Predicting the Survival of Patients Using Machine Learning

Machine Learning, Spring 2021

Yi Cui, Yiran Li, and Zipei Zhu[1]

1 Introduction

The goal of this research is to use a variety of machine learning methods to predict the survival of patients with heart failure based on their clinical, body, and lifestyle information. We also would like to find the best statistical models for our data set and identify the most important predictors in these models. Specifically, we use both unsupervised learning (i.e., PCA and clustering) and supervised learning methods (i.e., kNN, logistic regressions, LASSO, and decision trees) on our data set.

Consistent with the existing literature, we find that age, the level of serum creatinine in the blood, and the speed at which the blood running through the heart (i.e., ejection fraction) are the three most important predictors for the survival of patients with heart failure. In specific, younger patients, a lower level of serum creatine, and a higher level of ejection fraction are more likely to survive with heart failure.

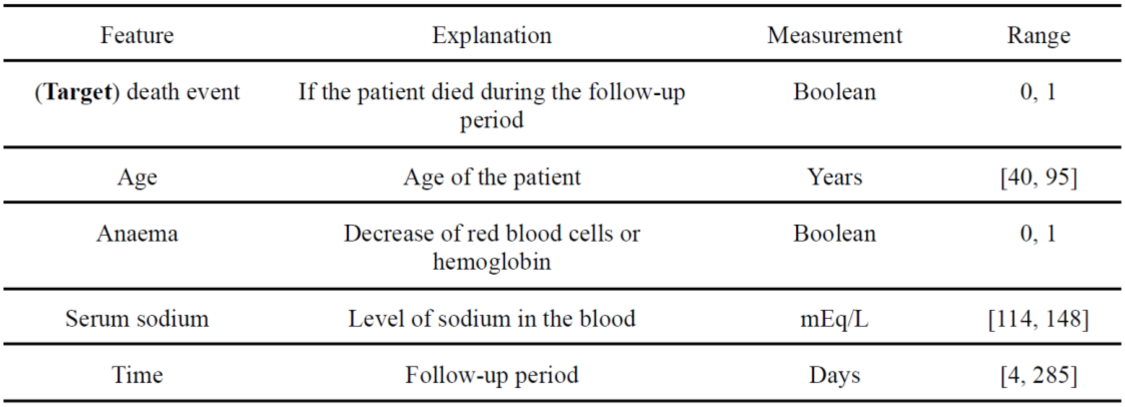
2 Data

Heart failure is a type of cardiovascular disease that causes the death of approximately 17 million people worldwide annually. The Heart Failure Clinical Records Dataset is one of the most important data sets at the frontiers of heart failure, and the data set was collected for conducting various clinical and biostatistical research. It contains the medical records of 299 patients collected at the Faisalabad Institute of Cardiology and at the Allied Hospital in Faisalabad, Pakistan, from April to December 2015. The patients consist of 105 women and 194 men with their ages ranging between 40 and 95 years old, and all 299 patients had heart failure during their follow-up period. There are 13 integer or real-valued features characterizing the clinical, body, and lifestyle information for each patient. Real-valued features include the test results for some clinical tests; Binary features consist of anemia3, high blood pressure, diabetes, sex, and smoking. A brief description of some selected features

[1] Cui: School of Economics, University of North Carolina, Chapel Hill, NC 27599 (e-mail: yicui@unc.edu); Li: School of Statistics, University of North Carolina, Chapel Hill, NC 27599 (e-mail: yiran1@live.unc.edu); Zhu: Kenan-Flagler Business School, University of North Carolina, Chapel Hill, NC 27599 (e-mail: zipei_zhu@kenan-flagler.unc.edu)

can be found in Table 1. This is a balanced data set with 299 sample, 13 features, and no missing values.

