Stats101C Kaggle 2

Prediction of Heart Disease

Group Mahjong

August 2nd 2022

Yuxuan Bai: baiyuxuan1203@gmail.com
Yuetong Li: liyuetong.cathy@gmail.com
Jinghong Zou: jinghongzou@ucla.edu
Xiaocong Xuan: xiaocong0116@gmail.com

Kaggle Score: 0.83928
Kaggle Rank: 8th

1 Introduction

According to the CDC[1], heart disease has been the leading cause of death since 1921, which shows the importance of diagnosing heart disease with a precise rate. Therefore, thousands of lives will be saved if there is a model that predicts whether a patient will get heart disease. The goal of this project is to predict if patients have a heart disease problem (more specifically, whether a patient has greater or less than 50% diameter narrowing) based on their body metrics.

Based on our subjective analysis on which predictors might be crucial in predicting if a patient will get heart disease, we believe that age is an important metric in predicting the result because heart disease is more likely to be diagnosed among older people. According to Harvard Health[2], men were about twice as likely as women to have heart disease, therefore gender is also a significant indicator. Also, people with high blood sugar are more likely to get heart disease. Based on our initial analysis, we can focus on the predictors that are important for prediction.

The techniques we used to predict the result include various classification models like KNN, Random Forest, LDA, QDA, SVM, etc. We constructed a series of analyses to reach our conclusion. It turns out that KNN performs the best in predicting our result, also SVM can be considered for its outstanding performance in the public board score.

2 Exploratory Data Analysis

In order to further investigate which predictors might be the crucial components in predicting diagnosis of heart disease (angiographic disease status), we decided to make several box plots to visualize the potential associations. We made multiple visualizations to investigate any potential associations between *num* and the rest of the predictors. It turned out that we found 10 predictors showing the fairly obvious association.

As shown from the box plots below, we can tell that *num* has association with *age*, *sex*, *cp*, *restecg*, *thalach*, *exang*, *oldpeak*, *slope*, *ca* and *thal*, but we need a further exploration on those predictors by feature selection methods and modeling process. Predictors like *trestbps*, *chol* and *fbs* may not be considered since the graphs of those two predictors show no difference in distribution.

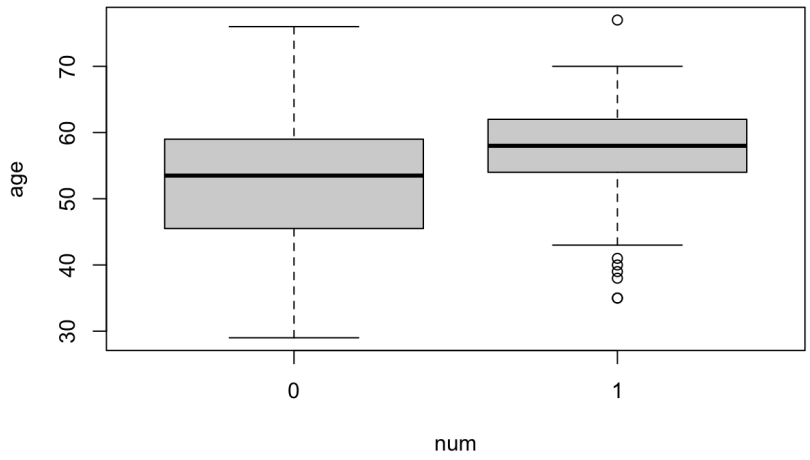

Figure 1: This box plot shows that there is a big difference in distribution between age and num (diagnosis of heart disease), hence this predictor may be significant for further prediction. This indicates older people are more likely to have a diagnosis of heart disease ($> 50\%$ diameter narrowing). This indeed reflects the real world situation.

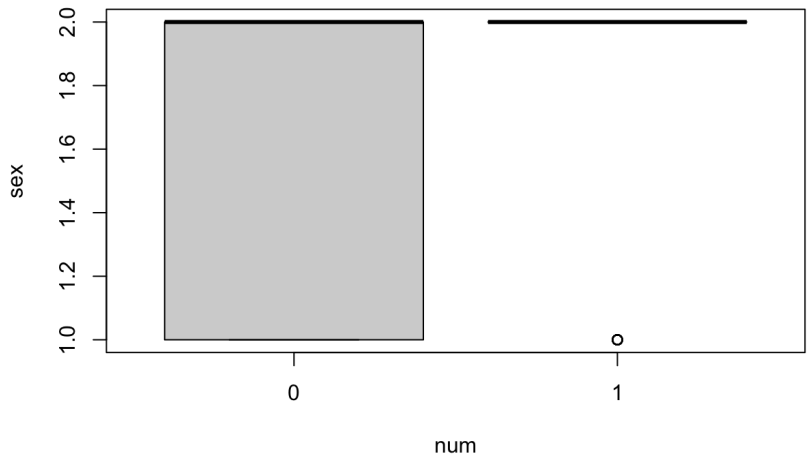

Figure 2: This box plot shows that there is a big difference in distribution between sex and num (diagnosis of heart disease), hence this predictor may be significant for further prediction. This indicates females are less likely to have a diagnosis of heart disease ($< 50\%$ diameter narrowing). This indeed reflects the real world situation.

