Heart Disease Analysis

Sathvika Chakka and Ishika Mutti

BUS2-193 Sec-01

December 6, 2022

Abstract

Our objective through this project is to conduct survival analysis of heart failure patients. In other words, we hope to find patterns and/or correlations to predict death events with respect to either some or all variables in the dataset. The goal is to achieve this by using data mining tools like classification, logistic regression, and random forest.

Background Information

Cardiovascular diseases kill about 17.9 million people globally. These diseases are the cause of heart failures. To be more specific, heart failure occurs when the heart is not able to pump enough blood for the body. For years, medical professionals have been trying to figure out what causes cardiovascular diseases. Of course, there are many factors that come into play and they differ for each and every person. In this analysis, we will be diving into different factors that can contribute to heart failure to see which ones are the most prominent. All the participants are patients “who were admitted to Institute of Cardiology and Allied hospital Faisalabad-Pakistan during April-December (2015).

Detailed Data Source

The data source we have chosen to conduct our analysis and project on is a heart failure data set from the University of California, Irvine machine learning repository site. This dataset consists of 13 attributes, 299 instances, as well as integer and real characteristics. The 299 patient records, comprising 105 women and 194 men, in our dataset are from people who all had heart failure with different clinical features that were measured. These clinical features (and independent variables) were age, anemia, high blood pressure, creatinine phosphokinase, diabetes, ejection fraction, platelets, sex, serum creatinine, serum sodium, smoking, and follow-up period of the patient sampled. Some of these independent variables are continuous integers, booleans, and binary. With these independent variables, we also have a response variable which will be the death event (if the patient deceased during the follow up period).

Preliminary Analysis

```
> str(clinic_data)
'data.frame':   299 obs. of  13 variables:
 $ age                     : num  75 55 65 50 65 90 75 60 65 80 ...
 $ anaemia                 : int  0 0 0 1 1 1 1 1 0 1 ...
 $ creatinine_phosphokinase: int  582 7861 146 111 160 47 246 315 157 123 ...
 $ diabetes                : int  0 0 0 0 1 0 0 1 0 0 ...
 $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 35 ...
 $ high_blood_pressure     : int  1 0 0 0 0 1 0 0 0 1 ...
 $ platelets               : num  265000 263358 162000 210000 327000 ...
 $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
 $ serum_sodium            : int  130 136 129 137 116 132 137 131 138 133 ...
 $ sex                     : int  1 1 1 1 0 1 1 1 0 1 ...
 $ smoking                 : int  0 0 1 0 0 1 0 1 0 1 ...
 $ time                    : int  4 6 7 7 8 8 10 10 10 10 ...
 $ DEATH_EVENT             : int  1 1 1 1 1 1 1 1 1 1 ...
```

Figure 1

Out of 13 variables, 10 variables have the data structured as integers and 3 of them are structured as numeric data points.

```
> summary(clinic_data)
      age            anaemia       creatinine_phosphokinase    diabetes      ejection_fraction high_blood_pressure
 Min.   :40.00   Min.   :0.0000   Min.   :  23.0           Min.   :0.0000   Min.   :14.00     Min.   :0.0000
 1st Qu.:51.00   1st Qu.:0.0000   1st Qu.: 116.5           1st Qu.:0.0000   1st Qu.:30.00     1st Qu.:0.0000
 Median :60.00   Median :0.0000   Median : 250.0           Median :0.0000   Median :38.00     Median :0.0000
 Mean   :60.83   Mean   :0.4314   Mean   : 581.8           Mean   :0.4181   Mean   :38.08     Mean   :0.3512
 3rd Qu.:70.00   3rd Qu.:1.0000   3rd Qu.: 582.0           3rd Qu.:1.0000   3rd Qu.:45.00     3rd Qu.:1.0000
 Max.   :95.00   Max.   :1.0000   Max.   :7861.0           Max.   :1.0000   Max.   :80.00     Max.   :1.0000
   platelets      serum_creatinine  serum_sodium       sex            smoking           time        DEATH_EVENT
 Min.   : 25100   Min.   :0.500    Min.   :113.0   Min.   :0.0000   Min.   :0.0000   Min.   :  4.0   Min.   :0.0000
 1st Qu.:212500   1st Qu.:0.900    1st Qu.:134.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 73.0   1st Qu.:0.0000
 Median :262000   Median :1.100    Median :137.0   Median :1.0000   Median :0.0000   Median :115.0   Median :0.0000
 Mean   :263358   Mean   :1.394    Mean   :136.6   Mean   :0.6488   Mean   :0.3211   Mean   :130.3   Mean   :0.3211
 3rd Qu.:303500   3rd Qu.:1.400    3rd Qu.:140.0   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:203.0   3rd Qu.:1.0000
 Max.   :850000   Max.   :9.400    Max.   :148.0   Max.   :1.0000   Max.   :1.0000   Max.   :285.0   Max.   :1.0000
```

