### 加载??? ###

library(tidyverse)

library(klaR)

library(kernlab)

library(caret)

library(pROC)

### 载入数据 ###

train <- read\_csv("train.csv")

test <- read\_csv("test.csv")

validation <- read\_csv("validation.csv")

### 将test集的变量与train和valid一??? ###

train <- train[, colnames(test)]

validation <- validation[, colnames(test)]

### 使用管道符定义变??? ####

train <- train %>%

mutate\_at(-3, as.factor)

test <- test %>%

mutate\_at(-3, as.factor)

validation <- validation %>%

mutate\_at(-3, as.factor)

### 设置种子??? ###

seed <- 1234

### 随机森林 ###

rf\_train <- train(outcome ~ ., data = train, method = "rf")

rf\_train$finalModel

### 内部验证集valid ###

rf\_pred\_valid <- predict(rf\_train, newdata = validation)

rf\_cm\_valid <- confusionMatrix(validation$outcome, rf\_pred\_valid, positive = "1")

rf\_cm\_valid

### 外部验证集test ###

rf\_pred <- predict(rf\_train, newdata = test)

rf\_cm <- confusionMatrix(test$outcome, rf\_pred, positive = "1")

rf\_cm

### 计算ROC ###

rf\_pred\_prob <- predict(rf\_train, newdata = test, type = "prob")

rf\_roc <- roc(test$outcome, rf\_pred\_prob[,2])

rf\_roc

### 决策???##递归分割???####

rp\_train <- train(outcome ~ ., data = train, method = "rpart")

rp\_train$finalModel

rp\_pred\_valid <- predict(rp\_train, newdata = validation)

rp\_cm\_valid <- confusionMatrix(validation$outcome, rp\_pred\_valid, positive = "1")

rp\_cm\_valid

rp\_pred <- predict(rp\_train, newdata = test)

rp\_cm <- confusionMatrix(test$outcome, rp\_pred, positive = "1")

rp\_cm

rp\_pred\_prob <- predict(rp\_train, newdata = test, type = "prob")

rp\_roc <- roc(test$outcome, rp\_pred\_prob[,2])

rp\_roc

### 贝叶??? ###

nb\_train <- train(outcome ~ ., data = train, method = "nb")

nb\_train$finalModel

nb\_pred\_valid <- predict(nb\_train, newdata = validation)

nb\_cm\_valid <- confusionMatrix(validation$outcome, nb\_pred\_valid, positive = "1")

nb\_cm\_valid

nb\_pred <- predict(nb\_train, newdata = test)

nb\_cm <- confusionMatrix(test$outcome, nb\_pred, positive = "1")

nb\_cm

nb\_pred\_prob <- predict(nb\_train, newdata = test, type = "prob")

nb\_roc <- roc(test$outcome, nb\_pred\_prob[,2])

nb\_roc

### KNN ###

knn\_train <- train(outcome ~ ., data = train, method = "knn")

knn\_train$finalModel

knn\_pred\_valid <- predict(knn\_train, newdata = validation)

knn\_cm\_valid <- confusionMatrix(validation$outcome, knn\_pred\_valid, positive = "1")

knn\_cm\_valid

knn\_pred <- predict(knn\_train, newdata = test)

knn\_cm <- confusionMatrix(test$outcome, knn\_pred, positive = "1")

knn\_cm

knn\_pred\_prob <- predict(knn\_train, newdata = test, type = "prob")

knn\_roc <- roc(test$outcome, knn\_pred\_prob[,2])

knn\_roc

### SVM ###

svm\_train <- train(outcome ~ ., data = train %>%

mutate(outcome = make.names(outcome)), method = "svmLinear",

trControl = trainControl(method = "repeatedcv", repeats = 5,

classProbs = TRUE))

svm\_train$finalModel

svm\_pred\_valid <- predict(svm\_train, newdata = validation) %>%

fct\_relabel(~{c("0", "1")})

svm\_cm\_valid <- confusionMatrix(validation$outcome, svm\_pred\_valid, positive = "1")

svm\_cm\_valid

svm\_pred <- predict(svm\_train, newdata = test) %>%

fct\_relabel(~{c("0", "1")})

svm\_cm <- confusionMatrix(test$outcome, svm\_pred, positive = "1")

svm\_cm

svm\_pred\_prob <- predict(svm\_train, newdata = test, type = "prob")

svm\_roc <- roc(test$outcome, svm\_pred\_prob[,2])

svm\_roc

### 逻辑回归 ###

glm\_train <- train(outcome ~ ., data = train, method = "glm")

glm\_train$finalModel

glm\_pred\_valid <- predict(glm\_train, newdata = validation)

glm\_cm\_valid <- confusionMatrix(validation$outcome, glm\_pred\_valid, positive = "1")