Table 1. Selected features of the dataset

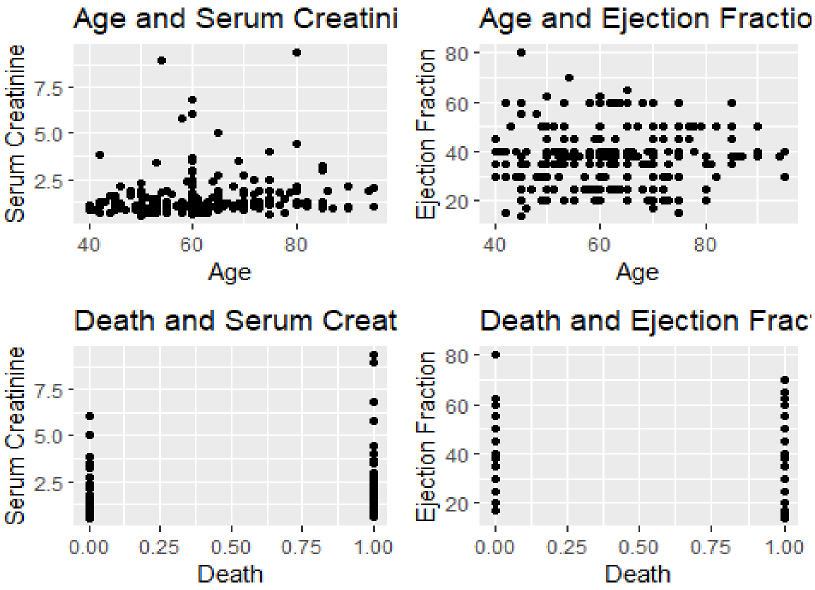


Feature	Explanation	Measurement	Range
(**Target**) death event	If the patient died during the follow-up period	Boolean	0, 1
Age	Age of the patient	Years	[40, 95]
Anaema	Decrease of red blood cells or hemoglobin	Boolean	0, 1
Serum sodium	Level of sodium in the blood	mEq/L	[114, 148]
Time	Follow-up period	Days	[4, 285]

For the following sections, we use the original data set for exploratory data analysis because this way we could see the mean, variance, distributions, and other summary statistics of our interest more clearly. Instead, we standardize the non-boolean variables when performing machine learning methods because different numerical features may have different means and variances such that they are not intuitively comparable with each other or not compatible with some statistical learning models.

2.1 Exploratory Data Analysis (EDA)

We are interested in the correlation between some predictors, such as age, serum_creatinine, and ejection_fraction, and DEATH_EVENT, the survival of patients, so we visualize their joint behaviors and calculate the correlation coefficients. We find that there is a weakly positive correlation between age and serum creatinine (i.e., 0.159), and the same is that between age and ejection fraction (i.e., 0.060). Instead, there is a relatively stronger and positive correlation between death and serum creatinine (i.e., 0.294). Also, the same is with that between death and ejection fraction whereas the correlation is negative (i.e., -0.269).

