# Heart failure classification using support vector machine and model based clustering

**STATS/CSE 790 Assignment 5**

**2024-03-19**

**Pao Zhu Vivian Hsu (400547994)**

## Introduction

In this study, we use Support Vector Machines (SVMs) and model based clustering to predict mortality attributed to heart failure. The data in this study comes from an open sourced website called Kaggle (Larxel, 2020). It contains 299 rows of patient data and 13 variables of measurements. The response variable is a binary categorical variable indicating the event of death. The remaining variables include age, anaemia, creatinine phosphokinase, diabetes, ejection fraction, high blood pressure, platelets, serum creatinine, serum sodium, sex, smoking, and time.

## Methods

We start the study by checking the shape of the data and ensuring there are no missing values. We performed data visualization using a pairs plot to check for any patterns in the data. Figure 1 shows a subset of this plot, where blue indicates a mortality case and red indicates the absence of one. From this plot, we can see that the observations have a non-linear boundary between them for many of the variable combinations, making the data suitable for SVM or model based clustering.

As such, we then split the data into two equal parts to form training and testing sets and applied these two different techniques to the data.

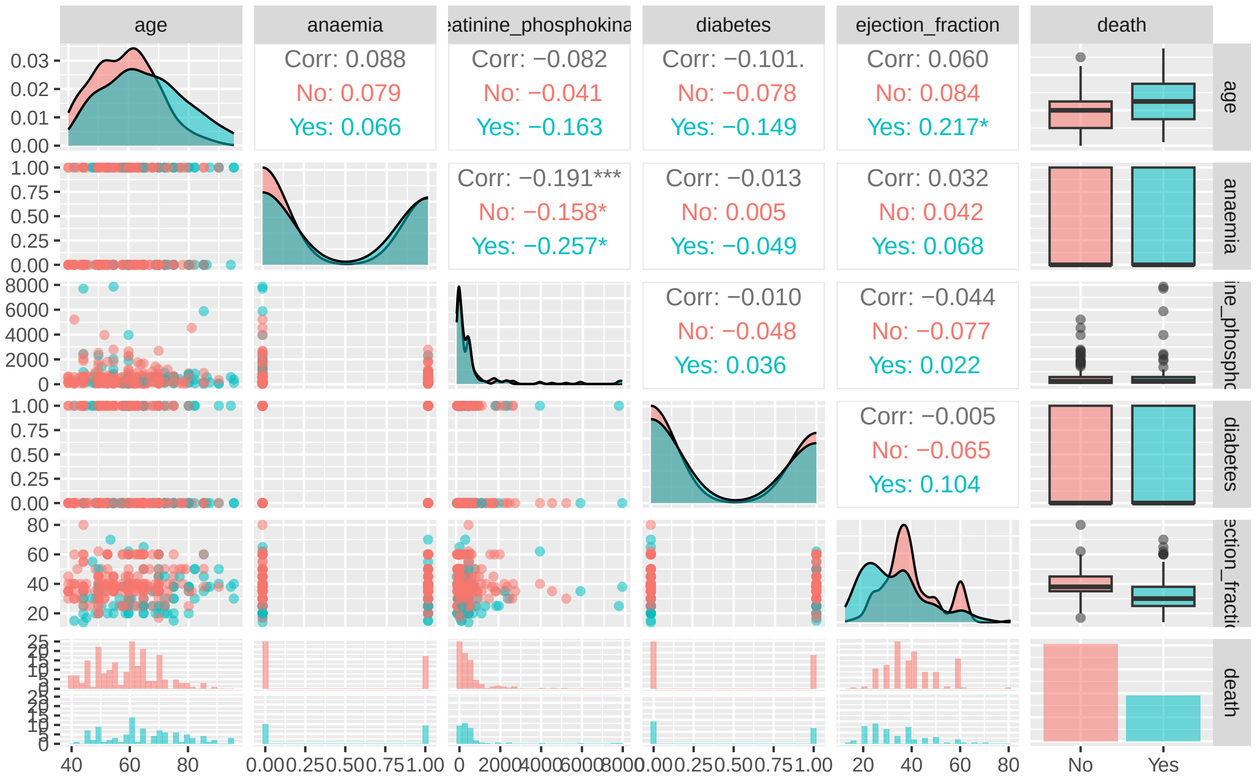

Figure 1: Pairs plot of a selection of the variables

The first technique we applied was SVM, which is a supervised learning method that performs classification using decision boundaries as defined by a kernel. A kernel is a function that is used to measure the similarity between observations and can be used to enlarge the feature space to accommodate non-linear boundaries between classes (King-Yu, 2024b).

For our study, we used linear and radial kernels (King-Yu, 2024b). The linear kernel can be written as:

$$K(x_i, x_i') = \sum_{j=1}^{p} x_{ij}, x_i{}'_j$$

where $i$ represents the $i$th observation. The radial kernel can be written as:

$$K(x_i, x_i') = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_i{}'_j)^2)$$

where $i$ represents the $i$th observation and $\gamma$ is a positive constant (King-Yu, 2024b).