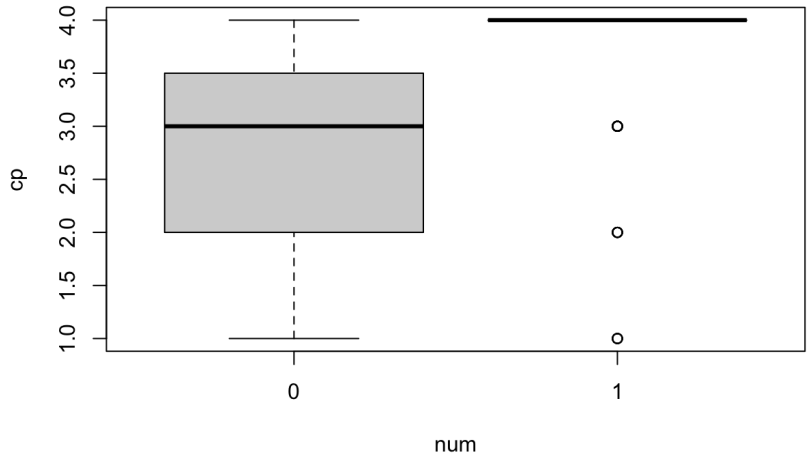

Figure 3: This box plot shows that there is a big difference in distribution between cp (chest pain type) and num (diagnosis of heart disease), hence this predictor may be significant for further prediction. This indicates people who have atypical angina, non-anginal pain, asymptomatic are less likely to have a diagnosis of heart disease ($<$ 50% diameter narrowing).

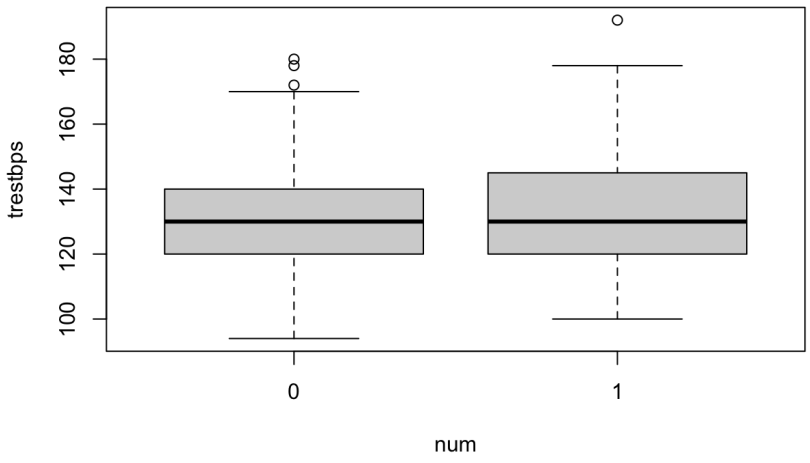

Figure 4: Here is an example of showing there is almost no difference in distribution between trestbps (resting blood pressure) and num (diagnosis of heart disease), hence this predictor might be insignificant in predicting the response variable. This indicates heart disease cannot be diagnosed simply by resting blood pressure

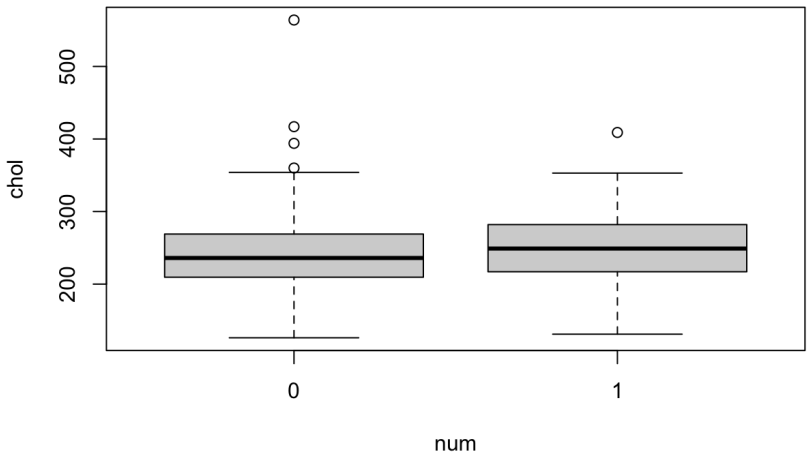

Figure 5: This box plot shows that there is almost no difference in distribution between chol (serum cholesterol in mg/dl) and num (diagnosis of heart disease), hence this predictor seem to be insignificant in predicting our response variable. This indicates heart disease might not be diagnosed simply by serum cholesterol.

