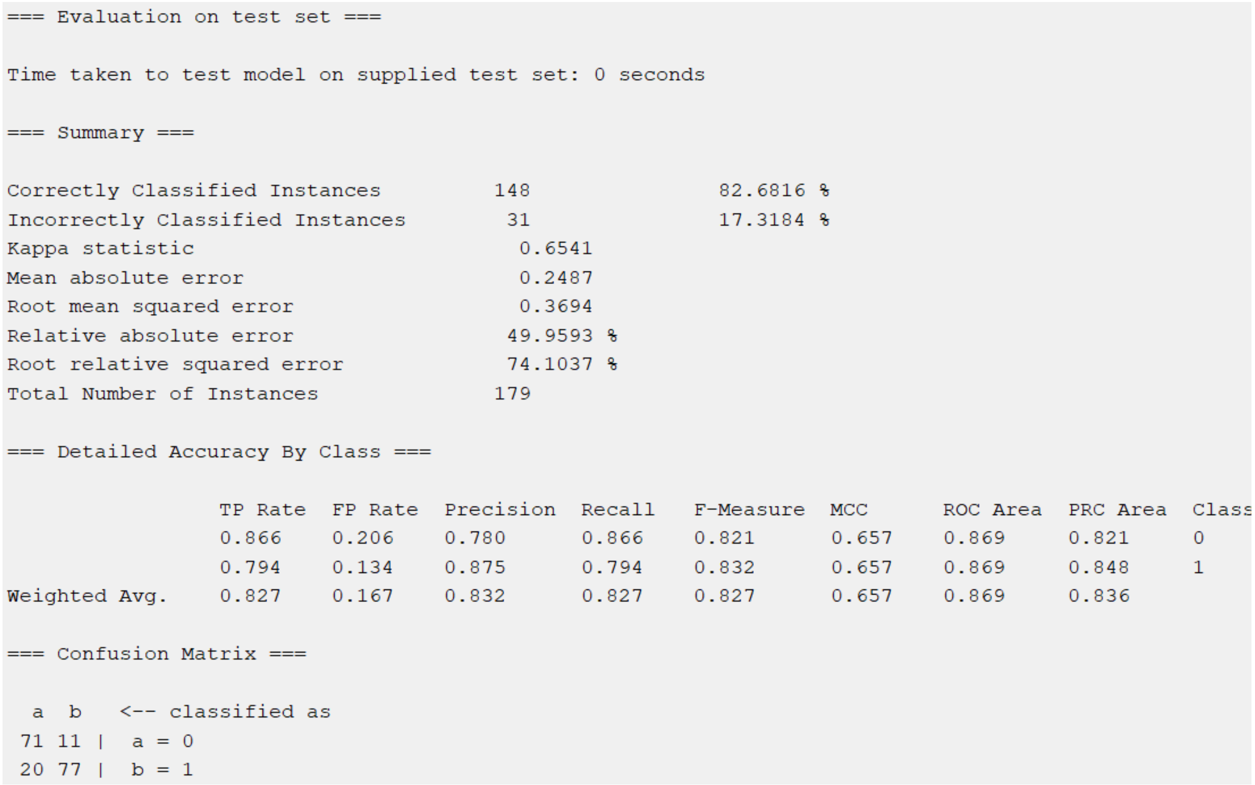
```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.03 seconds

=== Summary ===

Correctly Classified Instances         155               86.5922 %
Incorrectly Classified Instances        24               13.4078 %
Kappa statistic                          0.731
Mean absolute error                      0.1703
Root mean squared error                  0.3167
Relative absolute error                 34.2093 %
Root relative squared error             63.5394 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.878    0.144    0.837      0.878    0.857      0.732    0.930     0.919     0
                 0.856    0.122    0.892      0.856    0.874      0.732    0.930     0.925     1
Weighted Avg.    0.866    0.132    0.867      0.866    0.866      0.732    0.930     0.922

=== Confusion Matrix ===

  a  b   <-- classified as
 72 10 |  a = 0
 14 83 |  b = 1
```