Figure 2

Judging by the summary statistics in Figure 2, we can see that some of the independent variables, like anaemia, diabetes, sex and smoking, have binary data points. Among the binary independent variables, some of the variables, like anemia, smoking, and diabetes, were initially booleans that were converted into binary values 1 and 0. There are also some independent variables with data structured as integers. The data points of the response variables, DEATH_EVENTS, consist of binary variables.

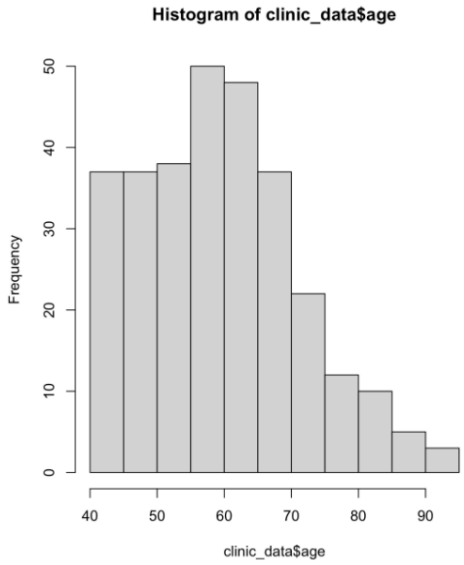


Figure 3

The histogram in Figure 3 shows us the age distribution of the participants in the study. We can see that all participant patients are 40 years old and above. We can see that a large portion of the participation falls under the 60-70 years range.

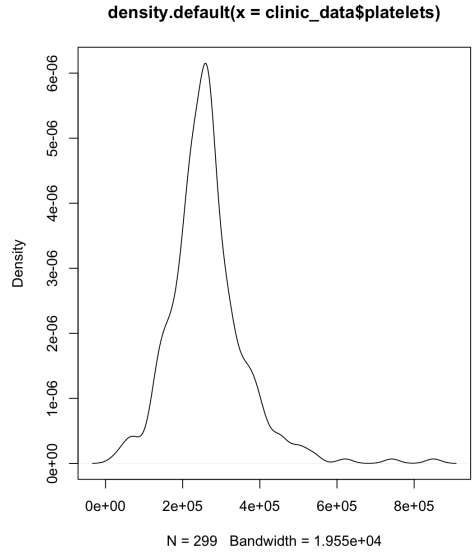


Figure 4

Figure 4 is a density plot graph for the platelets variable. The visual summarizes the overall platelet count of the participant patients. The peak of the density plot is the location where there is the highest concentration of datapoints of the patients.

Results and Conclusion

After comparing confusion matrices, cost functions, and AUC values, we got results supporting that the logistic regression model was the best model of all the three models. These 7 independent variables age, creatine_phosphokinase, ejection_fraction, serum_creatinine, serum_sodium, sex, and time have the most effect on our response variable DEATH_EVENT.

Logistic Regression

Since our dataset has a binary response variable for the DEATH_EVENT, we conducted logistic regression to find the best model.

