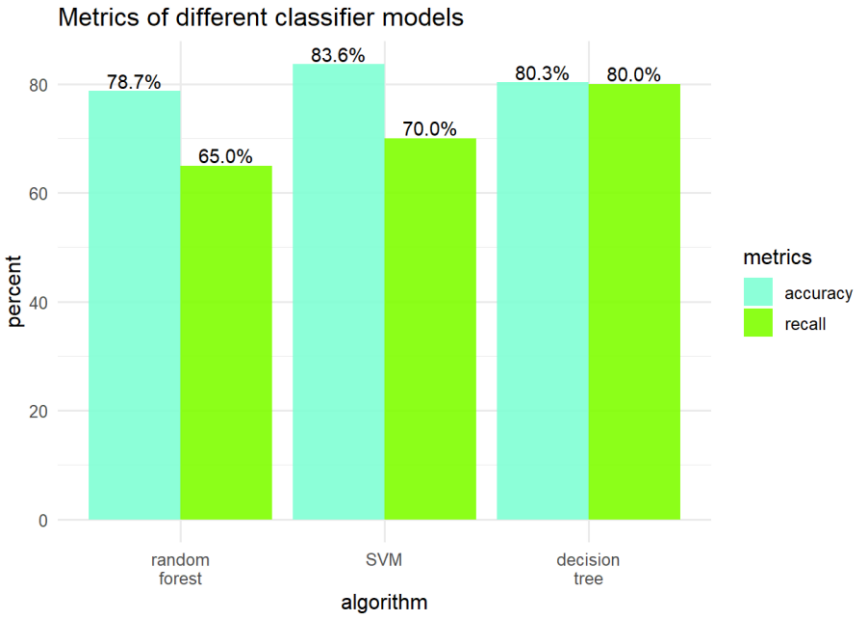
Summary

Heart failure is a prevalent condition that arises from cardiovascular diseases. This dataset with 12 features is available that can be utilized to forecast the likelihood of mortality resulting from heart failure.

There are some factors that affect death event. This dataset contains a person's information like age, sex, blood pressure, smoke, diabetes, ejection fraction, creatinine phosphokinase, serum_creatinine, serum_sodium and time and a machine learning model can be beneficial in predicting their death event. This can be used to help hospitals in assessing the severity of patients with cardiovascular diseases (CVDs).

In this report, I checked and visualized the data to gain insights. I found that these 12 features have different impacts on mortality. Therefore, I divided the dataset into two groups, selecting 80% as the training set and the remaining 20% as the test set. I used three regression models (SVM), decision tree, and random forest) to predict this dataset and compared their accuracy and recall rates.



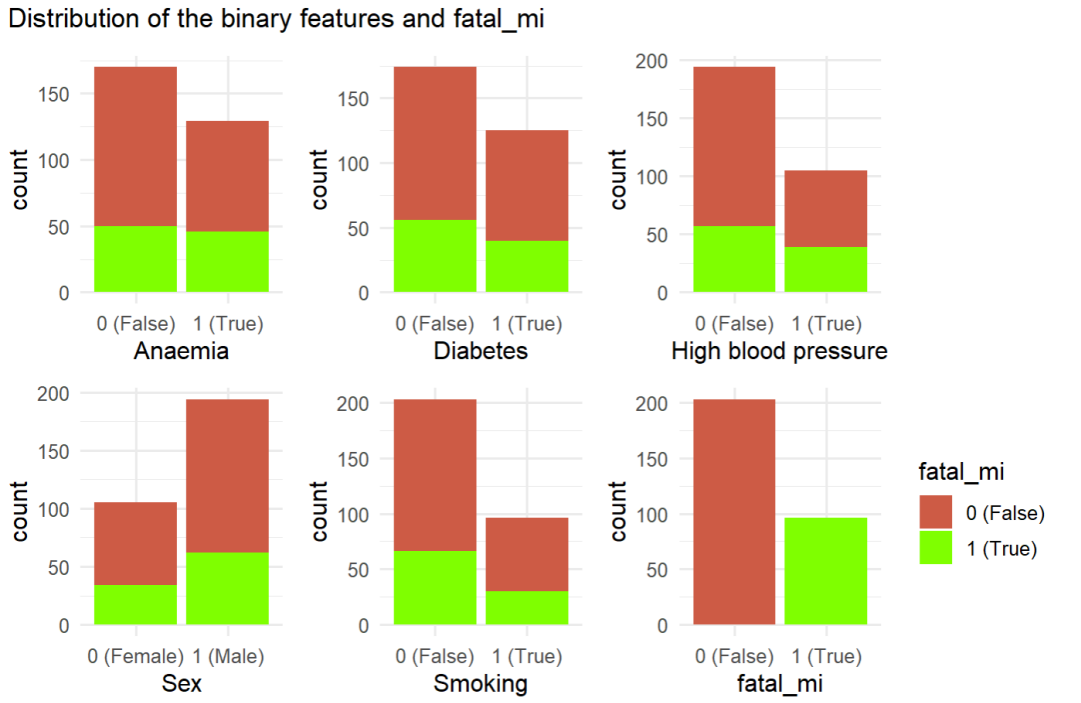
From the graph, we can see that the accuracy of the SVM model can reach as high as 83.6%, but the recall rate is only 70%. The accuracy and recall rates of the random forest model are not ideal, while the decision tree model has an accuracy and recall rate of 80%. After considering all factors, I ultimately chose the SVM algorithm. Because in this case, I care more about the accuracy of those models.