Random Forest



```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         135               75.419  %
Incorrectly Classified Instances        44               24.581  %
Kappa statistic                          0.4973
Mean absolute error                      0.2869
Root mean squared error                  0.3853
Relative absolute error                 57.6308 %
Root relative squared error             77.2926 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.634    0.144    0.788      0.634    0.703      0.506    0.866     0.828     0
                 0.856    0.366    0.735      0.856    0.790      0.506    0.866     0.893     1
Weighted Avg.    0.754    0.264    0.759      0.754    0.750      0.506    0.866     0.863

=== Confusion Matrix ===

  a  b   <-- classified as
 52 30 |  a = 0
 14 83 |  b = 1
```

Naive Bayes



```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         148               82.6816 %
Incorrectly Classified Instances        31               17.3184 %
Kappa statistic                          0.6541
Mean absolute error                      0.2487
Root mean squared error                  0.3694
Relative absolute error                 49.9593 %
Root relative squared error             74.1037 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.866    0.206    0.780      0.866    0.821      0.657    0.869     0.821     0
                 0.794    0.134    0.875      0.794    0.832      0.657    0.869     0.848     1
Weighted Avg.    0.827    0.167    0.832      0.827    0.827      0.657    0.869     0.836

=== Confusion Matrix ===

  a  b   <-- classified as
 71 11 |  a = 0
 20 77 |  b = 1
```