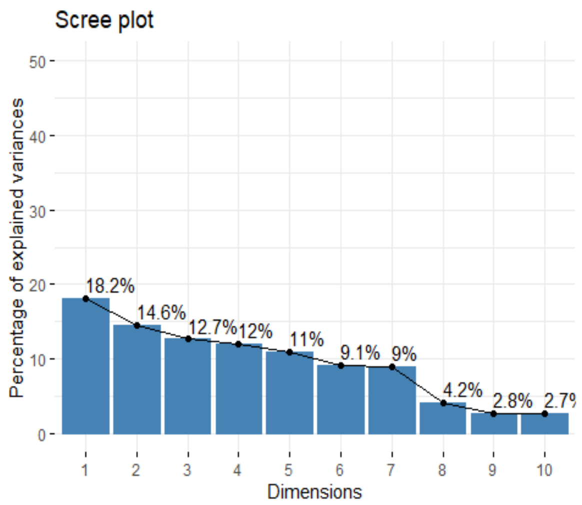


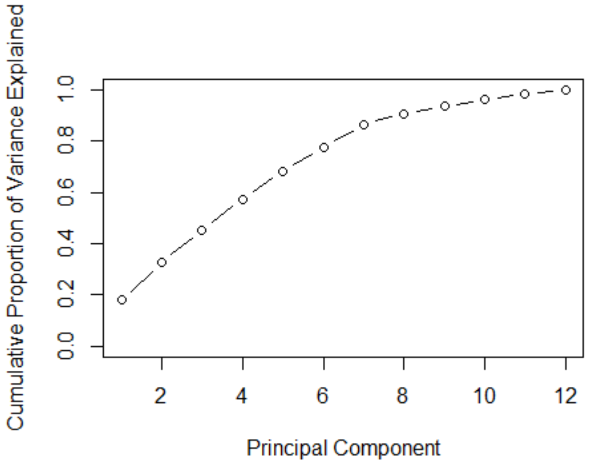
3 Learning Methods

3.1 Principal Components Analysis

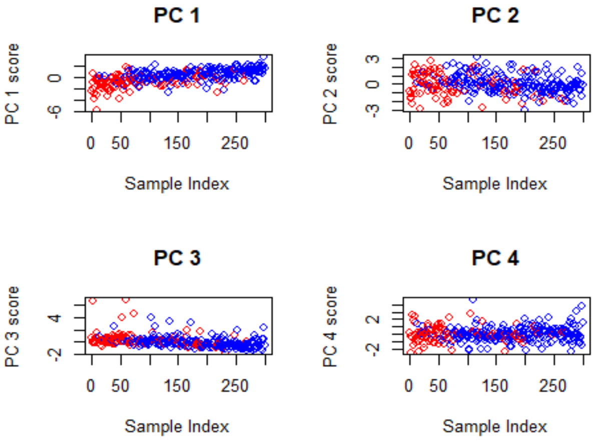
We first try the unsupervised learning method, PCA, and we would like to know if we could find a low-dimensional representation of the observations. PCA projects the original high-dimensional data onto a low-dimensional space and make its variance as large as possible. If the value of a certain feature (a column of the matrix) of the data is particularly large, then it has a large proportion of the entire error calculation. Because we don't know the importance of each feature before modeling, this is likely to lead to a large amount of information missing. For the sake of "fairness" and to prevent over-capturing certain features with large values, we will first standardize each feature so that their sizes are within the same range, and then perform PCA. From a computational point of view, another benefit of standardization before PCA is that this is beneficial to the gradient descent method of convergence. Because PCA is usually numerically approximated decomposition, rather than seeking eigenvalues, singular values to obtain analytical solutions, when we use gradient descent and other algorithms for PCA, we have to standardize the data first.

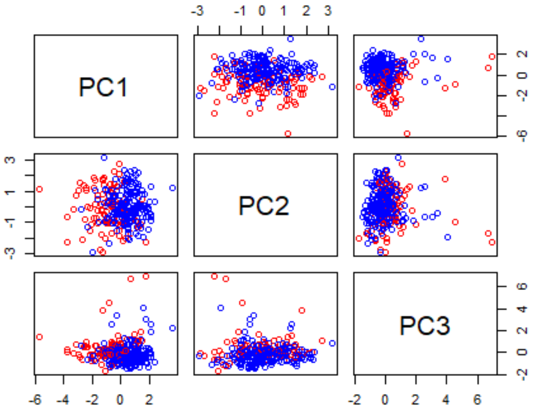
We run PCA on the data set, provide a numerical summary of the first 5 PCs, and plot a screeplot of the PCs.





We find that seven principal components are required to explain at least 80% of the variation in the data, which is more than half of the PCs; and PC1's variance percentage is only 18.57%, which is not really high. Based on this and the plot, it doesn't seem like PCA has done a good job in reducing the dimensionality of the data.