We used cross-validation to tune the cost parameter C and the gamma constant $\gamma$. The cost parameter measures the number of observations that can be on the wrong side of the hyperplane and the tolerance towards margin violations. In other words, it puts a limitation on the sum of slack variables $\epsilon_i$, where $\epsilon_i$ indicates whether the $i$th observation is on the correct side of the margin ($\epsilon_i = 0$), violates the margin ($\epsilon_i > 0$), or is on the wrong side of the hyperplane ($\epsilon_i > 1$). Thus, C controls the bias and variance in the model. When C is large, we have wide margins and the model will tolerate many margin violations. When C is small, we have small margins and the model will have less margin violations. We seek to optimize this bias-variance trade off through cross-validation (King-Yu, 2024a).

For the purposes of this study, we tried using 10 different values for each parameter. For the linear model, we found a cost parameter of $C = 1$ to be the optimal value. For the radial kernel, we found a cost parameter of $C = 5$ and a gamma value of $\gamma = 3$ to produce optimal results. After cross-validation was performed, the models were then fit using the test set and results were recorded.

The second technique we applied to the data was model based clustering using all the predictor variables and then a subset of them. The first model contained all 13 predictor variables. It had a ellipsoidal, equal volume and shape (EEV) structure. The second model contained only 2 predictor variables, sex and smoking status, as shown in Figure 2. This was selected using Variable Selection for Clustering and Classification (VSCC), which selects predictor variables that maximize the between-group variance and minimize the within-group variance (King-Yu, 2024c). The best reduced model had a diagonal, equal volume and shape (EEI) structure.

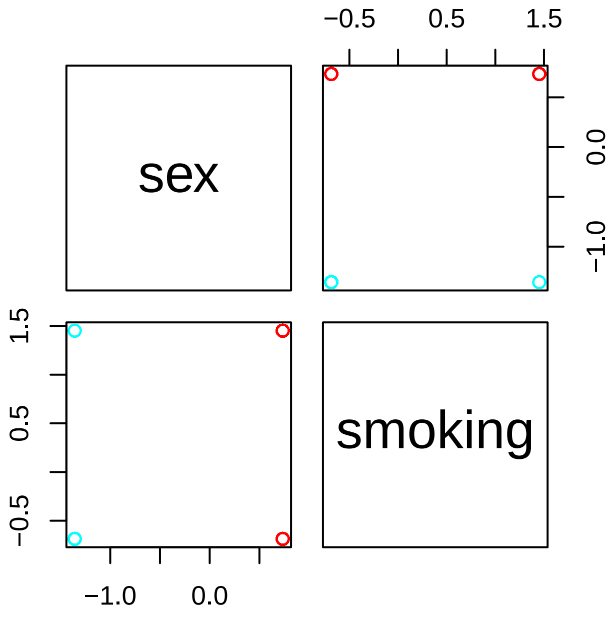

Figure 2: Sex and smoking were the variables selected

## Results

Table 1 below summarizes the SVM results using linear and radial kernels. The adjusted Rand index (ARI) is very poor for both kernels. For the linear kernel, the ARI is 0 indicating that the classification may not be very accurate as the results could be due to random chance. For the radial kernel, the ARI is negative indicating that the classification may not be very accurate as well. Similarly, the misclassfication error rate also shows a very poor performance for both kernels.

Table 1: SVM performance comparison

| Method | Adjusted Rand Index | Misclassification Error Rate |
|---|---:|---:|
| SVM Linear Kernel | 0.00000 | 0.99333 |
| SVM Radial Kernel | -0.00086 | 0.99333 |

Table 2 below summarizes the model based clustering results using the full and reduced model. The adjusted Rand index (ARI) is quite low for both kernels. For the full model, the ARI is 0.00206 indicating that the classification may not be very accurate as the results could be due to random chance. For the reduced model, the ARI is negative indicating that the classification may not be very accurate as well. The misclassification error rate, on the other hand, has shown substantial improvement compared to the SVM models. The rate is similar for both the full and reduced model, with the rate being slightly better for the full model compared to the reduced one.

Table 2: Model based clustering performance comparison

| Method | Adjusted Rand Index | Misclassification Error Rate |
|---|---:|---:|
| Full Model | 0.00206 | 0.44147 |
| Reduced Model | -0.00186 | 0.44482 |

## Conclusion

Overall, the results of our SVM and model based clustering analysis was not very effective to predict mortality attributed to heart disease. Of our results, the full model is likely the strongest model for prediction among the four models. Future studies may want to consider other kernels, such as a polynomial kernel, to achieve better results when applying SVM. Otherwise, performing feature engineering, cleaning the data more extensively prior to modelling, and incorporating subject-matter expertise during interpretation my be beneficial to improve the models.