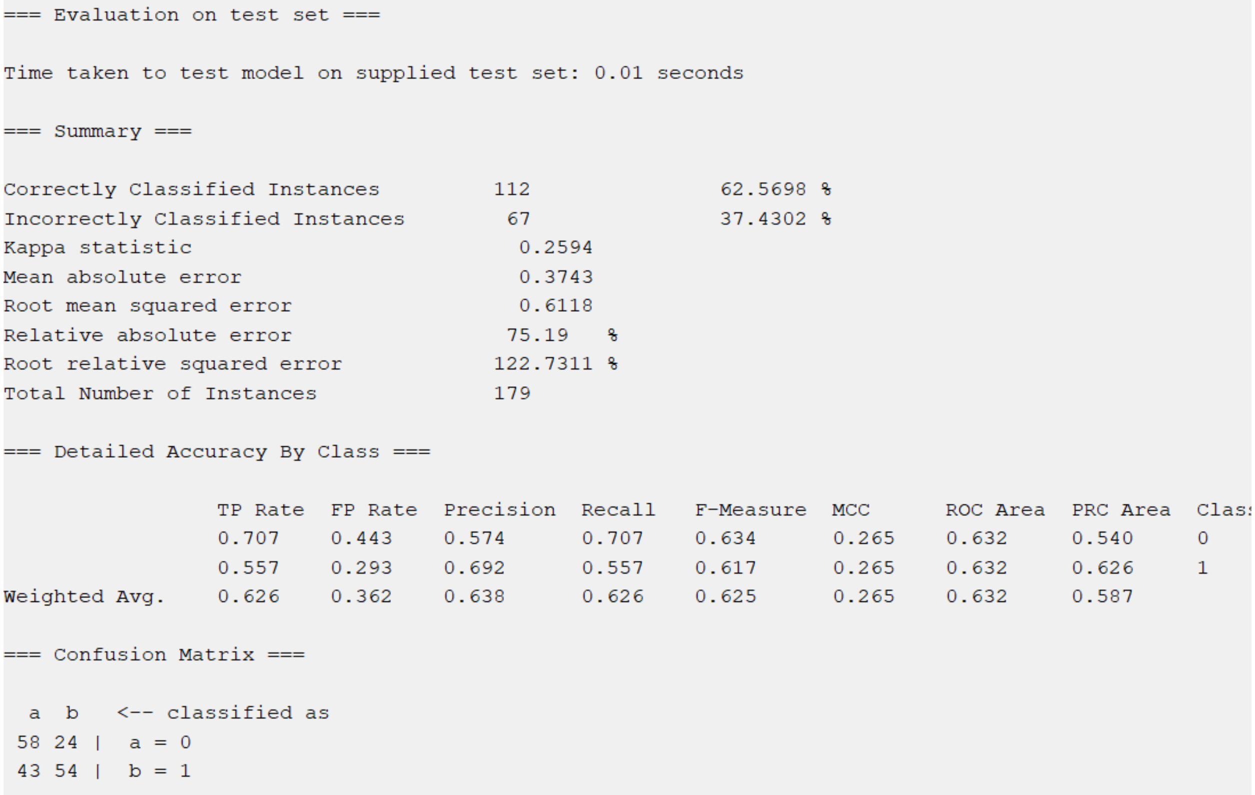
J48

Summary Table for Dataset 4:

	Performance Metric					
Classifier Used		Accuracy	TP Rate (False/True)	FP Rate (False/True	AUC	F-Measure (False/True)
	OneR	.765	.683 .835	.165 .317	.759	.727 .794
	Random Forest	.866	.878 .856	.144 .122	.930	.857 .874
	Naive Bayes	.754	.634 .856	.144 .366	.866	.703 .790
	J48	.827	.866 .794	.206 .134	.869	.821 .832

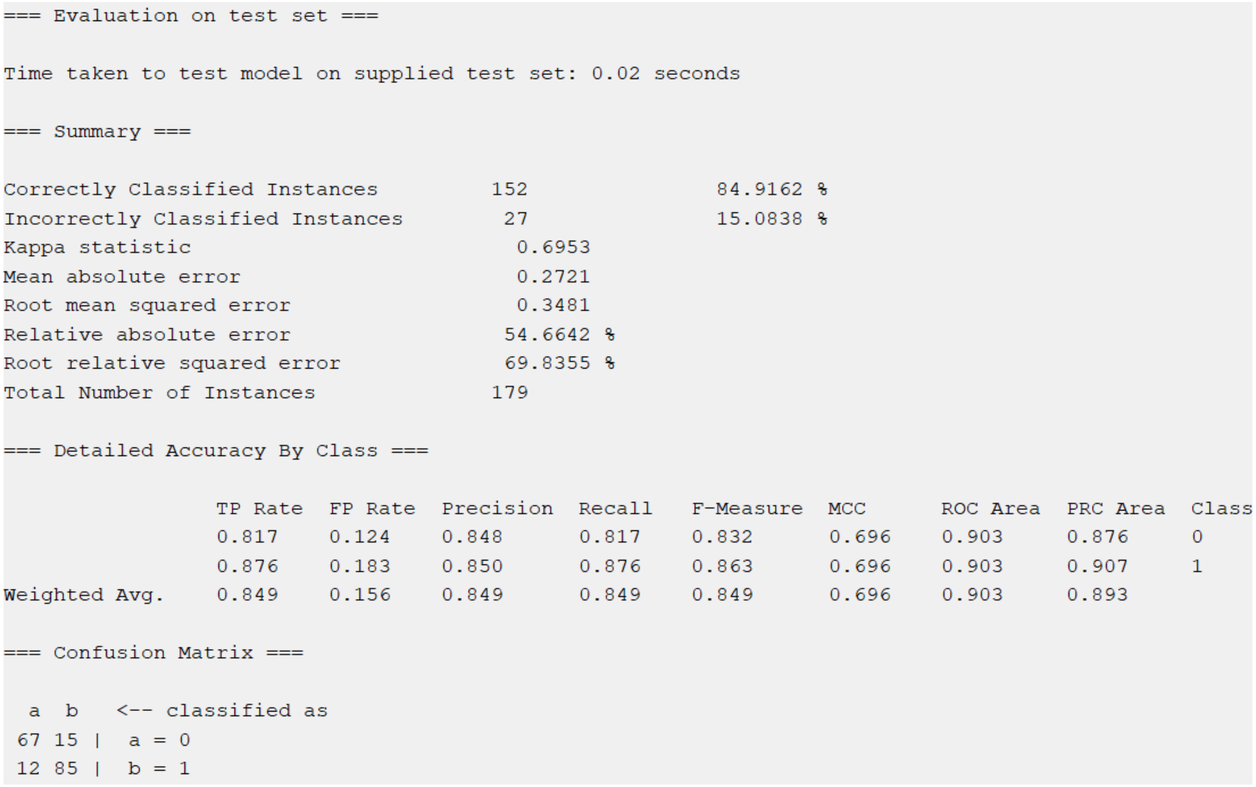
Dataset 5:

Here are the results for our fifth dataset, which contained the attributes: resting ecg, age, resting bp s, cholesterol, and max heart rate.



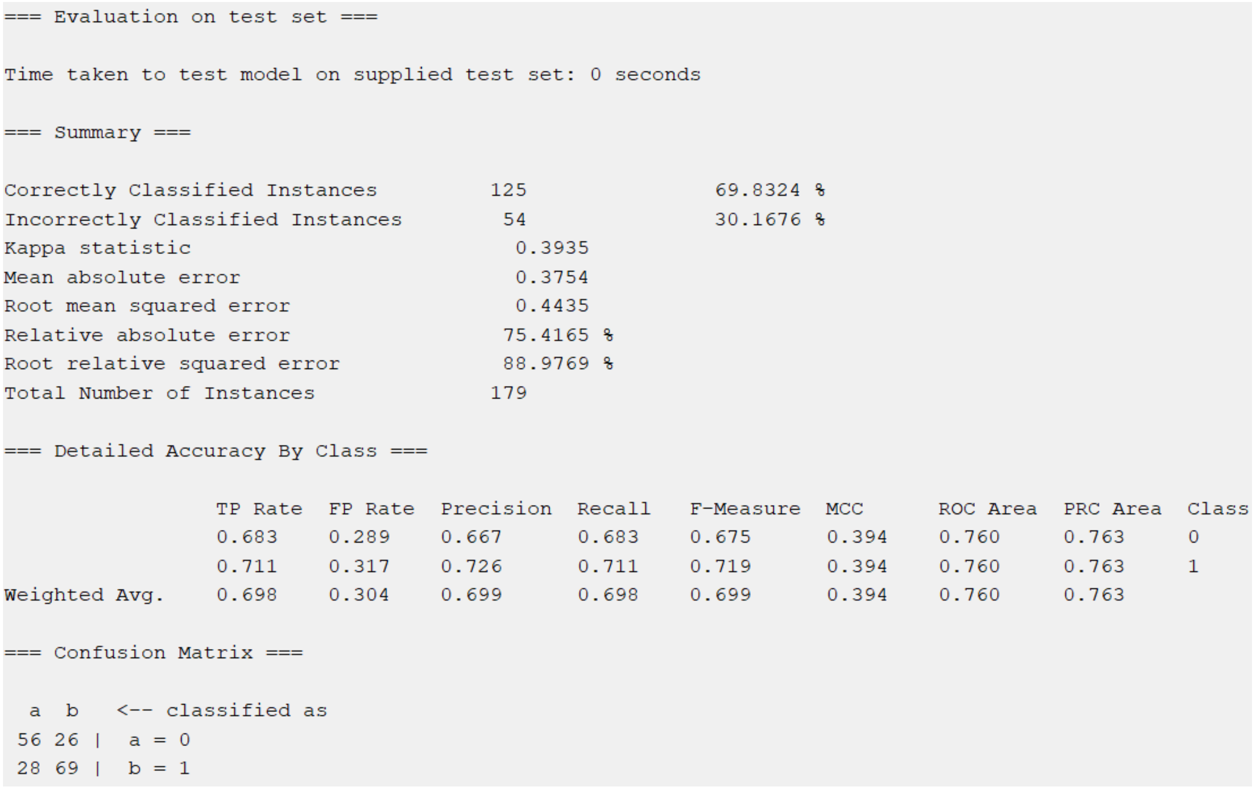
```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         112               62.5698 %
Incorrectly Classified Instances        67               37.4302 %
Kappa statistic                          0.2594
Mean absolute error                      0.3743
Root mean squared error                  0.6118
Relative absolute error                 75.19   %
Root relative squared error            122.7311 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Clas
                 0.707    0.443    0.574      0.707    0.634      0.265    0.632     0.540     0
                 0.557    0.293    0.692      0.557    0.617      0.265    0.632     0.626     1
Weighted Avg.    0.626    0.362    0.638      0.626    0.625      0.265    0.632     0.587

=== Confusion Matrix ===

  a  b   <-- classified as
 58 24 |  a = 0
 43 54 |  b = 1
```