Data and Visualizations

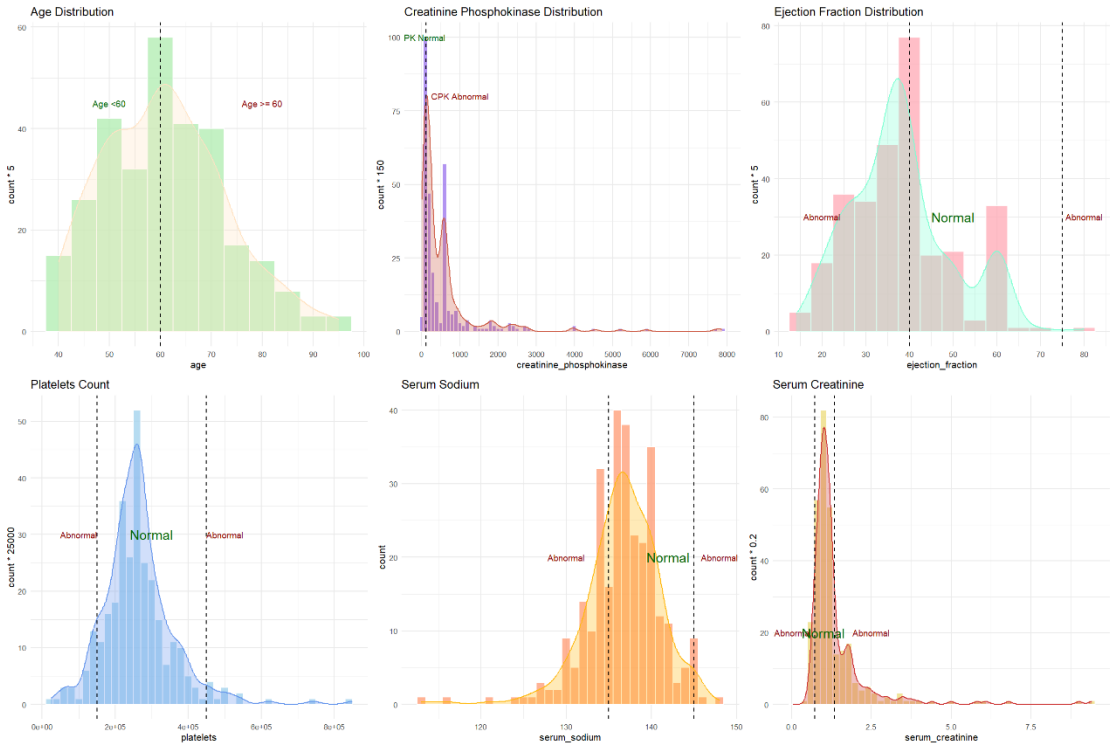
This is the explanation of variables:

age - Age
anaemia - Decrease of red blood cells or hemoglobin (boolean) (0:False, 1:True)
creatinine_phosphokinase - Level of the CPK enzyme in the blood (mcg/L)
diabetes - If the patient has diabetes (boolean) (0:False, 1:True)
ejection_fraction - Percentage of blood leaving the heart at each contraction (percentage)
high_blood_pressure - If the patient has hypertension (boolean) (0:False, 1:True)
platelets - Platelets in the blood (kiloplatelets/mL)
serum_creatinine - Level of serum creatinine in the blood (mg/dL)
serum_sodium - Level of serum sodium in the blood (mEq/L)
sex - Woman or man (binary) (0: Woman, 1: Man)
smoking - If the patient smokes or not (boolean) (0:False, 1:True)
time - Follow-up period (days)
fatal_mi - If the patient deceased during the follow-up period (boolean)

Firstly, we utilize data visualization to plot the distribution of some features and targets to gain a rough understanding of the data. Then we adjust the data to facilitate subsequent analysis.



