install.packages(c("ggpubr","corrplot",

"glmnet","caret","CBCgrps",

"tidyverse","rms"))

install.packages("data.table","xgboost","Matrix","caTools")

library(corrplot)

library(glmnet)

library(caret)

library(CBCgrps)

library(nortest)

library(tidyverse)

library(ggpubr)

library(rms)

library(pROC)

library(viridis)

library(caret)

set.seed(3464)

trainIndex <- createDataPartition(data$diagnosis, p = .7, list = FALSE)

train <- data[trainIndex, ]

test <- data[-trainIndex, ]

write.csv(train, file = "train.csv", row.names = FALSE)

write.csv(test, file = "test.csv", row.names = FALSE)

min\_max\_scale = function(x){

(x-min(x))/(max(x)-min(x))

}

library(dplyr)

library(glmnet)

data2 = train%>%

mutate\_if(.predicate = is.numeric,

.funs = min\_max\_scale)%>%

as.data.frame()

set.seed(123) #random number generator

x <- data.matrix(data2[, -1])

y <- data2[, 1]

y<-as.numeric(unlist(y))

library(tidyr)

library(glmnet)

lasso <- glmnet(x, y, family = "binomial", nlambda = 1000, alpha = 1)

print(lasso)

plot(lasso, xvar = "lambda", label = FALSE)

lasso.cv = cv.glmnet(x, y,alpha = 1,nfolds =20,family="binomial")

plot(lasso.cv)

lasso.cv$lambda.min #minimum

lasso.cv$lambda.1se #one standard error away

coef(lasso.cv, s = "lambda.1se")

lassov <- c("diagnosis","apgar\_1min","ps\_use\_within\_72h","eos")

library(rms)

mydata<-train

attach(mydata)

dd <- datadist(mydata)

options(datadist = 'dd')

fit0 <- lrm(diagnosis ~ apgar\_1min + ps\_use\_within\_72h + eos , data = mydata, x = T, y = T)

fit0

nom0 <- nomogram(fit0,fun = plogis,fun.at = c(.001,.01,.05,.5,.95,.99,.999),

lp = T,funlabel = "diagnosis rate")

plot(nom0)

install.packages("regplot")

library(regplot)

regplot(reg = fit0,

plots = c("density", "boxes"),

center = TRUE,

observation = mydata[1,],

title = "",

points = TRUE,

droplines = TRUE,

clickable = FALSE,

showP = TRUE,

rank = "sd",

failtime = c(1, 2, 3),

prfail = FALSE)

regplot(reg = fit0,

plots = c("density", "boxes"),

center = TRUE,

observation = mydata[1,],

title = "Diagnosis Probability",

points = TRUE,

droplines = TRUE,

clickable = FALSE,

showP = TRUE,

rank = "sd",

boxcol = "#808080",

dencol = "#4F94CD",

obscol = "red", #

spkcol = "#696969",

cexscales = 0.6,

cexvars = 0.7,

cexcats = 0.7,

leftlabel = FALSE)

gd<-predict(fit0,newdata = train,

se.fit = FALSE,dispersion = NULL,trems= NULL,

na.action = na.pass)

#roc

library(pROC)

roc.list<-roc(train$diagnosis,gd)#low

roc.list

g.list<- ggroc(roc.list,alpha = 1,size = 0.8,legacy.axes = TRUE,color="red")

g.list+theme\_classic()+ggtitle("data")+annotate(geom="segment",x = 0,y = 0,xend =1,yend = 1)

library(pROC)

library(ggplot2)

roc.list <- roc(train$diagnosis, gd)

g.list <- ggroc(roc.list, alpha = 1, size = 1, legacy.axes = TRUE, color = "#377EB8")

g.list <- g.list +

theme\_minimal(base\_size = 14) +

ggtitle("ROC Curve of Diagnostic Test") +

theme(plot.title = element\_text(hjust = 0.5, face = "bold", color = "#377EB8"),

axis.title = element\_text(face = "bold", color = "#4D4D4D"),

axis.text = element\_text(color = "#4D4D4D")) +

geom\_segment(aes(x = 0, y = 0, xend = 1, yend = 1), linetype = "dashed", color = "#E41A1C") +

