d1e.task = as\_task\_classif(

x=data1\_clin,

target="pfi",

id="Breast cancer PFI predicted by protein abundance")

set.seed(43)

dat.parts = partition(d1e.task, 4/5)

d1e = lrn(

"classif.cv\_glmnet",

alpha=0.5,

predict\_type="prob")

#train the model

d1e$train(d1e.task, dat.parts$train)

preds <- d1e$predict(d1e.task, dat.parts$test) # test the model in test data

confusion\_matrix <- preds$confusion

print(confusion\_matrix)

# evaluate the performance of this learner using cross-validation

rrd1e = resample(d1e.task, d1e, rsmp("cv",folds=3))

metrics = c("auc","acc","sensitivity","specificity",

"precision","recall","fbeta")

metrics = paste("classif", metrics, sep=".")

metrics = lapply(metrics, msr)

rrd1e$aggregate(metrics)

# generate the plot

roc\_d1e <- autoplot(rrd1e, type = "roc") +

theme\_minimal() + # 使用简洁主题作为基础

theme(

panel.background = element\_rect(fill = "white", colour = "white"),

plot.background = element\_rect(fill = "white", colour = "white")

)

roc\_d1e <- roc\_d1e + ggtitle("Elastic Net ROC Curve with Clinic Data")

auc\_d1e <- autoplot(rrd1e, measure=msr("classif.auc"))

ggsave(filename = "visulization/result of elastic net with clinic data.png",

plot = roc\_d1e,

width = 10, height = 8,

units = "in", dpi = 300)

#####use data3

d3e.task = as\_task\_classif(

x=data3\_clin\_prot,

target="pfi",

id="Breast cancer PFI predicted by protein abundance")

set.seed(43)

dat.parts = partition(d3e.task, 3/4)

d3e = lrn(

"classif.cv\_glmnet",

alpha=0.5,

predict\_type="prob")

#train the model

d3e$train(d3e.task, dat.parts$train)

preds <- d3e$predict(d3e.task, dat.parts$test) # test the model in test data

confusion\_matrix <- preds$confusion

print(confusion\_matrix)

# evaluate the performance of this learner using cross-validation

rrd3e = resample(d3e.task, d3e, rsmp("cv",folds=3))

metrics = c("auc","acc","sensitivity","specificity",

"precision","recall","fbeta")

metrics = paste("classif", metrics, sep=".")

metrics = lapply(metrics, msr)

rrd3e$aggregate(metrics)

roc\_d3e <- autoplot(rrd3e, type = "roc") +

theme\_minimal() + # 使用简洁主题作为基础

theme(

panel.background = element\_rect(fill = "white", colour = "white"),

plot.background = element\_rect(fill = "white", colour = "white")

)

roc\_d3e <- roc\_d3e + ggtitle("Elastic Net ROC Curve with Clinic and Protein Data")

auc\_d3e <- autoplot(rrd3e, measure=msr("classif.auc"))

ggsave(filename =

"visulization/result of elastic net with clinic data and protain data.png",

plot = roc\_d3e,

width = 10, height = 8,

units = "in", dpi = 300)

########use data5

d5e.task = as\_task\_classif(

x=data5\_clin\_mirna,

target="pfi",

id="Breast cancer PFI predicted by mirna data and clinical data")

set.seed(43)

dat.parts = partition(d5e.task, 3/4)

d5e = lrn(

"classif.cv\_glmnet",

alpha=0.5,

predict\_type="prob")

#train the model

d5e$train(d5e.task, dat.parts$train)

preds <- d5e$predict(d5e.task, dat.parts$test) # test the model in test data

confusion\_matrix <- preds$confusion

print(confusion\_matrix)

# evaluate the performance of this learner using cross-validation

rrd5e = resample(d5e.task, d5e, rsmp("cv",folds=3))

metrics = c("auc","acc","sensitivity","specificity",

"precision","recall","fbeta")

metrics = paste("classif", metrics, sep=".")

metrics = lapply(metrics, msr)

rrd5e$aggregate(metrics)

roc\_d5e <- autoplot(rrd5e, type = "roc") +

theme\_minimal() + # 使用简洁主题作为基础

theme(

panel.background = element\_rect(fill = "white", colour = "white"),

plot.background = element\_rect(fill = "white", colour = "white")

)

roc\_d5e <- roc\_d5e + ggtitle("Elastic Net ROC Curve with Clinic and miRNA Data")

auc\_d5e <- autoplot(rrd5e, measure=msr("classif.auc"))

ggsave(filename =

"visulization/result of elastic net with clinic data and mirna data.png",

plot = roc\_d5e,

width = 10, height = 8,

units = "in", dpi = 300)

########use data4 clinic and mutation data in elastic net

d4.task = as\_task\_classif(

x = data4\_clin\_mut,

target = "pfi",

id = "Breast cancer PFI predicted by clinic data and mutation data"

)

set.seed(43)

dat.parts = partition(d4.task, 3/4)

d4e = lrn(

"classif.cv\_glmnet",

alpha=0.5,

predict\_type="prob")

#train the model

d4e$train(d4.task, dat.parts$train)

preds <- d4e$predict(d4.task, dat.parts$test) # test the model in test data

confusion\_matrix <- preds$confusion

print(confusion\_matrix)

# evaluate the performance of this learner using cross-validation

rrd4e = resample(d4.task, d4e, rsmp("cv",folds=3))

metrics = c("auc","acc","sensitivity","specificity",

"precision","recall","fbeta")

metrics = paste("classif", metrics, sep=".")

metrics = lapply(metrics, msr)

rrd4e$aggregate(metrics)

roc\_d4e <- autoplot(rrd4e, type = "roc") +

theme\_minimal() + # 使用简洁主题作为基础

theme(

panel.background = element\_rect(fill = "white", colour = "white"),

plot.background = element\_rect(fill = "white", colour = "white")

)

roc\_d4e <- roc\_d4e + ggtitle("Elastic Net ROC Curve with Clinic and Mutation Data")

auc\_d4e <- autoplot(rrd4e, measure=msr("classif.auc"))

ggsave(filename =

"visulization/result of elastic net with clinic data and mutation data.png",

plot = roc\_d4e,

width = 10, height = 8,

units = "in", dpi = 300)