OneR



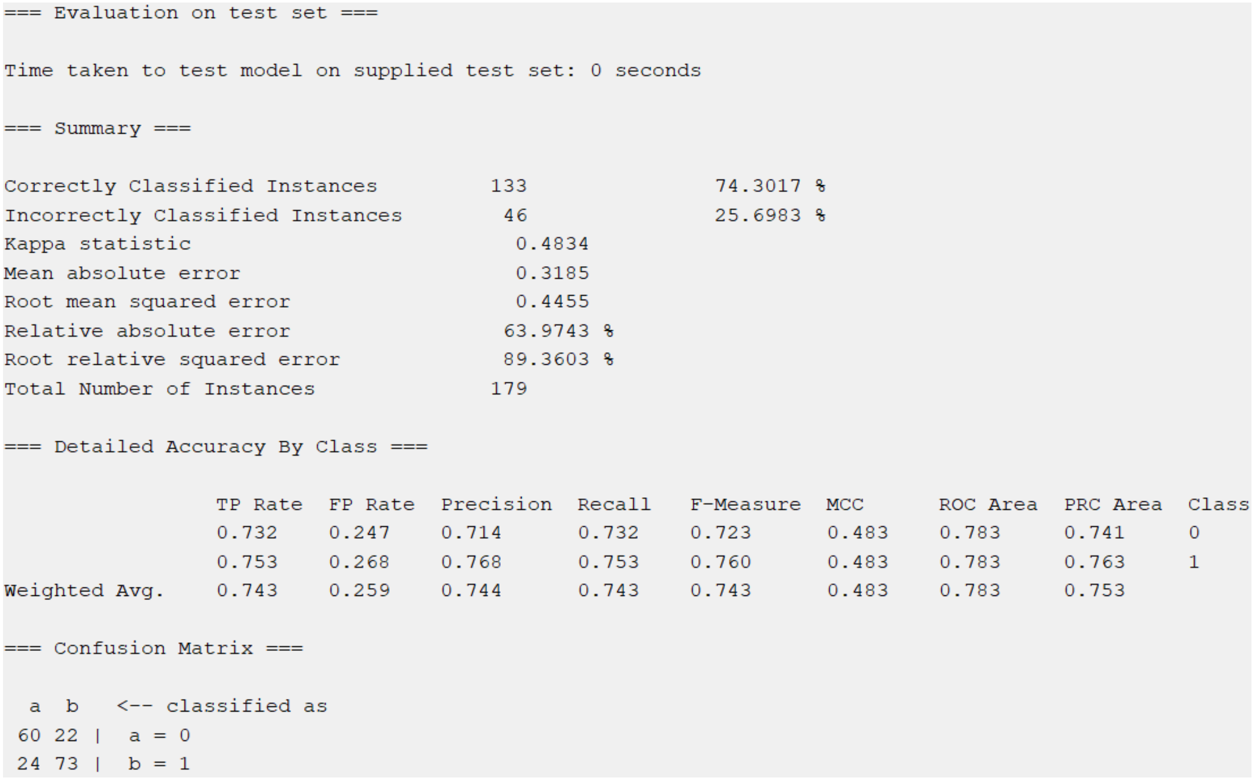
```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.02 seconds

=== Summary ===

Correctly Classified Instances         152               84.9162 %
Incorrectly Classified Instances        27               15.0838 %
Kappa statistic                          0.6953
Mean absolute error                      0.2721
Root mean squared error                  0.3481
Relative absolute error                 54.6642 %
Root relative squared error             69.8355 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.817    0.124    0.848      0.817    0.832      0.696    0.903     0.876     0
                 0.876    0.183    0.850      0.876    0.863      0.696    0.903     0.907     1
Weighted Avg.    0.849    0.156    0.849      0.849    0.849      0.696    0.903     0.893

=== Confusion Matrix ===

  a  b   <-- classified as
 67 15 |  a = 0
 12 85 |  b = 1
```