annotate("text", x = 0.2, y = 0.9, label = sprintf("AUC = %.3f", auc(roc.list)), size = 5, color = "#4D4D4D") #

print(g.list)

gd <- predict(fit0, newdata = train, type = "fitted")

roc.list <- roc(train$diagnosis, gd)

best\_coords <- coords(roc.list,

x = "best",

best.method = "youden",

ret = c("threshold", "sensitivity", "specificity", "accuracy", "ppv", "npv"))

print(round(best\_coords, 3))

gd<-predict(fit0,newdata = test,

se.fit = FALSE,dispersion = NULL,trems= NULL,

na.action = na.pass)

#roc

library(pROC)

roc.list<-roc(test$diagnosis,gd)#low

roc.list

g.list<- ggroc(roc.list,alpha = 1,size = 0.8,legacy.axes = TRUE,color="red")

g.list+theme\_classic()+ggtitle("data")+annotate(geom="segment",x = 0,y = 0,xend =1,yend = 1)

library(pROC)

library(ggplot2)

roc.list <- roc(test$diagnosis, gd)

g.list <- ggroc(roc.list, alpha = 1, size = 1, legacy.axes = TRUE, color = "#377EB8")

g.list <- g.list +

theme\_minimal(base\_size = 14) +

ggtitle("ROC Curve of Diagnostic Test") +

theme(plot.title = element\_text(hjust = 0.5, face = "bold", color = "#377EB8"),

axis.title = element\_text(face = "bold", color = "#4D4D4D"),

axis.text = element\_text(color = "#4D4D4D")) +

geom\_segment(aes(x = 0, y = 0, xend = 1, yend = 1), linetype = "dashed", color = "#E41A1C") +

annotate("text", x = 0.2, y = 0.9, label = sprintf("AUC = %.3f", auc(roc.list)), size = 5, color = "#4D4D4D")

print(g.list)

library(pROC)

gd <- predict(fit0, newdata = test, type = "fitted")

roc.list <- roc(test$diagnosis, gd)

best\_coords <- coords(roc.list,

x = "best",

best.method = "youden",

ret = c("threshold", "sensitivity", "specificity", "accuracy", "ppv", "npv"))

print(round(best\_coords, 3))

train <- train[complete.cases(train[, c("diagnosis", "apgar\_1min", "ps\_use\_within\_72h", "eos")]), ]

fit0 <- lrm(diagnosis ~ apgar\_1min + ps\_use\_within\_72h + eos, data = train, x = TRUE, y = TRUE)

cal <- calibrate(fit0, cmethod = "hare", method = "boot", B = 1000)

par(mar = c(5.1, 4.1, 4.1, 2.1))

plot(cal, xlim = c(0, 1), ylim = c(0, 1),

xlab = "Nomogram Predicted Survival", ylab = "Actual Survival")

y\_train\_num <- as.numeric(as.character(train$diagnosis))

p\_hat\_train <- predict(fit0, type = "fitted")

brier\_manual <- mean((p\_hat\_train - y\_train\_num)^2)

df\_cal <- data.frame(

logit\_p = qlogis(p\_hat\_train),

y = y\_train\_num

)

fit\_cal\_intercept <- glm(

y ~ offset(logit\_p),

family = binomial(link = "logit"),

data = df\_cal

)

intercept\_manual <- coef(fit\_cal\_intercept)[1]

fit\_cal\_slope <- glm(

y ~ logit\_p,

family = binomial(link = "logit"),

data = df\_cal

)

slope\_manual <- coef(fit\_cal\_slope)[2]

if (!requireNamespace("ResourceSelection", quietly = TRUE)) {

install.packages("ResourceSelection")

}

library(ResourceSelection)

hl\_res <- hoslem.test(y\_train\_num, p\_hat\_train, g = 10)

E\_MANUAL <- sum(p\_hat\_train)