Model Selection

```
> AIC(model_1)
[1] 199.8791
>
>
> model_2 <- glm(DEATH_EVENT~ age + anaemia + creatinine_phosphokinase + diabetes +
+                  ejection_fraction + high_blood_pressure + serum_creatinine +
+                  serum_sodium + sex + smoking + time, family = binomial, data = clinic_train)
> AIC(model_2)
[1] 197.8808
>
>
> model_3 <- glm(DEATH_EVENT~ age + creatinine_phosphokinase + diabetes + ejection_fraction +
+                  high_blood_pressure + serum_creatinine + serum_sodium + sex +
+                  smoking + time, family = binomial, data = clinic_train)
> AIC(model_3)
[1] 195.8874
>
>
> model_4 <- glm(DEATH_EVENT~ age + creatinine_phosphokinase + diabetes + ejection_fraction +
+                serum_creatinine + serum_sodium + sex + smoking + time, family = binomial, data = clinic
_train)
> AIC(model_4)
[1] 193.9209
>
>
> model_5 <- glm(DEATH_EVENT~ age + creatinine_phosphokinase + diabetes + ejection_fraction +
+                  serum_creatinine + serum_sodium + sex + time, family = binomial, data = clinic_train)
> AIC(model_5)
[1] 191.9718
>
>
> model_6<- glm(DEATH_EVENT~ age + creatinine_phosphokinase + ejection_fraction +
+                  serum_creatinine + serum_sodium + sex + time, family = binomial, data = clinic_train)
> AIC(model_6)
[1] 190.1518
```

Figure 5

To start the process of finding the best model, we used the backward selection method which spit out six different models. After finding the AIC of each of the six models, we found that model_6 was the best model out of all as it had the lowest AIC value of 190.15. Model_6 consisted of seven independent variables which were age, creatine_phosphokinase, ejection_fraction, serum_creatinine, serum_sodium, sex, and time, which means that these variables had the most influence when predicting the death events of the patient.

ROC and AUC Value (In Sample)

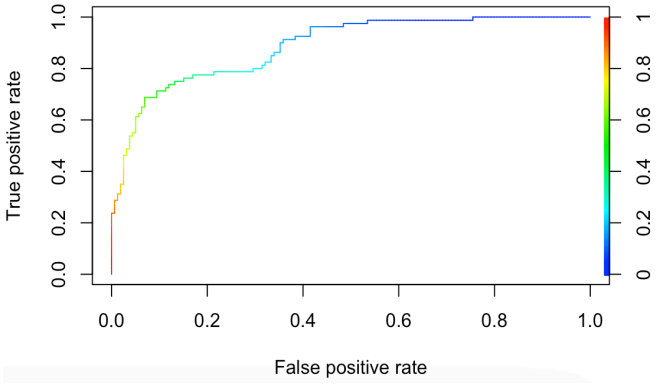


Figure 6

```
> unlist(slot(performance(pred, "auc"), "y.values"))
[1] 0.8865566
```

Figure 7

To further test our model, we conducted an in-sample ROC curve and AUC value and we came to find that the AUC value of model_6 is .89. Since this value is much above .7, we can conclude that the model does an excellent job of classifying the observational data points.

Misclassification Table and Asymmetric Cost (In Sample)

```
> table(clinic_train$DEATH_EVENT, (pred_resp > 0.5)*1, dnn=c("Truth", "Predicted"))
     Predicted
Truth   0   1
    0 144  15
    1  23  57
> table(clinic_train$DEATH_EVENT, (pred_resp > 0.2)*1, dnn=c("Truth", "Predicted"))
     Predicted
Truth   0   1
    0 105  54
    1  11  69
```

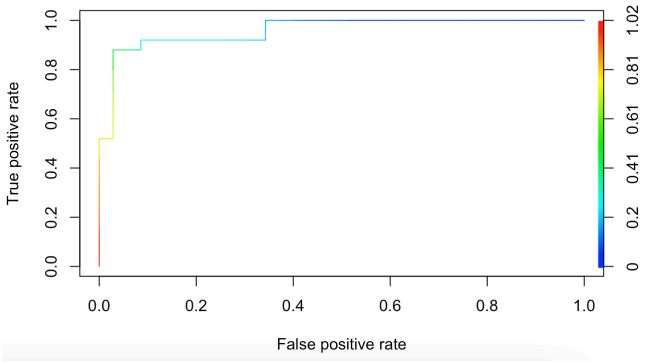
Figure 8

```
> cost2(r = clinic_train$DEATH_EVENT, pi = pred_glm0_train, pcut)
[1] 0.2552301
```