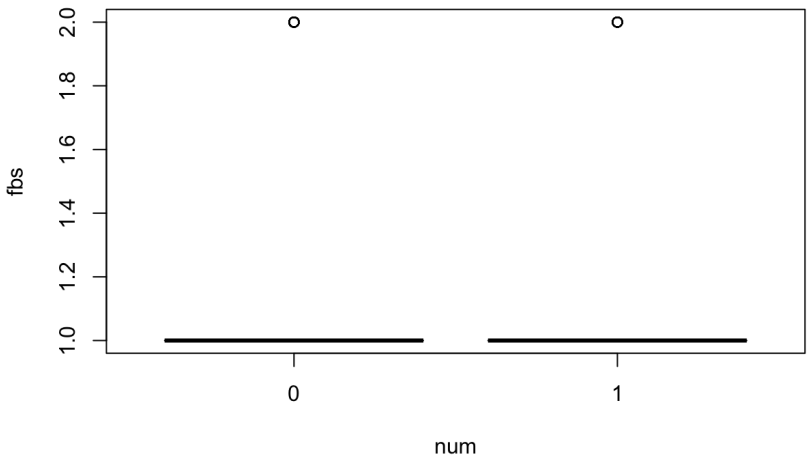

Figure 6: This box plot shows that there is almost no difference in distribution between fbs (fasting blood sugar) and num (diagnosis of heart disease), hence this predictor may be insignificant for further prediction. This indicates heart disease might not be diagnosed simply by fasting blood sugar.

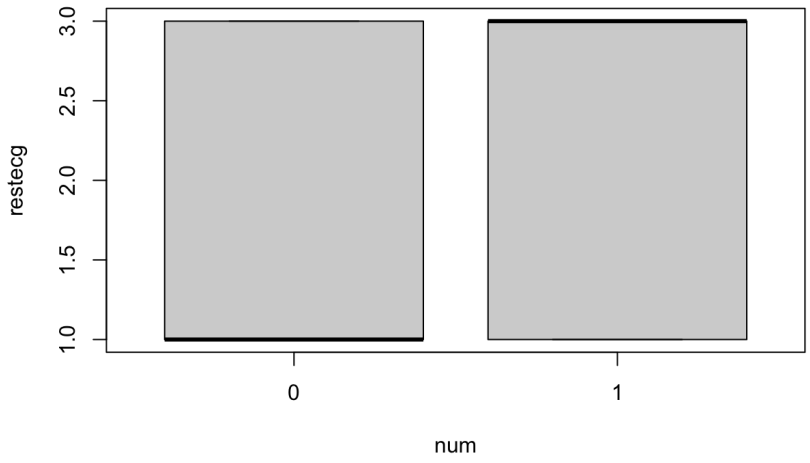

Figure 7: This box plot shows that there is a big difference in distribution between restecg (resting electrocardiographic results) and num (diagnosis of heart disease), hence this predictor may be significant for further prediction. This indicates people who have shown probable or definite left ventricular hypertrophy by Estes' criteria are more likely to have a diagnosis of heart disease ($> 50\%$ diameter narrowing).

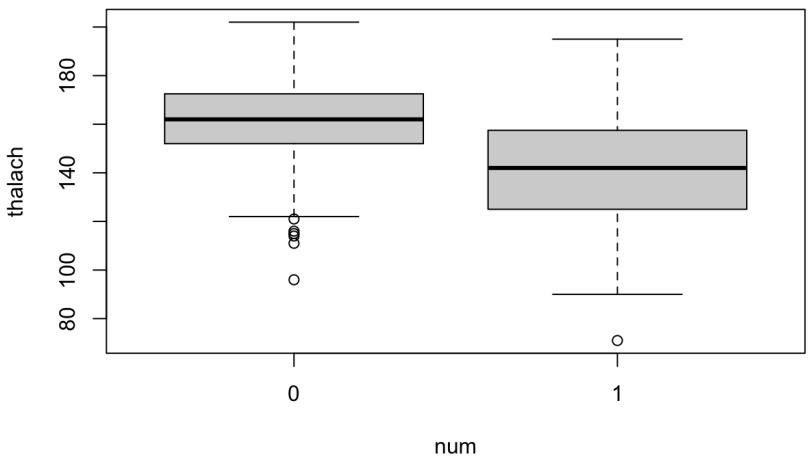

Figure 8: This box plot shows that there is a big difference in distribution between thalach (maximum heart rate achieved) and num (diagnosis of heart disease), hence this predictor may be significant for further prediction. This indicates people who have lower maximum heart rate are more likely to have a diagnosis of heart disease ($> 50\%$ diameter narrowing).