Random Forest



```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         125               69.8324 %
Incorrectly Classified Instances        54               30.1676 %
Kappa statistic                          0.3935
Mean absolute error                      0.3754
Root mean squared error                  0.4435
Relative absolute error                 75.4165 %
Root relative squared error             88.9769 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.683    0.289    0.667      0.683    0.675      0.394    0.760     0.763     0
                 0.711    0.317    0.726      0.711    0.719      0.394    0.760     0.763     1
Weighted Avg.    0.698    0.304    0.699      0.698    0.699      0.394    0.760     0.763

=== Confusion Matrix ===

  a  b   <-- classified as
 56 26 |  a = 0
 28 69 |  b = 1
```

Naive Bayes



```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         133               74.3017 %
Incorrectly Classified Instances        46               25.6983 %
Kappa statistic                          0.4834
Mean absolute error                      0.3185
Root mean squared error                  0.4455
Relative absolute error                 63.9743 %
Root relative squared error             89.3603 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.732    0.247    0.714      0.732    0.723      0.483    0.783     0.741     0
                 0.753    0.268    0.768      0.753    0.760      0.483    0.783     0.763     1
Weighted Avg.    0.743    0.259    0.744      0.743    0.743      0.483    0.783     0.753

=== Confusion Matrix ===

  a  b   <-- classified as
 60 22 |  a = 0
 24 73 |  b = 1
```

J48

Summary Table for Dataset 5:

	Performance Metric					
Classifier Used		Accuracy	TP Rate (False/True)	FP Rate (False/True)	AUC	F-Measure (False/True)
	OneR	.626	.707 .557	.443 .293	.632	.634 .617
	Random Forest	.849	.817 .876	.124 .183	.903	.832 .863
	Naive Bayes	.698	.683 .711	.289 .317	.760	.675 .719
	J48	.743	.732 .753	.247 .268	.783	.723 .760

Table of AUC for each of the 20 models

		Attribute Selection Algorithm					
Classifier Used		Correlation Analysis	CFS Subset Eval	Relief Attribute Eval	Gain Ratio	Self-Select	AVG
	OneR	.794	.772	.766	.759	.632	.745
	Random Forest	.907	.962	.970	.930	.903	.934
	Naive Bayes	.869	.901	.914	.866	.760	.862
	J48	.870	.882	.878	.869	.783	.856
	AVG	.860	.879	.882	.856	.770	

Part 6 – Analysis

After we finished running all 20 classifier models, using 4 different classifiers on 5 different datasets, each created by a different attribute selection model, we had a some models that were on average, more accurate than others. It seemed to be that any attribute selection algorithm dataset that used the Random Forest Classifier would achieve the highest raw accuracy, with Random Forest on dataset 3, where we used ReliefAttributeEval, achieving an accuracy of around 95%, which was the highest in any of our runs.