Figure 9

To dive deeper into the patient outcomes of the model, we created the in sample misclassification table and rate. We decided to give more weightage to the false negative misclassification. It is more risky when a patient is falsely diagnosed to be fine, when in truth they have a high risk of having a heart disease that could threaten their life. If a patient is diagnosed with a heart disease but in truth they are in good shape, there is much less risk. To better understand the data, we decided to create a misclassification table using two cutoff ratios 0.5 and 0.2. As you can see from the table above, in the first matrix with a cutoff of 0.5, there are 23 false negatives patients and 15 false positive patients. Whereas, in the second matrix with a cutoff of 0.2, there are only 11 false negatives but 54 false positives.

ROC and AUC Value (Out of Sample)




```
> unlist(slot(performance(pred, "auc"), "y.values"))
[1] 0.9588571
```

Figure 10

After conducting the ROC curve for out of sample, we got an AUC value of .96 which was much higher than out in sample value. Out of sample testing is spitting out a better result than in sample.

Misclassification Matrix and Asymmetric Cost (Out of Sample)

```
> table(clinic_test$DEATH_EVENT, (pred_resp > 0.5)*1, dnn=c("Truth", "Predicted"))
     Predicted
Truth  0  1
    0 34  1
    1  6 19
> table(clinic_test$DEATH_EVENT, (pred_resp > 0.2)*1, dnn=c("Truth", "Predicted"))
     Predicted
Truth  0  1
    0 24 11
    1  2 23
```

Figure 11

```
> cost4(r = clinic_test$DEATH_EVENT, pi = pred_glm0_test, pcut)
[1] 0.2166667
```

Figure 12

The matrix tables in Figure 11 are from our out of sample predictions with the testing data. From the first out of sample misclassification matrix with a cutoff of 0.5, we see that there are 6 false negative patients and 1 false positive patient. In the second matrix that has a cutoff value of 0.2, we see 2 false negative patients and 11 false positive patients. The cost function for out of sample testing comes out to be .22 which is lower than in sample. Overall, our out of sample testing had better results than our in sample which is a good sign since out of sample is our prediction.

Classification tree

To find the best model we decided to build classification trees, a decision tree, since our response is binary.

Model Selection

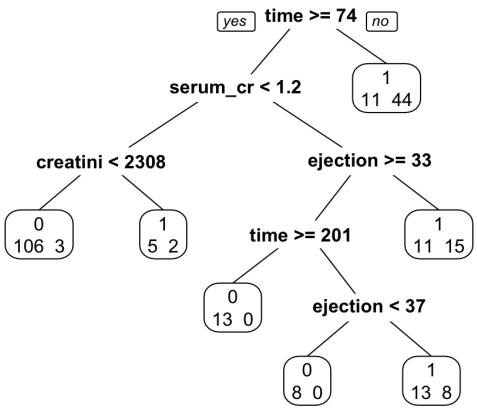


Figure 13

Figure 13 is a classification tree for the training dataset that contains 80% of the original data. A classification tree is composed of branches that represent attributes (dependent variables), while the leaves represent the decisions. The tree shows the conditions the independent variable

(DEATH_EVENT) has to follow in order to get the best model. As we see, it starts off with the attribute time and depending on the outputs, it'll go down a pathway that would lead to the best model.

```
n= 239

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 239 167 1 (0.69874477 0.30125523)
   2) time>=73.5 184 140 0 (0.84782609 0.15217391)
     4) serum_creatinine< 1.19 116  25 0 (0.95689655 0.04310345)
       8) creatinine_phosphokinase< 2307.5 109  15 0 (0.97247706 0.02752294) *
       9) creatinine_phosphokinase>=2307.5 7   5 1 (0.71428571 0.28571429) *
     5) serum_creatinine>=1.19 68  45 1 (0.66176471 0.33823529)
      10) ejection_fraction>=32.5 42  34 1 (0.80952381 0.19047619)
        20) time>=200.5 13   0 0 (1.00000000 0.00000000) *
        21) time< 200.5 29  21 1 (0.72413793 0.27586207)
          42) ejection_fraction< 36.5 8   0 0 (1.00000000 0.00000000) *
          43) ejection_fraction>=36.5 21  13 1 (0.61904762 0.38095238) *
      11) ejection_fraction< 32.5 26  11 1 (0.42307692 0.57692308) *
   3) time< 73.5 55  11 1 (0.20000000 0.80000000) *
```