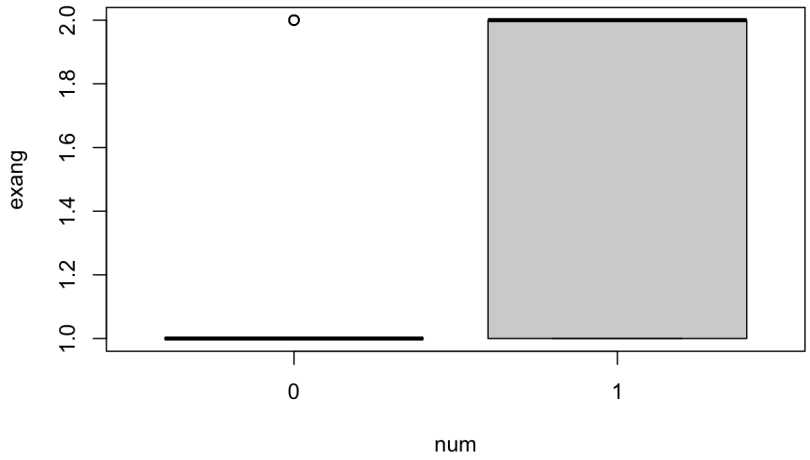

Figure 9: This box plot shows that there is a big difference in distribution between exang (exercise induced angina) and num (diagnosis of heart disease), hence this predictor may be significant for further prediction. This indicates people who have angina induced by exercise are more likely to have a diagnosis of heart disease ($>$ 50% diameter narrowing).

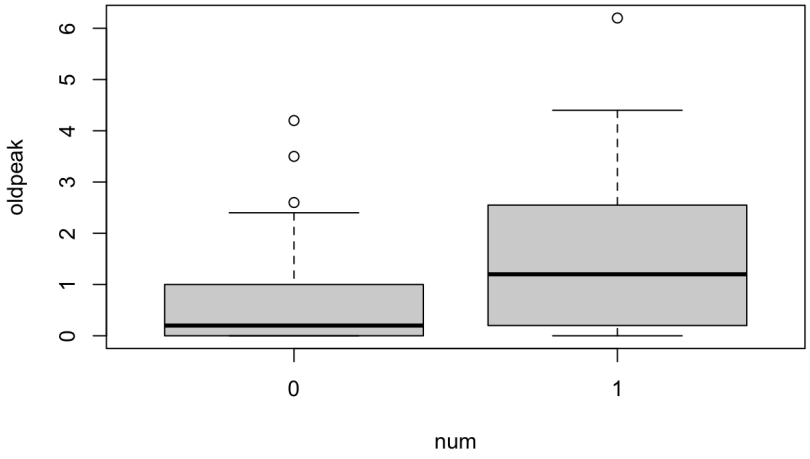

Figure 10: This box plot shows that there is a big difference in distribution between oldpeak (ST depression induced by exercise relative to rest) and num (diagnosis of heart disease), hence this predictor may be significant for further prediction. This indicates people who have higher ST depression induced by exercise relative to rest are more likely to have a diagnosis of heart disease ($>$ 50% diameter narrowing).

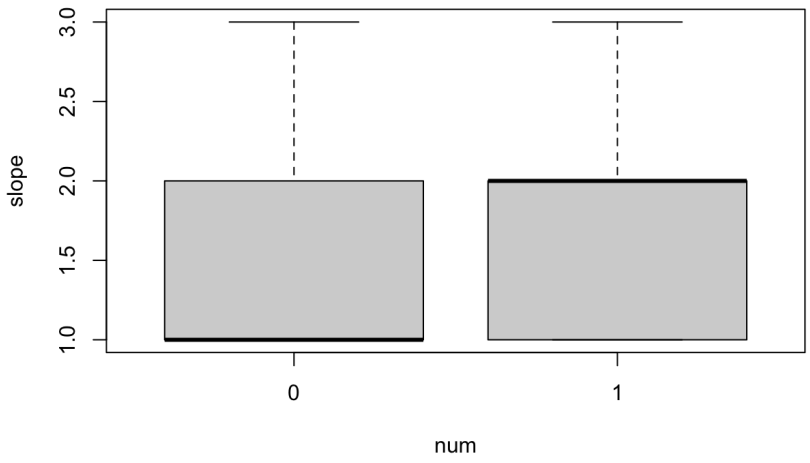

Figure 11: This box plot shows that there is a big difference in distribution between slope (the slope of the peak exercise ST segment) and num (diagnosis of heart disease), hence this predictor may be significant for further prediction. This indicates people who have a diagnosis of heart disease ($>$ 50% diameter narrowing) tend to have flat and downsloping peak exercise ST segment.

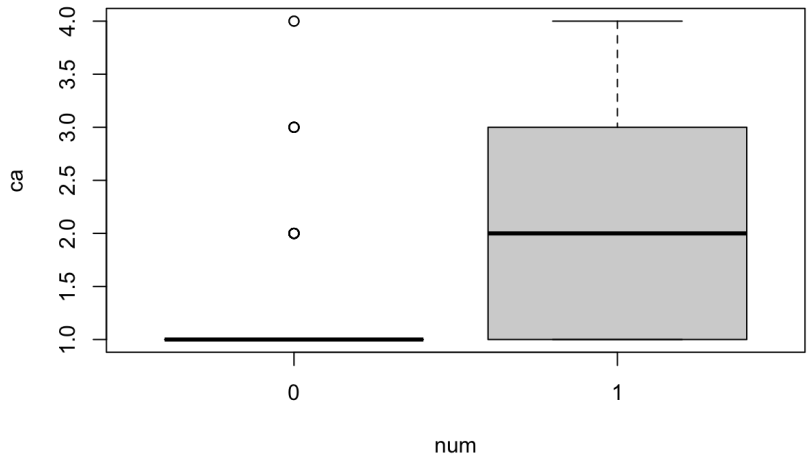

Figure 12: This box plot shows that there is a big difference in distribution between ca (number of major vessels (0-3) colored by fluoroscopy) and num (diagnosis of heart disease), hence this predictor may be significant for further prediction. This indicates people who have more number of major vessels colored by fluoroscopy are more likely to have a diagnosis of heart disease ($>$ 50% diameter narrowing).