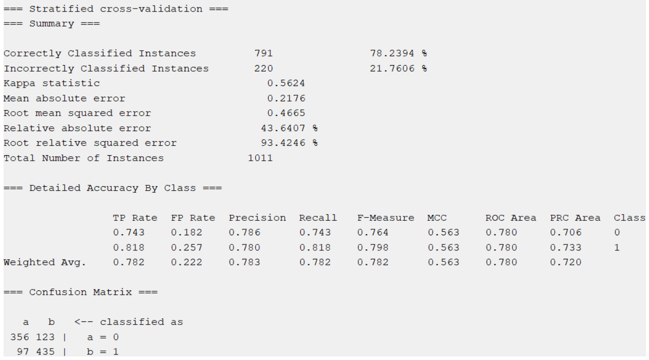
Although it would be easy to simply say that this was our best model and leave it at that. Our main goal was to create a model that can correctly identify heart disease as accurately as possible. This entails looking at TP and FP rate for the class value that we are interested in, which is 1 (patient has heart disease). In all the TP rates for class 1, the highest rate is still by Random Forest on dataset 3, with a rate of 96.9%. This means that the model will correctly identify when a patient has heart disease around 97% of the time. Even though Random Forest far outstrips its competition in this dataset, in dataset 4, where we used gain ratio, the TP rates for Random Forest and Naive Bayes were the same, with both being 85.6%. From this information it is reasonable to assume that Random Forest and Naive Bayes are the 2 best models; however, consistency is a big factor of any model.

In dataset 5, which is the set were we self-selected the attributes, we selected the attributes that were picked the fewest by the other attribute selection algorithms. One would point out that such a strategy might lead to attributes that were suboptimal for modeling this dataset. We also figured this out and decided to use this to our advantage, setting this dataset as a “tiebreaker” if there were multiple models that had similar performance through other datasets. In dataset 5 we can see that the TP rate for Naive Bayes is only 71.1%, while the TP rate for Random Forest is still maintains around 85%. While doing research on the topic we found out that generally, Naive Bayes is found to be unreliable when training data is unreliable. This likely gives the tree models (Random Forest and J48) a leg up when it comes to performance.

Up until now we have only discussed the performance of the models and how they fared through the different datasets. As we showed with dataset 5 (self-selected), the attributes used to create a model are sometimes as important as the model itself, with dataset 5 having on average 10% lower accuracy than dataset 3 (Relief Attribute Eval). Below we have redisplayed the attributes used for each dataset and what attribute selection method we used:

1. (Correlation Analysis) - ST slope, exercise angina, chest pain type, oldpeak, sex, and max heart rate
2. (CFS Subset Eval) - age, sex, chest pain type, cholesterol, max heart rate, exercise angina, oldpeak, ST slope
3. (Relief Attribute Eval) - chest pain type, ST slope, resting ecg, fasting blood sugar, sex, and cholesterol
4. (Gain Ratio) - ST slope, exercise angina, chest pain type, oldpeak, and sex
5. (Self-Selected) - ecg, age, resting bp s, cholesterol, and max heart rate

As you can see, there is one attribute that is selected by all the attribute selection algorithms that we performed and is a part of every dataset, except the one that we self-selected. That attribute is ST slope, which is the slope of your heart rate while doing exercise. The fact that this attribute is selected in all 4 datasets must mean that it has some relationship with the class. To test this relationship, we conducted a trial using the OneR model, where the only attribute was ST slope. Here are the results of the 10 fold cross-validation test that we conducted. As you can see, the model reports an accuracy of 78.2%, which is extremely similar to the OneR results of all of the datasets where ST-slope is present, where all accuracies are from 75% to 80%. This led us to conclude that for datasets 1-4, the model is using ST-Slope as the one attribute.