For anaemia and diabetes, there was little difference in the distribution of the objective variable. For high blood pressure, sex, and smoking, there are some differences in the distributions of the objective variables, but we don't know if these differences are significant or not. And I have created more distribution plots for these data.



We can find that the age distribution of the patients peaked at around 60 years old, and then decreased in a symmetrical bell-shaped pattern around this age. The distribution of creatinine phosphokinase is significantly skewed towards one end, with a maximum value that is over 30 times higher than the median. The distribution of ejection fraction is discrete, not continuous, with the first peak near 49 and the second peak at 60. When analyzing the target variable, there are notable variations in both the distribution shape and median. Survivors are primarily concentrated around the first and second peaks, while the values of the deceased are mainly centered around 30 and gradually decrease from that point. For the platelets, the distribution is roughly symmetrical and almost bell-shaped. And we can see that survivors are clustered around the median value. For serum sodium distribution, it is roughly symmetrical and almost like a normal distribution, with few data over 150, and also rare below 125. The values of survivors are clustered around the median value, while the values of deaths are lower and tend to be more dispersed. Finally, the distribution of serum creatinine is heavily skewed to one side. There are rare values more than four times the median value.

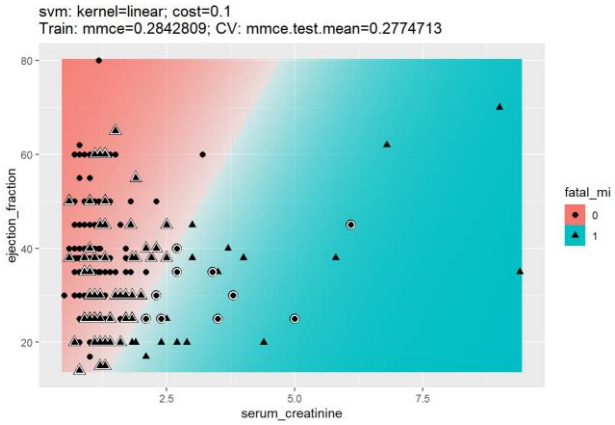
Train models and make predictions

I split the dataset into the train set(0.8) and the test set(0.2). And then create 3 different models and check the performance measures. In this report, we care about their accuracy, recall, and confusion matrix.

SVM(Support vector machine)

A support vector machine (SVM) is a model used for classification and other purposes. It creates a boundary, called a separation hyperplane, that clearly separates two or more classes.

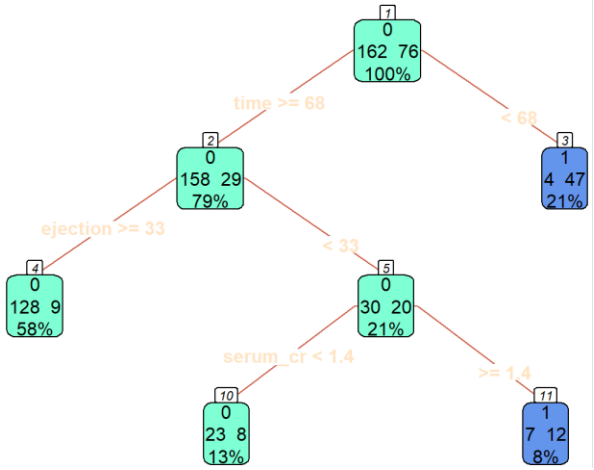
To train an SVM, I use training data and then visualize the learning algorithm with the plotLearnerPrediction() function.




```
          Reference
Prediction  0  1
         0 36  6
         1  5 14

               Accuracy : 0.8197
                 95% CI : (0.7002, 0.9064)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : 0.007899

                  Kappa : 0.5855

 Mcnemar's Test P-Value : 1.000000

            Sensitivity : 0.7000
            Specificity : 0.8780
         Pos Pred Value : 0.7368
         Neg Pred Value : 0.8571
             Prevalence : 0.3279
         Detection Rate : 0.2295
   Detection Prevalence : 0.3115
      Balanced Accuracy : 0.7890

       'Positive' Class : 1
```

