Final_Project_Q2_JL

December 9, 2020

Jasper Lemberg
Professor Brendan Mort
DSCC 201
9 December 2020
Final Project - Question 2

2. Now we will repeat question 1, but using R and a different model. Use R version 3.5.1/b1. Using the R console, the R kernel of a Jupyter notebook, or R Studio, perform the following tasks: (50 points)

A. Import the /public/bmort/python/heartdisease.csv file into a data frame in R.

```
In [1]: heart_disease <- read.csv("/public/bmort/python/heartdisease.csv")
        head(heart_disease)
```

A data.frame: 6 Œ 14

age <int>	sex <int>	pain <int>	bp <int>	chol <int>	sugar <int>	ecg <int>	rate <int>	angina <int>	stv <dbl>	sts <int>
63	1	1	145	233	1	2	150	0	2.3	3
67	1	4	160	286	0	2	108	1	1.5	2
67	1	4	120	229	0	2	129	1	2.6	2
37	1	3	130	250	0	0	187	0	3.5	3
41	0	2	130	204	0	2	172	0	1.4	1
56	1	2	120	236	0	0	178	0	0.8	1

B. Is there any missing data in the data frame? What is missing? Perform any necessary data imputation before proceeding. Explain the reason behind the choice made.

```
In [2]: sum(is.na(heart_disease))
```

1

There is one missing datum in the data frame, which I know is in the rate column from question 1B. To fix this, I imputed the column's median in that spot, as shown by the code below.

```
In [3]: heart_disease[is.na(heart_disease)] <- median(heart_disease$rate, na.rm = T)
        sum(is.na(heart_disease))
```

0

C. Check the summary statistics on the data. How do the ranges of the values in the columns compare? Does each column of data have similar magnitudes and ranges?

In [4]:
```
summary(heart_disease)
```

```
      age             sex             pain             bp              chol      
 Min.   :29.00   Min.   :0.00   Min.   :1.000   Min.   : 94.0   Min.   :126.0  
 1st Qu.:48.00   1st Qu.:0.00   1st Qu.:3.000   1st Qu.:120.0   1st Qu.:211.0  
 Median :56.00   Median :1.00   Median :3.000   Median :130.0   Median :241.5  
 Mean   :54.48   Mean   :0.68   Mean   :3.153   Mean   :131.6   Mean   :246.9  
 3rd Qu.:61.00   3rd Qu.:1.00   3rd Qu.:4.000   3rd Qu.:140.0   3rd Qu.:275.2  
 Max.   :77.00   Max.   :1.00   Max.   :4.000   Max.   :200.0   Max.   :564.0  
     sugar             ecg              rate           angina      
 Min.   :0.0000   Min.   :0.0000   Min.   : 71.0   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:135.5   1st Qu.:0.0000  
 Median :0.0000   Median :0.5000   Median :153.0   Median :0.0000  
 Mean   :0.1467   Mean   :0.9867   Mean   :149.8   Mean   :0.3267  
 3rd Qu.:0.0000   3rd Qu.:2.0000   3rd Qu.:166.0   3rd Qu.:1.0000  
 Max.   :1.0000   Max.   :2.0000   Max.   :202.0   Max.   :1.0000  
      stv              sts              mvn              thal          disease      
 Min.   :0.00   Min.   :1.000   Min.   :0.00   Min.   :3.000   Min.   :0.00  
 1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.00   1st Qu.:3.000   1st Qu.:0.00  
 Median :0.80   Median :2.000   Median :0.00   Median :3.000   Median :0.00  
 Mean   :1.05   Mean   :1.603   Mean   :0.67   Mean   :4.727   Mean   :0.46  
 3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.00   3rd Qu.:7.000   3rd Qu.:1.00  
 Max.   :6.20   Max.   :3.000   Max.   :3.00   Max.   :7.000   Max.   :1.00  
```

Age, blood pressure, cholestrol, heart rate, and stv all have ranges over 3 and standard deviations over 1. Therefore, these columns are not all on the same range.

D. Partition your data into a training set (80%) and a testing set (20%) that is randomly selected from the heartdisease.csv data. Hint: Use the caret library's createDataPartition() function.

In [6]:
```
library(caret)
```

```
dp <- createDataPartition(heart_disease$disease, p = 0.8, list = F)
training_set <- heart_disease[dp,]
testing_set <- heart_disease[-dp,]
```

E. Use the support vector machine (SVM) method with a linear basis function kernel from R's caret library to develop a model to predict heart disease diagnosis based on the 13 features provided in the data set for each patient. Make sure to use the caret library's built-in repeated cross-validation capability for training your model. You will need to define a training control object using caret's trainControl() function. The training of the model is called by the train() function of the caret library.

```
In [7]: tc <- trainControl(method = "cv", number = 5)
        model <- train(as.factor(disease) ~ ., data =  training_set,
                       method = "svmLinear", trControl = tc)
```

F. Generate a confusion matrix using the data from your test set to show the accuracy of the model using the confusionMatrix() function. Comment on the accuracy of the model. What percent are false positives? What percent are false negatives? How does the SVM model compare with the logistic regression model generated in Question 1?

```
In [8]: test_prediction <- predict(model, newdata = testing_set)
        confusionMatrix(as.factor(test_prediction), as.factor(testing_set$disease))
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 28  8
         1  4 20

               Accuracy : 0.8
                 95% CI : (0.6767, 0.8922)
    No Information Rate : 0.5333
    P-Value [Acc > NIR] : 1.609e-05

                  Kappa : 0.5946

 Mcnemar's Test P-Value : 0.3865

            Sensitivity : 0.8750
            Specificity : 0.7143
         Pos Pred Value : 0.7778
         Neg Pred Value : 0.8333
             Prevalence : 0.5333
         Detection Rate : 0.4667
   Detection Prevalence : 0.6000
      Balanced Accuracy : 0.7946

       'Positive' Class : 0
```

Of all the tests, 13.3% are false positives and 6.7% are false negatives. That being said, the model has 80% accuracy, so it is a fairly good model and just as good as the logistic regression model from question 1, despite the lack of standardization.