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         791               78.2394 %
Incorrectly Classified Instances       220               21.7606 %
Kappa statistic                          0.5624
Mean absolute error                      0.2176
Root mean squared error                  0.4665
Relative absolute error                 43.6407 %
Root relative squared error             93.4246 %
Total Number of Instances             1011

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.743    0.182    0.786      0.743    0.764      0.563    0.780     0.706     0
                 0.818    0.257    0.780      0.818    0.798      0.563    0.780     0.733     1
Weighted Avg.    0.782    0.222    0.783      0.782    0.782      0.563    0.780     0.720

=== Confusion Matrix ===

   a   b   <-- classified as
 356 123 |   a = 0
  97 435 |   b = 1
```

This makes sense as while doing research on the attributes, we found that ST slope has been shown by many studies to be heavily correlated with pulmonary heart disease would likely explain why it is such a good predictor for heart disease. Although on a method-wise scale, it does not help us decide which is best since it is in pretty much all of them. For that, we can look at the summary table for all 20 models that we ran. The title for best method appears to be a close battle between method #2 (CFS Subset Eval) and method #3 (Relief Attribute Eval). Although difference in AUC might be differing by only .003, our discussion about the OneR model on this dataset might give us a clear answer. As we concluded earlier, the OneR model's ruleset consisted of the ST-Slope Attribute, which was the same in datasets (1-4), this means that any differences in the OneR results can be attributed to random chance and can likely be disregarded. With this knowledge we can come to the conclusion that method #3 performs better than method #2 for 2 models, Random Forest and Naive Bayes, which are not 2 regular models but our two best performing models, while method #2 performs better for 1 model, J48. With this we can conclude that method #3, Relief Attribute Eval, was the best attribute selection method and the best model for that is Random Forest, which we determined earlier.

Part 7 – Conclusion & Steps to Reproduce

7.1 – Conclusion

As we stated above in the "Analysis" section, our best model was Relief Attribute Eval with Random Forest, a model that produced a ROC curve area (AUC) of .970, our highest score with our most consistent model. Using weka we were still able to still create a model that could correctly diagnose a patient that had heart disease (TP rate for class value 1) 96.9% of the time, which is a pretty good level of precision. Although all of our classifiers, including OneR, achieved an average accuracy of over 75% in a balanced dataset, we feel as there is still more room to improve. Being able to use only weka, we were sort of handcuffed to the classifiers that weka had to offer. While most of those classifiers would be fine to use to get >95% accuracy, they are still surface level, low-complexity algorithms. If we had more time and computing power, our approach to solve this problem would be to use a deep neural network, as neural networks tend to work much better than decision trees when a large portion of data is continuous, as are about 50% of our datasets attributes.

7.2 – Steps to Reproduce

To reproduce the results of our best model that used Random Forest with Relief Attribute Eval:

1. Download the **arf copies** folder in the datasets folder of our project folder
2. Open **arfdf3train.arff** in Weka
3. Go to the **Classify** tab in Weka
4. Under **Test Options** select **Supplied Test Set**
5. For the test set use **arfdf3test.arff**, it should be in the same folder
6. Make sure the class is set to **(Nom) Target**
7. Under **Classifer** click **Choose > classifiers > trees > RandomForest** and select it
8. Now hit start and the results should appear

Part 8 - Team Members and Tasks Performed

Finding the Data & Building Proposal: Adarsh
Preprocessing Initial Attempt: Adarsh
Preprocessing Final Attempt: Aryan
Non-Weka Attribute Selection Algorithm: Aryan
Attribute Selection Algorithms: Aryan
Attribute Selection Classifiers: Adarsh
Results Output: Adarsh
Results Analysis: Aryan
Building Final Report: Adarsh & Aryan

Part 9 - Appendix & Sources

Data Source Website:
https://www.kaggle.com/datasets/mexwell/heart-disease-dataset

Files Attached in Google Drive:
arf copies - Pre-processed files used to create models
original csv files - Original data from the source
make_sets.py - Code to split data into 5 datasets with train and test

Sources:

Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten (2009). The WEKA Data Mining Software: An Update. SIGKDD Explorations, Volume 11, Issue 1.