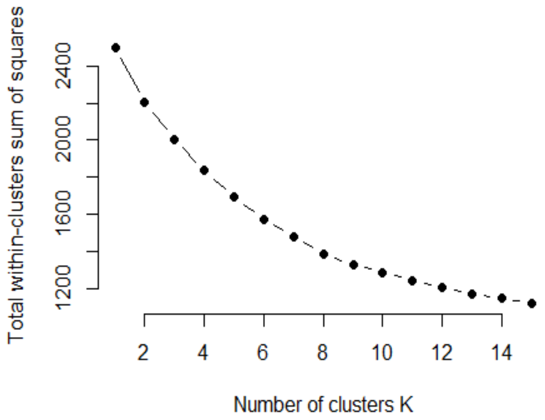


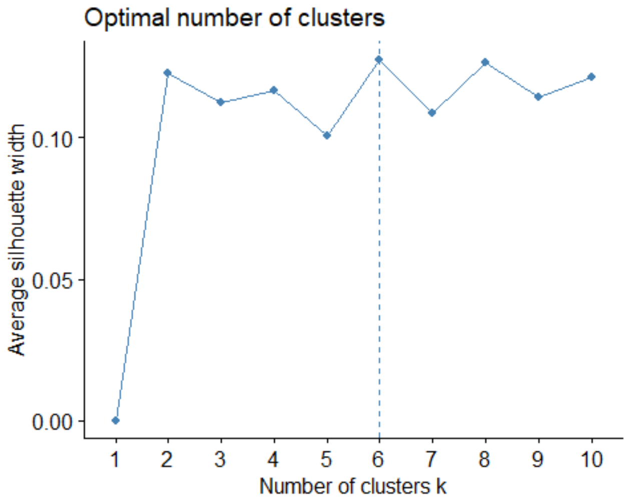
Plotting the PCs separately, we find that there doesn't seem to have any apparent clusters. All clusters seem to overlap on each other. However, after controlling for PC2 and PC3, dead patients colored in red seem to have lower PC1 than the patients alive. But there does not seem to be any clustering depending the levels of PC2 and PC3.

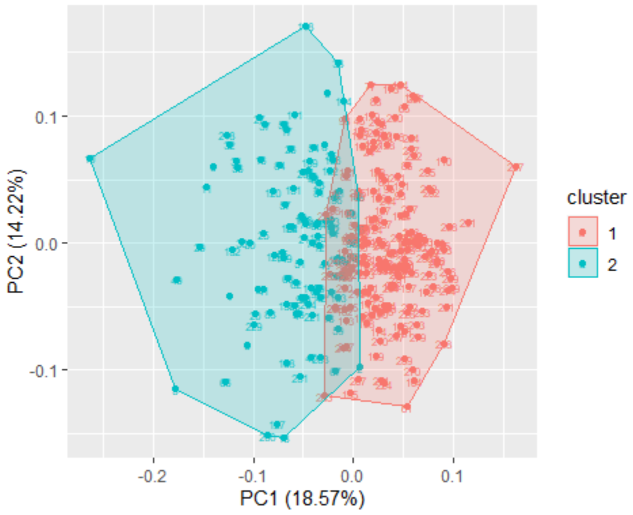
3.2 Clustering

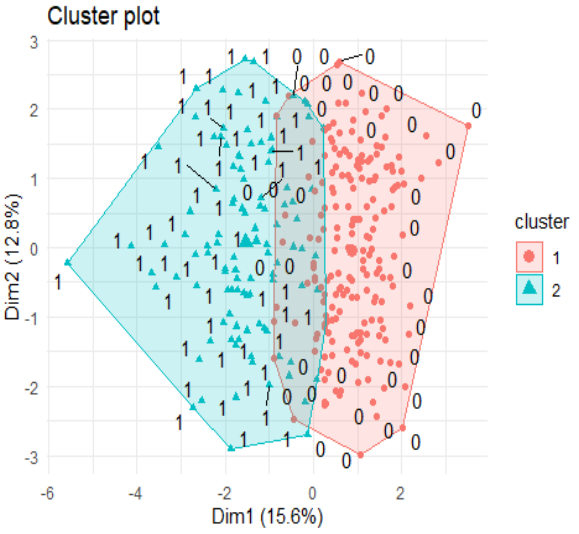
Next, we perform a cluster analysis on the data.

We are interested in partitioning around medioids, self organizing maps. Intuitively, it means we could use K-means algorithm to classify the data into several different groups. Clustering is a method of unsupervised learning and the objects being clustered of course are the data points. It is quite useful for this data set because many features are binary, so we just need to cluster the data points based on a limited number of features.









```
## # A tibble: 4 x 3
## # Groups:   cluster [2]
##   cluster class     n
##     <int> <int> <int>
## 1       1     0   175
## 2       1     1    15
## 3       2     0    28
## 4       2     1    81
```

We can see that the clustering algorithm divides the data into two distinct clusters with cluster 1 being composed of mainly 1s (i.e., 1 stands for the dead patients, and 0 for survived patients) with some 0s along the boundaries, and cluster 2 being mostly composed of 1s. Therefore, it seems safe to say that both clusters are relatively homogeneous. Next, we figure out how many data points in each cluster are 1s and 0s. We find that that both two clusters are entirely homogeneous with very few misclassified data points.