Figure 14

To further dive into our analysis, we decide to look into the performance of the confusion matrix with asymmetric cost for in-sample and out-of-sample and also their cost function. For the matrix we used a weightage of 5:1 for false negative and false positive respectively and the rest as 0.

Confusion Matrix and Asymmetric Cost (In Sample)

```
> table(clinic_train$DEATH_EVENT, clinic_train.pred.tree1, dnn=c("Truth","Predicted")) # table the matrix
     Predicted
Truth   0   1
    0 127  40
    1   3  69
```

Figure 15

```
> cost(clinic_train$DEATH_EVENT, predict(clinic_rpart, clinic_train, type="prob")) #training dataset # asymmetric
[1] 0.4644351
```

Figure 16

For our in-sample performance (training dataset), from Figure 15, we can see that there are 40 false positives and 3 false negatives. In this case, having less false negatives is a good sign. For the cost, also, we used a weightage of 5:1 and we got the cost of 0.46.

Confusion Matrix and Asymmetric Cost (Out of Sample)

```
> table(clinic_test$DEATH_EVENT, clinic_test.pred.tree1, dnn=c("Truth","Predicted"))
     Predicted
Truth  0  1
    0 27  9
    1  2 22
```

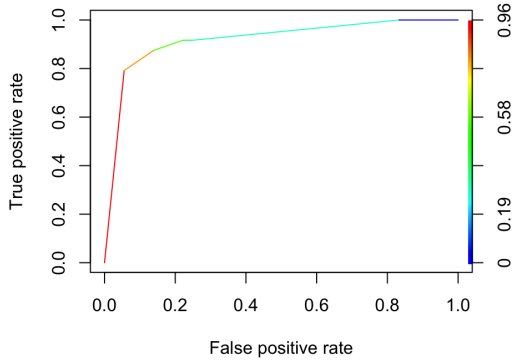
Figure 17

```
> cost(clinic_test$DEATH_EVENT, predict(clinic_rpart, clinic_test, type="prob")) #testing dataset # asymmetric
[1] 0.4583333
```

Figure 18

As for our out-of-sample performance (testing dataset), as you can in Figure 17, we got 2 false negatives and 9 false positives, which again is a good sign. To add on, we got a cost of 0.46 which is close to the cost of the training dataset.

ROC Curve and AUC Value (Out of Sample)




```
> slot(performance(pred, "auc"), "y.values")[[1]]
[1] 0.9172454
```

Figure 19

We wanted to look into how well our binary classification model is performing irrespective of the cost function. So we built the ROC curve on the testing dataset and found the AUC value. This basically tells us how well the performance was on an unknown dataset. The graph is a visual of the ROC curve which is based on the testing dataset. We got an AUC value of .92 which can be considered outstanding for classifying the observational data points.

Random Forest

We chose random forest as the method we wanted to explore since it allows us to further breakdown the tree we created in classification to create more trees. From those trees, we can find the best one and compare them to the logistic regression and classification models we did prior to this. To find the model, we used the randomForest function and input the rest of the arguments.

Confusion Matrix (In Sample)

```
Call:
 randomForest(formula = DEATH_EVENT ~ ., data = clinic_train,      mtry = 12, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 12

        OOB estimate of  error rate: 18.83%
Confusion matrix:
    0  1 class.error
0 147 21   0.1250000
1  24 47   0.3380282
```

Figure 20

The in-sample model came out to be classification (which we expected) with a total of 500 trees and 12 variables tried at each split. From the output above, there was a 18.83% out of bag error rate, meaning that there is a 18.83% discriminate between the different random forest tree classifiers. It also calculated a confusion matrix in which we can further interpret the

misclassification of the data points. There are 3 more false negatives than those of the false positives, which as mentioned before, is concerning as potential heart issues are being ignored.