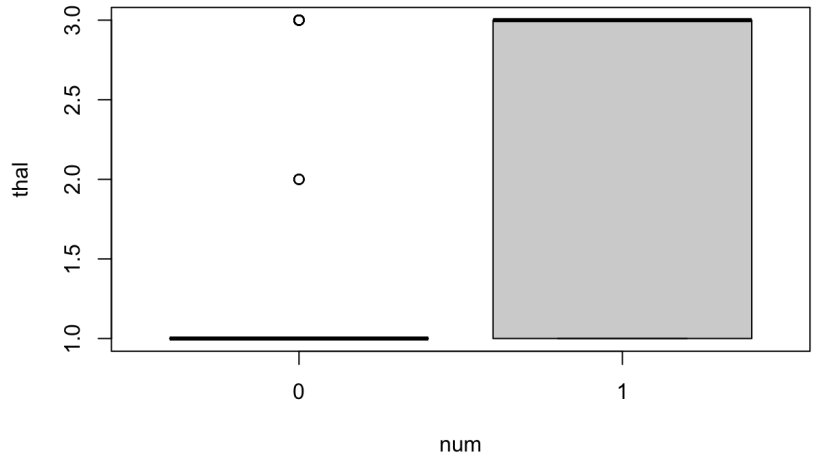

Figure 13: This box plot shows that there is a big difference in distribution between thal (1 = normal; 2 = fixed defect; 3= reversible defect) and num (diagnosis of heart disease), hence this predictor may be significant for further prediction. This indicates people who have 2nd and 3rd level of thal are more likely to have a diagnosis of heart disease (> 50% diameter narrowing).

3 Preprocessing

(1). Convert all categorical predictors to factor type
During the data cleaning process, we found that some predictors like sex, resting electrocardiographic result are in type of integer, which is meaningless while doing statistically analysis because those predictors, in reality, are categorical type of predictor. Therefore, we transformed predictors including, sex, cp, fbs, restecg, exang, slope, thal, ca, and num to factor data type for reasonable data analysis.

(2). Eliminate abnormal observations in the training and testing data set.
In the training data set, we noticed that there are several observations that obtain "?" value in predictors like ca. With "?" in our data, the corresponding predictors will be impacted and result in a failure in modeling. Hence, we removed the observations that contain the "?" value in our training data set.

In the testing data set, we found the same thing that ca contains "?" value. If we keep the "?" value in the testing data set after applying our classification model, we would end up getting 74 observations due to the reason that observations that contain "?" will not produce the output. Therefore, we came up with an idea that replacing all "?" value with the mode of the data, which is "0".

(3). Remove insignificant variables
In order to distinguish which predictor can be used to predict the result, we managed to produce boxplots of num against each predictors (see Part 2: Exploratory Data Analysis).

As we analyzed in Part 2, we decided to remove the trestbps, chol, and fbs due to the identicality in distribution from their boxplot, which shows the irrelevance in producing the result.

(4). Normalize all numerical data
To have a better modeling process and reduce fluctuation in accuracy, we decided to standardize all numerical data. This can help us to adjust the scale.

4 Candidate models

Model identifier	Type of model	Engine	Hyperparameters
KNN_model	classification	kknn	neighbors
SVM_model	Support Vector Machine	kernlab	cost, rbf_sigma
rf_model	Random Forest	ranger	mtry, trees, min_n
log_model	Logistic Regression	glm	N/A

We eventually selected four models including logistic regression, random forest, KNN, and SVM. KNN performs the best from the cross-validation process, and SVM performs the best in the public scoreboard. We listed these four models in the chart above, including engines and hyperparameters.

For the table below, we listed all the variables that are selected during section 3, we ended up getting 10 variables to fit the model.

Model identifier	Variables
KNN_model	"age" "sex" "cp" "restecg" "thalach" "exang" "oldpeak" "slope" "ca" "thal"
rf_model	"age" "sex" "cp" "restecg" "thalach" "exang" "oldpeak" "slope" "ca" "thal"
log_model	"age" "sex" "cp" "restecg" "thalach" "exang" "oldpeak" "slope" "ca" "thal"
SVM_model	"age" "sex" "cp" "restecg" "thalach" "exang" "oldpeak" "slope" "ca" "thal"

Candidate model 1: Random Forest in Classification
Random Forest Classification algorithm is one of the most accurate modeling methods available. It can produce many decision trees to make sub-decisions based on the random sampled data set. It uses packaging and features randomness while producing each tree branch and then creates an uncorrelated forest of trees that performs better than any other trees created.