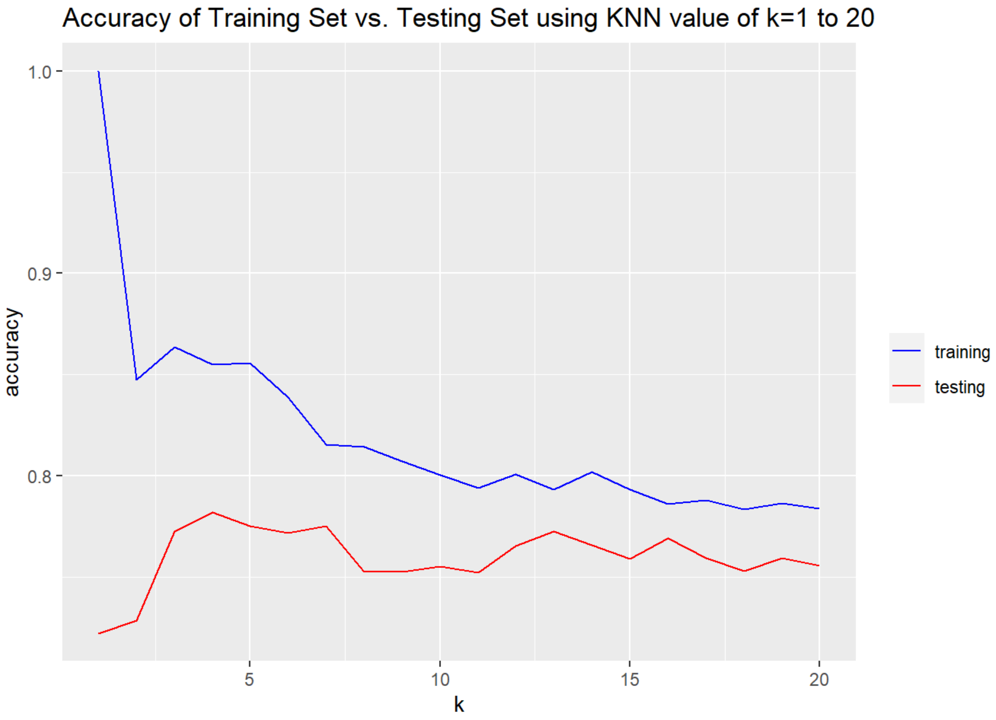
3.3 Classification

Classification is simply the process in which one decides which class a new data point may fall into. A single object in the case of our dataset, is the set of survival results which are classified as either "0" or "1". The variable that would be most interesting to predict is the "DEATH_EVENT" variable. This variable is either "0" or "1" in our dataset, indicating whether a patient has survived or not in their follow-up period. As stated before, predicting this variable can predict which subtype a set of measurements corresponds to results. For doing cross-validation in all classification methods in this section, we randomly break our data set into a training set (roughly 80% of the data) and a test set (roughly 20% of the data); while doing the 10-fold cross-validation, we randomly partition the data into 10 equal size subsamples, of the 10 subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 9 subsamples are used as training data.