## References

King-Yu, S. (2024a). *Statistical learning lecture 13: Support vector machines i.*

King-Yu, S. (2024b). *Statistical learning lecture 14: Support vector machines II.*

King-Yu, S. (2024c). *Statistical learning lecture 15: Model selection i.*

Larxel. (2020). *Heart failure prediction.* Kaggle. https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data/data

## Appendix

```
# ----- PACKAGE & DATA SETUP ----- #
library(tidyverse)
library(GGally)
library(e1071)
library(mclust)
library(vscc)
library(kableExtra)

# Load data
heart_raw <-
  read_csv("/Users/Vivian/Documents/Workspaces/R_Workspace/stats790/A5/heart_failure_clinica

# ----- DATA TRANSFORMATION ----- #
# Check for missing data
sum(sapply(heart_raw, function(col){ifelse(is.na(col), 1, 0)}))
sum(sapply(heart_raw, function(col){ifelse(sum(col == ""), 1, 0)}))

# Full pairs plot
heartV <- within(heart_raw, death <- ifelse(DEATH_EVENT==1,"Yes", "No"))
ggpairs(data=heartV[,-c(13)], aes(colour=death, alpha=0.4))

# Smaller pairs plot
heartV <- within(heart_raw, death <- ifelse(DEATH_EVENT==1,"Yes", "No"))
ggpairs(data=heartV[, -c(13,6,7,8,9,10,11,12)], aes(colour=death, alpha=0.4))

# Split data into train and test
set.seed(790)
heart <- heart_raw
heart[, -13] <- scale(heart_raw[, -13])
train.ind <- sample(1:nrow(heart), nrow(heart) / 2)
```

```
heart.train <- heart[train.ind,]
heart.test <- heart[-train.ind,]
heart.test.labs <- heart[-train.ind, "DEATH_EVENT"]

svmfit <- svm(DEATH_EVENT ~ ., data = heart.train, kernel = "linear",
    cost = 1, scale = FALSE)

summary(svmfit)

# Cross-validation to choose the best cost
set.seed(790)
tune.out <- tune(svm, DEATH_EVENT ~ ., data = heart.train,
    kernel = "linear",
    ranges = list(
      cost = c(0.01, 0.1, 0.5, 1, 2, 3, 4, 5, 10, 20)
    )
  )

summary(tune.out)
# Prediction
pred <- predict(tune.out$best.model, newdata = heart.test)

tabSvmLinear <- table(pred, heart.test.labs$DEATH_EVENT)

svmfit <- svm(DEATH_EVENT ~ ., data = heart.train, kernel = "radial", gamma = 1, cost = 1)

summary(svmfit)

# Cross-validation to choose the best gamma and cost
set.seed(790)
tune.out <- tune(svm, DEATH_EVENT ~ ., data = heart.train,
    kernel = "radial",
    ranges = list(
      cost = c(1, 5, 6, 7, 8, 9, 10, 15, 20, 30),
      gamma = c(1, 2, 3, 4, 5, 6, 8, 10, 15, 20)
    )
  )

summary(tune.out)
# Prediction
pred <- predict(tune.out$best.model, newdata = heart.test)

tabSvmRadial <- table(pred, heart.test.labs$DEATH_EVENT)

# RESULTS
method <- c("SVM Linear Kernel", "SVM Radial Kernel")
crand <- c(classAgreement(tabSvmLinear)$crand,
           classAgreement(tabSvmRadial)$crand)
misclass <- c(1-classAgreement(tabSvmLinear)$diag,
              1-classAgreement(tabSvmRadial)$diag)
summary <- data.frame("Method" = method,
                      "Adjusted Rand Index" = round(crand,5),
                      "Misclassification Error Rate" = round(misclass,5),
                      check.names = FALSE)
kable(summary)

# MCLUST
x <- scale(heart[,-13])
modclust <- Mclust(x, 2)
summary(modclust)

tabFull <- table(class=heart$DEATH_EVENT,
                  predictions=factor(as.character(map(modclust$z)-1)))

# VSCC
x <- scale(heart[,-13])
modvscc <- vscc(x)
tabReduced <- table(class=heart$DEATH_EVENT,
                    predictions=modvscc$bestmodel$classification)
modvscc
head(modvscc$topselected)

plot(modvscc)

# RESULTS
method <- c("Full Model", "Reduced Model")
crand <- c(classAgreement(tabFull)$crand,
           classAgreement(tabReduced)$crand)
misclass <- c(1-classAgreement(tabFull)$diag,
              1-classAgreement(tabReduced)$diag)
summary <- data.frame("Method" = method,
                      "Adjusted Rand Index" = round(crand,5),
                      "Misclassification Error Rate" = round(misclass,5),
                      check.names = FALSE)
kable(summary)
```