Candidate model 2: Logistic Regression
Logistic regression is a statistical model that is regularly used for classification analytics, it estimates the probability of an event happening, and the outputs are the event happened

or the event did not happen. Since our output contains two results, the output is usually denoted as "0" and "1". Logistic regression uses a different transformation called logit transformation, the core idea is odds, the probability of success divided by the probability of failure (this is also called log odds). Generally, for binary classification, after the logistic model is applied, a probability less than 50% will output "0", while probability greater than 50% will output "1".

Candidate model 3: Support Vector Machine (SVM)
SVM is a supervised learning method that is used for classification and regression, in this project we will focus on classification. SVM allows the user to produce a hyperplane in an N-dimensional field (N denotes the number of predictors in our dataset) which can be used to distinguish our data. The hyperplane can help us to separate the two categories of data, therefore, we have an infinite many possible selections of the hyperplane. What we were looking for is the hyperplane that maximized the distance between the data of the both classes, which can provide us some assurance that the future observations can be classified with more confidence.

Candidate model 4: K-Nearest Neighbors (KNN)
K-Nearest Neighbors Algorithm is a non-parametric supervised learning method that uses proximity to make classification based on the grouping of data observations. It is commonly used in the classification problems. Basically, data will be categorized as the same group if the surrounding of one observation is occupied by a group that has the most probability. The way to identify if an observation is close to another is using Euclidean distance.

5 Model evaluation and tuning

Tuning of hyperparameters:

Model identifier	hyperparameters
KNN_model	k: 14
SVM_model	cost: 2.902571, rbf_sigma: 0.002395448
rf_model	mtry:1, trees:1131, min_n:8
log_model	N/A

1. Random Forest in Classification:
We tuned the selected hyperparameters: mtry, ntree, and min_n. Here, 'min_ n' denotes an integer for the minimum number of data points in a node that is required for the node to be split further; 'ntree' denotes an integer for the number of trees contained in the ensemble;

and ‘mtry‘ represents an integer for the number of predictors that will be randomly sampled at each split when creating the tree models. The tuning result gives us that the best ‘min_n‘ should be 8, ‘mtry‘ should be 1, and ‘trees‘ should be 1131. Gaining the best value for these three hyperparameters is able to help us construct a final random forest model by using the tidymodels. To measure the performance of this model, we used v-fold cross validation with setting: ‘set.seed(123)‘, fold number (v=7), and the mean of f scores is 0.8256708.

2. Logistic Regression:
Since there is no hyperparameter in Logistic Regression, so we will not consider the tuning part. We used set.seed(123), set the size of cross-validation folds to 7, used the logistic_reg() function. After fitting the model, we get the mean of 7 f scores from the created folds of 0.8330251 with set.seed(123).

3. Support Vector Machine (SVM):
We tuned the selected hyperparameters: cost, and rbf_sigma. Here, ‘cost‘ denotes a positive number for the cost of predicting a sample within or on the wrong side of the margin; ‘rbf_sigma‘ denotes a positive number for radial basis function. The tuning result gives us that the best ‘cost‘ should be 2.902571 and ‘rbf_sigma‘ should be 0.002395448. Gaining the best value for these two hyperparameters is able to help us construct a final SVM model by using the tidymodels. To measure the performance of this model, we used v-fold cross validation with setting: ‘set.seed(123)‘, fold number (v=7), and the mean of f scores is 0.826195.

4. K-Nearest Neighbors (KNN):
We tuned the selected hyperparameters: neighbors. Here, ‘neighbors‘ denotes a single integer for the number of neighbors to consider. The tuning result gives us that the best ‘k‘ should be 14. Gaining the best value for this hyperparameters is able to help us construct a final KNN model by using the tidymodels. To measure the performance of this model, we used v-fold cross validation with setting: ‘set.seed(123)‘, fold number (v=7), and the mean of f scores is 0.843528.

The performance of each model:

Model identifier	mean_f_score
KNN_model	0.843528
SVM_model	0.826195
log_model	0.8330251
rf_model	0.8256708

According to Figure 14, we can conclude that KNN_model have higher quantiles of f score. It means that models that are constructed by KNN perform better in this dataset. Hence, we decided to use KNN_model as our final model. Furthermore, SVM_model has a higher