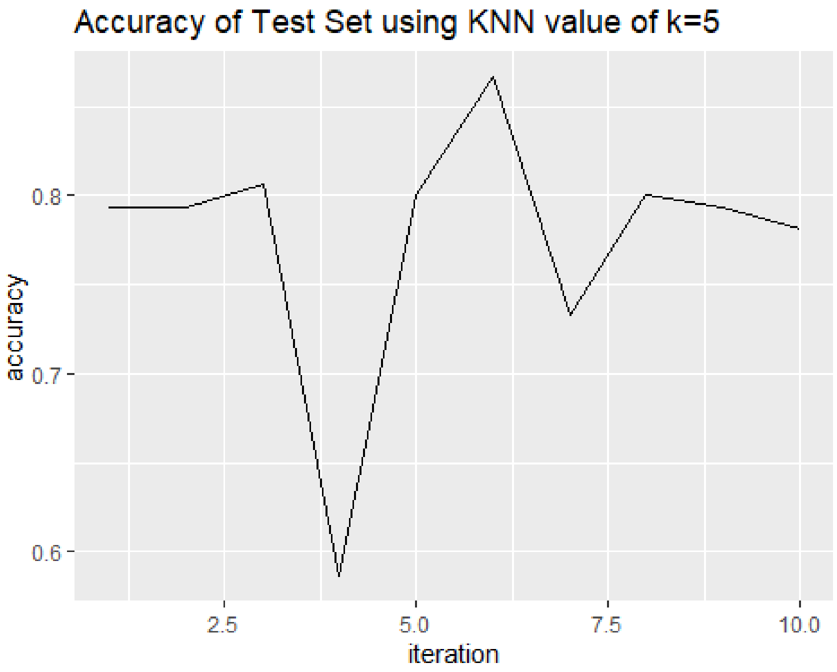
3.3.1 K Nearest Neighbors

Then we apply the supervised learning method k-nearest neighbors (kNN) on our dataset. We choose kNN because of its easy of interpretation and low calculation time. Also, with the optimal k value, we can easily make a boundary of the two classes (dead or survived) that clearly segregates them from each other.

```
## # A tibble: 4 x 3
## # Groups:   cluster [2]
##   cluster class     n
##     <int> <int> <int>
## 1       1     0   175
## 2       1     1    15
## 3       2     0    28
## 4       2     1    81
```



From the plot we could see that the accuracy reaches its first maxima when k = 4. Since we want k to be an odd number to avoid even vote, we segregate the training and validation from the initial dataset, apply 10-fold cross-validation on k = 3 and k = 5, plot the validation accuracy for each, and calculate the average accuracy.



From the calculated result, the optimal k value is 5 with the accuracy of 0.7753. This value of k should be used for all predictions.

3.3.2 The Logistic Regression and LASSO

Since our predict target is a binary variable, and one of our goal is to find out which features are significant for the prediction, we then apply the logistic regression on our data set. Logistic regression model is easy to interpret, and all variables are in there, so we consider it as our baseline.

```
##
## Call:
## glm(formula = DEATH_EVENT ~ ., family = binomial, data = scale_heart.logi,

##     subset = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0510  -0.7559  -0.4778   0.8913   2.6557
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -1.197451   0.399098  -3.000  0.00270 **
## age                         0.658037   0.167733   3.923 8.74e-05 ***
## creatinine_phosphokinase  0.143166   0.171856   0.833  0.40481   
## ejection_fraction        -0.566296   0.185103  -3.059  0.00222 **
## platelets                -0.308396   0.192955  -1.598  0.10998   
## serum_creatinine          0.495018   0.188036   2.633  0.00847 **
## serum_sodium             -0.388286   0.191921  -2.023  0.04306 * 
## anaemia                   0.410482   0.325761   1.260  0.20764   
## diabetes                 -0.008289   0.328848  -0.025  0.97989   
## high_blood_pressure       0.499224   0.342861   1.456  0.14538   
## sex                      -0.261777   0.396256  -0.661  0.50885   
## smoking                   0.208883   0.380170   0.549  0.58270   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 301.89  on 238  degrees of freedom
## Residual deviance: 242.93  on 227  degrees of freedom
## AIC: 266.93
## 
## Number of Fisher Scoring iterations: 5

## [1] 0.75
```

From the summary of results, we know that age, ejection fraction and serum creatinine are the top three significant predictors with large magnitude relative to the other predictors for a patient’s survival. We cannot control a patient’s age, but we are able to give clinical control over the level of ejection fraction and serum creatinine in a patient’s blood. Judging from the signs of their coefficients, we know that low level of serum creatinine (-) in the blood and high percentage of ejection fraction (+) would increase the log-odds of the survival of a patient, which might be helpful for clinical research. Though not perfect, the accuracy of 0.65 is good enough for a baseline model, and we are likely to have found the three most significant predictors.

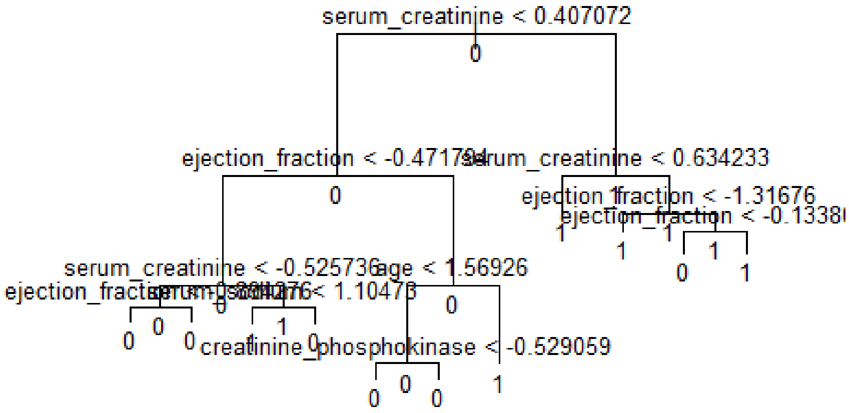
```
## [1] 0.02878973

## [1] 0.1905741

## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                   s0
## (Intercept)               0.31742208
## age                       0.08009002
## creatinine_phosphokinase  0.00614508
## ejection_fraction        -0.09596062
## platelets                 .         
## serum_creatinine          0.09028105
## serum_sodium             -0.02537043
## anaemia                   .         
## diabetes                  .         
## high_blood_pressure       0.01038856
## sex                          .
## smoking                      .
```