Decision tree

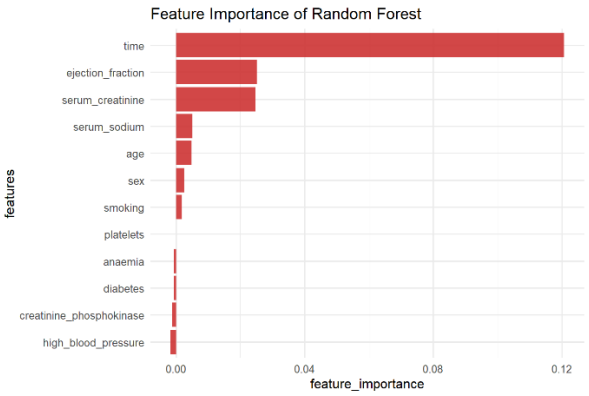
I use CART (Classification and Regression Trees), which always splits a segment into two to create a binary tree. I found when the complexity parameter is 0.03, the decision tree looks simpler and get higher accuracy and recall rate than other values.




```
          Reference
Prediction  0  1
         0 34  4
         1  7 16

               Accuracy : 0.8197
                 95% CI : (0.7002, 0.9064)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : 0.007899

                  Kappa : 0.606

 Mcnemar's Test P-Value : 0.546494

            Sensitivity : 0.8000
            Specificity : 0.8293
         Pos Pred Value : 0.6957
         Neg Pred Value : 0.8947
             Prevalence : 0.3279
         Detection Rate : 0.2623
   Detection Prevalence : 0.3770
      Balanced Accuracy : 0.8146

       'Positive' Class : 1
```

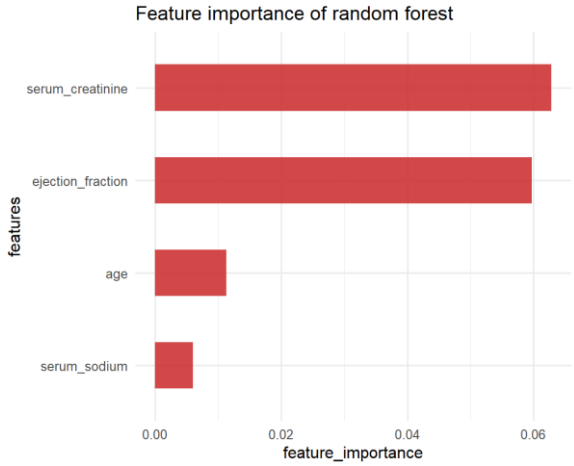
Random Forest

Random Forest is a machine learning model that works by randomly sampling and selecting subsets of data to create decision trees in parallel. These decision trees are used to make predictions on the test data. The final prediction is made by taking the average or majority vote of the predictions from each individual decision tree. To use the Random Forest model, we need to train it on a set of training data and then use it to make predictions on new data.




```
          Reference
Prediction  0  1
         0 36  8
         1  5 12

               Accuracy : 0.7869
                 95% CI : (0.6632, 0.8814)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : 0.03468

                  Kappa : 0.4971

 Mcnemar's Test P-Value : 0.57910

            Sensitivity : 0.6000
            Specificity : 0.8780
         Pos Pred Value : 0.7059
         Neg Pred Value : 0.8182
             Prevalence : 0.3279
         Detection Rate : 0.1967
   Detection Prevalence : 0.2787
      Balanced Accuracy : 0.7390

       'Positive' Class : 1
```

But we can see that the accuracy and recall rate are both unsatisfactory. And I create a simpler model using the same parameter as in the decision tree, but only with the ejection fraction, serum creatinine, serum sodium, and age and without time.




```
          Reference
Prediction  0  1
         0 34  8
         1  7 12

               Accuracy : 0.7541
                 95% CI : (0.6271, 0.8554)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : 0.108

                  Kappa : 0.4348

 Mcnemar's Test P-Value : 1.000

            Sensitivity : 0.6000
            Specificity : 0.8293
         Pos Pred Value : 0.6316
         Neg Pred Value : 0.8095
             Prevalence : 0.3279
         Detection Rate : 0.1967
   Detection Prevalence : 0.3115
      Balanced Accuracy : 0.7146

       'Positive' Class : 1
```

Then the result of this model is worse. By feature culling, we can see that sometimes only considering features with larger weights does not improve the accuracy of the model.