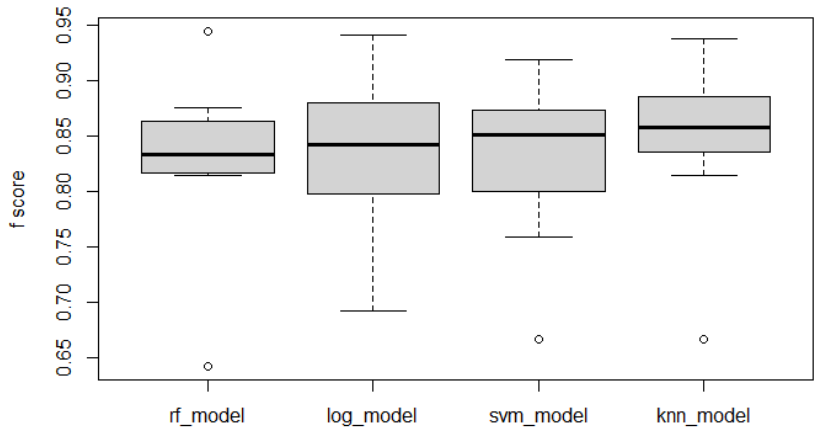

Figure 14: The distribution of rmse of the Four Models

median compared to rf_model and log_model, also SVM_model with tuned parameter performs well on the public scoreboard (SVM scored 0.9 in the leaderboard).

6 Discussion of final model

For the final two models, we decided to select the Support Vector Machine (SVM) model and K-Nearest Neighbor model. As for the KNN model, we are confident to select the best hyperparameters because we tested many values of nearest neighbors from 1 to 100, each neighbor will create 5 folds to get the mean of the f score. Besides, according to the final Kaggle score, our KNN model performs better compared to SVM model. This process will increase the chance of getting the best value of the nearest neighbor. As for SVM model, according to the boxplot comparison of their f scores, it has a higher median f score compared to logistic regression and random forest. Also, another reason for us to select SVM is that this model performed the best in the public scoreboard, scored 0.9 and made us the first place.

The advantage of KNN is that it has a very easy implementation, it only requires the user to input the number of nearest neighbors. Also, KNN is resilient regarding the search space (categories do not need to be linearly separable), also the classifier can be easily updated at the minimum cost. However, there is no such thing as a perfect model. When dealing with a data set with a large number of observations, KNN can be computationally expensive, which degrades the performance of the model. Also, KNN would not work well with the dataset that has more predictors than observations as it is complicated for the model to compute

the defined distance in each dimension; Moreover, KNN needs variable standardization to perform well, if the scales are not standardized, KNN may generate inaccurate predictions. Some of the possible improvements can be applied by doing dimension reduction techniques to reduce the numbers of predictors, removing more insignificant predictors, or removing all abnormal observations that contain NA values as KNN is sensitive to that.

The advantage of SVM is that it performs quite well when dealing with data that has clear margins between the classes; also it is suitable to process the dataset in higher dimensional spaces. It is a complement of KNN as KNN is poor at dealing with the dataset with higher dimensions. However, SVM performs poorly while dealing with large data sets; for datasets with overlapping classes, SVM performs poorly; also there is no probabilistic interpretation for the classification model; compared to Random Forest model, SVM does not have a self-optimized mechanism to adjust the input hyperparameters, hence deciding the proper parameters is crucial in getting the accurate results. Some of the improvements that we can adopt are doing a more precise parameter tuning process, and doing data wrangling to make classes more distinct and interpretable.

For our modeling process, we did not consider the interaction between variables, but in the real world situation, it is highly possible that two of the predictors might have mutual effects on the output or other variables, which might change the significance of the other variables drastically. If more ample time is given for this project, we would do more diagnoses on investigating the interaction effects between variables.

7 Appendix: Final annotated script

7 Appendix: Final annotated script

```{r}
#load data
heart_test <- read.csv("heart_test.csv")
heart_train <- read.csv("heart_train.csv")
```

```{r}
#load related libraries
library(tidyverse)
library(tidymodels)
library(ISLR2)
tidymodels_prefer()
library(dplyr)
library(ggplot2)
library(grid)
library(kernlab)
library(kknn)
library(tune)
library(MASS)
library(discrim)
```

```{r}
#clean data
heart_train$thal <- suppressWarnings(as.numeric(heart_train$thal))
heart_train$ca <- suppressWarnings(as.numeric(heart_train$ca))
na <- c(which(is.na(heart_train$thal)), which(is.na(heart_train$ca)))
heart_train <- heart_train[-na,]
```

```{r}
#convert all categorical variables to factor type in training set
heart_train$sex <- as.factor(heart_train$sex)
heart_train$cp <- as.factor(heart_train$cp)
heart_train$fbs <- as.factor(heart_train$fbs)
heart_train$restecg <- as.factor(heart_train$restecg)
heart_train$exang <- as.factor(heart_train$exang)
heart_train$slope <- as.factor(heart_train$slope)
heart_train$thal <- as.factor(heart_train$thal)
heart_train$ca <- as.factor(heart_train$ca)
heart_train$num <- as.factor(heart_train$num)

#convert all categorical variables to factor type in testing set
heart_test$sex <- as.factor(heart_test$sex)
heart_test$cp <- as.factor(heart_test$cp)
heart_test$fbs <- as.factor(heart_test$fbs)
heart_test$restecg <- as.factor(heart_test$restecg)
heart_test$exang <- as.factor(heart_test$exang)
heart_test$slope <- as.factor(heart_test$slope)
heart_test$thal <- as.factor(heart_test$thal)
heart_test$ca <- as.factor(heart_test$ca)


#select predictors, deleted irrelevant trestbps, chol, fbs
heart_train <- heart_train[,-c(5,6,7)]
heart_test <- heart_test[,-c(5,6,7)]
```