Next, we use Least Absolute Shrinkage and Selection Operator (LASSO) to help us select the most important features to better predict the survival of patients in a linear setting. We find that only six variables with high statistical significance and large magnitude are left after the model's selection: age, creatinine_phosphokinase, ejection_fraction, serum_creatinine, serum_sodium, and high_blood_pressure. Among them, the coefficients of age, ejection_fraction, and serum_creatinine have the largest magnitude. This result validates what we have discovered from the baseline.

3.3.3 Decision Trees

The final classification method we use is decision trees, because it is a white box model that is easy to understand and interpret. Also, it can potentially judge our previous results. We fit the whole data set on the basic tree, do cross-validation, and finally create a pruned tree to improve the performance.

```
##
## Classification tree:
## tree(formula = death ~ . - DEATH_EVENT - time - sex, data = scale_heart_te
mp)
## Variables actually used in tree construction:
## [1] "serum_creatinine"         "ejection_fraction"
## [3] "serum_sodium"             "age"
## [5] "creatinine_phosphokinase"
## Number of terminal nodes:  11
## Residual mean deviance:  0.8297 = 238.9 / 288
## Misclassification error rate: 0.1773 = 53 / 299
```


serum_creatinine < 0.407072
0
ejection_fraction < -0.471794
serum_creatinine < 0.634233
0
ejection_fraction < -1.31676
ejection_fraction < -0.1338
1
1
1
1
1
0
1
serum_creatinine < -0.525736
age < 1.56926
ejection_fraction
serum_sodium < 1.10473
0
0
0
0
0
1
1
0
creatinine_phosphokinase < -0.529059
0
0
0
1

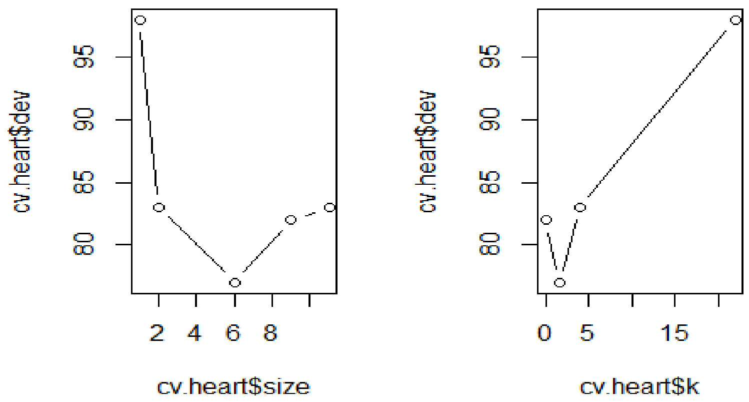
```
##             death.test
## tree.predict  1  0
##            1 11  8
##            0  7 34
```

Consider the error on the whole data set. There are five variables actually used as internal nodes in the tree: serum_creatinine, ejection_fraction, age, serum_sodium, creatinine_phosphokinase. Plus, there are eleven terminal nodes. The split criterion, the number of observations, the deviance, and the overall prediction in that branch (between 1 and 0) are also shown above. Overall, the training error rate is 17.73%. This relatively small deviance indicates a tree that provides a good fit to the training data.

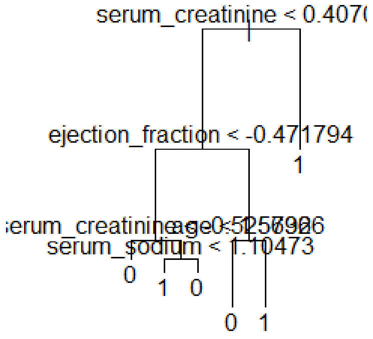
From the graph, we find that the most important feature is serum_creatinine since the first branch differetiates the level of serum creatine in the patient's blood using a threshold of 0.407. Plus, this feature is used once again in the following internal nodes. The other most important feature seems to be ejection_fraction, which is used three times in the following internal nodes.