Confusion Matrix and Asymmetric Cost (Out of Sample)

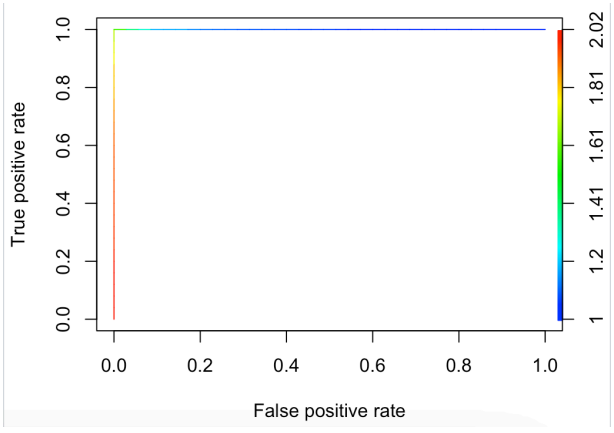
```
Call:
 randomForest(formula = DEATH_EVENT ~ ., data = clinic_test, mtry = 12,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 12

        OOB estimate of  error rate: 11.67%
Confusion matrix:
   0  1 class.error
0 33  2  0.05714286
1  5 20  0.20000000
```

```
> cost3(r = clinic_test$DEATH_EVENT, pi = pred_glm0_rf, pcut)
[1] 0.6333333
```

Figure 21

Next we found the confusion matrix for out-of-sample and we got 5 false negatives and 2 false positives. The OOB (class.error) estimate of error rate for this came out to be 11.67% which means that there is a 11.67% discriminate between the different random forest tree classifiers. Along with this, we got a cost of 0.633.

ROC Curve and AUC (Out of Sample)



```
> unlist(slot(performance(pred, "auc"), "y.values"))
[1] 1
```

Figure 22

We found the out of sample ROC curve and AUC value for our random forest model to be exactly 1. This means that all the predictions in our out of sample testing were 100% correct. Our out of sample random forest model has had the greatest AUC value out of all others tested.

Conclusion

To officially determine which model out of the three (logistic regression, classification tree, and random forest) tested was the best, we have 3 parameters we can compare them by: the confusion matrices, cost functions, and AUC values.

When we compared the misclassification matrix we made sure to mainly focus on the false negatives in the tables. We noticed for the in-sample matrices, that random forest had the highest data points classified, followed by logistic regression, and last was classification tree. For out-of-sample, we got logistic regression and classification trees as the same (2 data points), but random forest had a larger number of 5. From this, we concluded that the model from the classification tree would be the best model.

Next, we compared the out-of-sample's cost function value for all the three models. The higher the cost function, the more concerning it is because the health of a patient is at stake and the hospital could face serious lawsuits. Logistic regression had a cost of 0.217, classification tree had a cost of 0.46, and lastly random forest had the highest cost with value of 0.633. We can say that, in this case, logistic regression is the best model.

We can now compare the AUC values of three models we created. The out of sample logistic regression had a better value of .95 than in sample testing. The classification tree model had an AUC value of .92 and lastly, the out of sample random forest model had a better value of 1 than in sample testing. Out of all the models, the out of sample random forest model had the largest AUC value (1) of all, meaning that the predictions were 100% correct.

At the end, we decided to pick the best model which has less false negatives, less cost value and more AUC value. On this basis, we found that the logistic regression model was the best model of all the three models. These 7 independent variables age, creatine_phosphokinase, ejection_fraction, serum_creatinine, serum_sodium, sex, and time have the most effect on our response variable DEATH_EVENT.

References

UCI Machine Learning Repository: Heart Failure Clinical Records Data Set. (n.d.). https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records

Ahmad, T., Munir, A., Bhatti, S. H., Aftab, M., & Raza, M. A. (n.d.). *Survival analysis of heart failure patients: A case study*. PLOS ONE. Retrieved December 6, 2022, from https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0181001#sec002