glm\_cm\_valid

glm\_pred <- predict(glm\_train, newdata = test)

glm\_cm <- confusionMatrix(test$outcome, glm\_pred, positive = "1")

glm\_cm

glm\_pred\_prob <- predict(glm\_train, newdata = test, type = "prob")

glm\_roc <- roc(test$outcome, glm\_pred\_prob[,2])

glm\_roc

### Summary ###

summ <- list(rf\_cm, rp\_cm, nb\_cm, knn\_cm, svm\_cm, glm\_cm)

names <- c("Random Forest", "Decision Tree", "Naive Bayes", "KNN", "SVM", "Logistic")

summ\_tab <- map2\_dfr(summ, names, ~{

data.frame(Model = .y,

Accuracy = .x$overall[1],

Sensitivity = .x$byClass[1],

Specificity = .x$byClass[2])

})

summ\_tab

write\_tsv(summ\_tab, "summary\_table.txt")

### Correlation heatmap ###

train[, -1] %>%

mutate\_all(as.numeric) %>%

cor() %>%

as.data.frame() %>%

rownames\_to\_column("y") %>%

pivot\_longer(-y, names\_to = "x") %>%

ggplot(aes(x, y)) +

geom\_tile(aes(fill = value)) +

scale\_fill\_distiller(palette = "RdYlBu", name = NULL) +

scale\_y\_discrete(limit = rev, expand = expansion(c(0, 0))) +

scale\_x\_discrete(expand = expansion(c(0, 0))) +

theme\_minimal() +

coord\_equal() +

theme(axis.title = element\_blank(),

axis.text = element\_text(size = rel(1.2),

colour = "black",

face = "bold"),

axis.ticks = element\_line(size = 0.8),

axis.text.x = element\_text(angle = 90,

hjust = 1,

vjust = 0.5),

legend.key.height = unit(2.1, "cm"))

ggsave("pearson\_heatmap.pdf", width = 6, height = 5)

train[, -1] %>%

mutate\_all(as.numeric) %>%

cor(method = "spearman") %>%

as.data.frame() %>%

rownames\_to\_column("y") %>%

pivot\_longer(-y, names\_to = "x") %>%

ggplot(aes(x, y)) +

geom\_tile(aes(fill = value)) +

scale\_fill\_distiller(palette = "RdYlBu", name = NULL) +

scale\_y\_discrete(limit = rev, expand = expansion(c(0, 0))) +

scale\_x\_discrete(expand = expansion(c(0, 0))) +

theme\_minimal() +

coord\_equal() +

theme(axis.title = element\_blank(),

axis.text = element\_text(size = rel(1.2),

colour = "black",

face = "bold"),

axis.ticks = element\_line(size = 0.8),

axis.text.x = element\_text(angle = 90,

hjust = 1,

vjust = 0.5),

legend.key.height = unit(2.1, "cm"))

###保存??? pdf格式 ###

ggsave("spearman\_heatmap.pdf", width = 6, height = 5)

### ROC曲线绘制 ###

rocs <- list(rf\_roc, rp\_roc, nb\_roc, knn\_roc, svm\_roc, glm\_roc)

names <- c("Random Forest", "Decision Tree", "Naive Bayes", "KNN", "SVM", "Logistic")

plot\_df <- map2\_dfr(rocs, names, ~{

data.frame(Sensitivity = rev(.x$sensitivities),

Specificity = rev(.x$specificities)) %>%

mutate(Model = .y,

AUC = as.numeric(.x$auc))

}) %>%

mutate(Specificity = 1 - Specificity)

auc\_df <- plot\_df %>%

dplyr::select(Model, AUC) %>%

distinct() %>%

add\_row(Model = "AUC:", .before = 1) %>%

mutate(y = seq(0.5, 0.1, length.out = 7))

plot\_df %>%

ggplot(aes(x = Specificity, y = Sensitivity)) +

geom\_abline(slope = 1) +

geom\_line(aes(group = Model, color = Model), size = 0.8) +

scale\_color\_brewer(palette = "Spectral") +

labs(x = "1 - Specificity") +

annotate("text", x = 0.8, y = auc\_df$y, label = auc\_df$Model, hjust = 1) +

annotate("text", x = 0.83, y = auc\_df$y[-1], label = sprintf("%.2f", auc\_df$AUC[-1]), hjust = 0) +

theme\_bw() +

coord\_equal() +

theme(axis.text = element\_text(size = rel(1.2),

colour = "black",

face = "bold"),

axis.title = element\_text(size = rel(1.2),

colour = "black",

face = "bold"),

legend.text = element\_text(size = rel(1.1),

colour = "black",

face = "bold"),

legend.title = element\_text(size = rel(1.1),

colour = "black",

face = "bold"),

panel.grid = element\_blank())

ggsave("roc.pdf", width = 8, height = 6)