In order to properly evaluate the performance of this classification tree on the data, we must estimate the test error rather than simply computing the training error. Hence, we split our data into a training set and a test set. This approach leads to correct predictions for around 75% of the data points in the test data set.

```
## [1] "size"   "dev"    "k"      "method"

## $size
## [1] 11  9  6  2  1
## 
## $dev
## [1] 83 82 77 83 98
## 
## $k
## [1]      -Inf  0.000000  1.666667  4.000000 22.000000
## 
## $method
## [1] "misclass"
## 
## attr(,"class")
## [1] "prune"         "tree.sequence"
```



```
##                 death.test
## tree.heart.pred  1  0
##                1 16 10
##                0  2 32
```


serum_creatinine < 0.4070
ejection_fraction < -0.471794
1
serum_creatinine age0<5125796926
serum_sodium < 1.10473
0
1
0
0
1

Finally, we consider whether pruning the tree might lead to improved results. To start off, we perform cross-validation in order to determine the optimal level of tree complexity, and we do cost complexity pruning to select a sequence of trees for consideration. From the plot, we observe that the tree with 6 terminal nodes has the lowest cross-validation error rate, with 77 cross-validation errors. Next, we prune the tree to obtain the six-node tree. We find that now 80% of the observations are correctly classified, and there are only four predictors remained in our pruned tree: serum_creatinine, ejection fraction, serum sodium, and age. This resulting pruned tree also validates our base-line result that regards serum_creatinine and ejection fraction as two most important predictors. To sum up, the pruning process produces a more interpretable tree, and it also improves the classification accuracy.

Conclusion

We perform both unsupervised and unsupervised learning methods on the Heart Failure Clinical Records Dataset with 299 samples and 13 variables to predict the survival of patients with heart failures based on their clinical, body, and lifestyle information.

Since our data set is not very highly dimensional, PCA does not yield good results as expected since it is most powerful in reducing the dimensionality of data. However, K-means clustering does a great job in separating the data into two clusters with distinct values of our target variables, DEATH_EVENT. Similarly, exploratory data analysis (EDA) by looking at correlation between some predictors and our target is also helpful. Results from both clustering and EDA suggest that patients with heart failures that survived may have very different characteristics from those that are dead from the disease. Hence, it would be promising to further use supervised methods predict their survival based on their attributes.

Due to the binary nature of patients' survival, naturally, we use the logistic regression as our baseline model. With a good accuracy of 0.65, our baseline shows that the most important predictors are age, serum creatinine, and ejection fraction, because their coefficients are the most statistically significant with the largest magnitude. The results of LASSO validate this prediction as the three variables are also the most important predictors among all six predictors after the model's selection. We also try k-nearest neighbors since this model is easy to interpret and simple to compute, though we can not directly infer the relative importance of patients' attributes. Using 10-fold cross-validation, we find k equal to 5 to be the optimal choice of the number of the nearest neighbors with a high accuracy around 0.78.

The final method we use is decision trees because it is a white box model that is easy to understand and interpret. The baseline tree model gives a great training accuracy of 0.83 and indicates that serum creatine and ejection fraction are two most important predictor for our target, consistent with our previous results. Plus, cross-validation yields a 0.75 accuracy on the test data set. Also, we try pruning the tree in order to improve the preceding results. We find that the tree with six nodes has the lowest cross-validation error rate. As a result, the test error rate increases from 0.75 to 0.80, and the tree is tuned such that it is more obvious to see the three most important predictors mentioned above.

To conclude, both EDA and our classification methods shows that the most two important predictors for patients' survival are serum creatinine and ejection fraction. It might be helpful for doctors to reduce the level of serum creatine in the patient's blood and boost the velocity of bloodstream in the patient's heart by some clinical treatment to increase their survival rate. Our research can be further extended to using other machine learning methods, such as support vector machines or Naive Bayes. Though Naive Bayes is not recommendable because of its far-reaching independence assumption, which is unrealistic for our data, SVMs could be worth trying in order to get better results in the future.