KNN

```{r}
#create the recipe, normalize all numerical variables
heart_recipe <- recipe(num ~., data = heart_train) %>%
step_normalize(all_numeric(), -all_outcomes())

#create workflow
wf <- workflow() %>%
  add_recipe(heart_recipe)

#tuning number of neighbors
set.seed(3)
folds <- vfold_cv(heart_train, v = 5, repeats = 1)

#tuning process, set number of neighbors as tuning part
tune_spec <- nearest_neighbor(neighbors = tune()) %>%
  set_mode("classification") %>%
  set_engine("kknn")

#set k = 1 to 100
neighbor_grid <- expand.grid(neighbors = seq(1,100, by = 1))

#tuning
knn_grid <- tune_grid(
  wf %>% add_model(tune_spec),
  resamples = folds,
  grid = neighbor_grid,
  metrics = metric_set(f_meas),
  control = control_grid(save_pred = TRUE,
                         verbose = TRUE)
)

#f_score of all five folds
knn_grid %>%
  collect_metrics() %>% filter(.metric == "f_meas")

#the best k = 14 with mean of f score of 0.8364736
knn_grid %>%
  select_best("f_meas")
```

SVM
```{r}
# reference:
https://r4ds.github.io/bookclub-tmwr/svm-model-as-motivating-example.html
#create the recipe
heart_recipe <- recipe(num ~., data = heart_train) %>%
step_normalize(all_numeric(), -all_outcomes())

#tune the hyperparameters
svm_spec <-
  svm_rbf(cost = tune(), rbf_sigma = tune()) %>%
  set_mode("classification") %>%
  set_engine("kernlab")

#create workflow
svm_wflow <-
  workflow() %>%
  add_model(svm_spec) %>%
  add_recipe(heart_recipe)

#create folds
```

```
set.seed(99)
heart_folds <- vfold_cv(heart_train, v = 7)


tune_res <- tune_grid(
  svm_wflow,
  resamples = heart_folds,
  metrics = metric_set(f_meas),
  grid = 30
)

#collect all f_meas
tune_res %>%
  collect_metrics() %>%
  filter(.metric == "f_meas")

#best hyperparameter are 2.902571    0.002395448, mean f score is 0.8302179
tune_res %>%
  select_best("f_meas")
```

```{r}
#generate final output
knn_model <- nearest_neighbor(neighbors = 14) %>%
  set_mode("classification") %>%
  set_engine("kknn")

svm_model <- svm_rbf(cost = 2.902571 , rbf_sigma = 0.002395448) %>%
  set_mode("classification") %>%
  set_engine("kernlab")

final_knn <- workflow() %>%
  add_model(knn_model) %>%
  add_formula(num ~ .)

final_svm <- workflow() %>%
  add_model(svm_model) %>%
  add_formula(num ~ .)

#replace ? as 0(mode)
heart_test$ca[60] <- 0
heart_test$ca[9] <- 0

p_knn<-fit(final_knn, heart_train)
r_knn<-predict(p_knn, heart_test)

p_svm<-fit(final_svm, heart_train)
r_svm<-predict(p_svm, heart_test)


ans_knn <- cbind(Id=heart_test$id, Predicted=as.numeric(r_knn[[1]])-1)
ans_svm <- cbind(Id=heart_test$id, Predicted=as.numeric(r_svm[[1]])-1)
write.csv(ans_knn, "knn.csv", row.names = F)
write.csv(ans_svm, "svm.csv", row.names = F)
```

8 Appendix: Team member contributions

Yuxuan Bai:
1. write Part 3 and 4 of the report
2. perform modeling process.
3. use v-fold cross validation to measure the performance of the candidate models

Yuetong Li:
1. construct the KNN, Logistic Regression model
2. perform data cleaning
3. write the report Part 1 and 2

Jinghong Zou:
1. construct the random forest model
2. tune the all hyperparameters
3. write the report Part 4 and Part 5

Xiaocong Xuan:
1. construct the SVM model
2. tune the hyperparameters
3. write the report Part 6
4. provide final report template and finalize the final report

9 Reference

[1] Central Disease Control. *Throughout life, heart attacks are twice as common in men than women*

https://www.cdc.gov/mmwr/preview/mmwrhtml/mm4830a1.htm

[2] Harvard Health Publishing. *Achievements in Public Health, 1900-1999: Decline in Deaths from Heart Disease and Stroke – United States, 1900-1999*

https://www.health.harvard.edu/heart-health/throughout-life-heart-attacks-are-twice-as-
~:text=Researchers%20found%20that%20throughout%20life,mass%20index%2C%20and%
20physical%20activity.