O\_MANUAL <- sum(y\_train\_num)

eo\_manual <- E\_MANUAL / O\_MANUAL

cat("\n--- Manual Calibration Metrics (Training) ---\n")

cat("Intercept (manual) :", round(intercept\_manual, 3), "\n")

cat("Slope (manual) :", round(slope\_manual, 3), "\n")

cat("Brier Score :", round(brier\_manual, 3), "\n")

cat("HL Chi-square :", round(hl\_res$statistic, 2), "\n")

cat("HL P-value :", signif(hl\_res$p.value, 2), "\n")

cat("E/O Ratio :", round(eo\_manual, 2), "\n")

test <- test[complete.cases(test[, c("diagnosis", "apgar\_1min", "ps\_use\_within\_72h", "eos")]), ]

fit0 <- lrm(diagnosis ~ apgar\_1min + ps\_use\_within\_72h + eos, data = test, x = TRUE, y = TRUE)

cal <- calibrate(fit0, cmethod = "hare", method = "boot", B = 1000)

par(mar = c(5.1, 4.1, 4.1, 2.1))

plot(cal, xlim = c(0, 1), ylim = c(0, 1),

xlab = "Nomogram Predicted Survival", ylab = "Actual Survival")

y\_test\_num <- as.numeric(as.character(test$diagnosis))

p\_hat\_test <- predict(fit0, type = "fitted")

brier\_manual <- mean((p\_hat\_test - y\_test\_num)^2)

df\_cal <- data.frame(

logit\_p = qlogis(p\_hat\_test),

y = y\_test\_num

)

fit\_cal\_intercept <- glm(

y ~ offset(logit\_p),

family = binomial(link = "logit"),

data = df\_cal

)

intercept\_manual <- coef(fit\_cal\_intercept)[1]

fit\_cal\_slope <- glm(

y ~ logit\_p,

family = binomial(link = "logit"),

data = df\_cal

)

slope\_manual <- coef(fit\_cal\_slope)[2]

if (!requireNamespace("ResourceSelection", quietly = TRUE)) {

}

library(ResourceSelection)

hl\_res <- hoslem.test(y\_test\_num, p\_hat\_test, g = 10)

E\_MANUAL <- sum(p\_hat\_test)

O\_MANUAL <- sum(y\_test\_num)

eo\_manual <- E\_MANUAL / O\_MANUAL

cat("\n--- Manual Calibration Metrics (testing) ---\n")

cat("Intercept (manual) :", round(intercept\_manual, 3), "\n")

cat("Slope (manual) :", round(slope\_manual, 3), "\n")

cat("Brier Score :", round(brier\_manual, 3), "\n")

cat("HL Chi-square :", round(hl\_res$statistic, 2), "\n")

cat("HL P-value :", signif(hl\_res$p.value, 2), "\n")

cat("E/O Ratio :", round(eo\_manual, 2), "\n")

library(rmda)

library(ggplot2)

library(rmda)

train$diagnosis <- ifelse(train$diagnosis == 2, 1, 0)

train <- train[complete.cases(train[, c("diagnosis", "apgar\_1min", "ps\_use\_within\_72h", "eos")]), ]

dca\_model <- decision\_curve(diagnosis ~ apgar\_1min + ps\_use\_within\_72h + eos,

data = train,

study.design = "cohort",

bootstraps = 100)

plot\_decision\_curve(dca\_model,

curve.names = "Logistic Regression Model",

xlim = c(0, 1),

cost.benefit.axis = TRUE,

confidence.intervals = TRUE) +

theme(

legend.title = element\_blank(),

legend.text = element\_text(size = 8),

legend.key.size = unit(0.6, "lines"),

legend.position = c(1, 1),

legend.justification = c(1, 1)

)

test$diagnosis <- ifelse(test$diagnosis == 2, 1, 0)

test <- test[complete.cases(test[, c("diagnosis", "apgar\_1min", "ps\_use\_within\_72h", "eos")]), ]

dca\_model <- decision\_curve(diagnosis ~ apgar\_1min + ps\_use\_within\_72h + eos,

data = test,

study.design = "cohort",

bootstraps = 100)

plot\_decision\_curve(dca\_model,

curve.names = "Logistic Regression Model",

xlim = c(0, 1),

cost.benefit.axis = TRUE,

confidence.intervals = TRUE) +

theme(

legend.title = element\_blank(),

legend.text = element\_text(size = 8),

legend.key.size = unit(0.6, "lines"),

legend.position = c(1, 1),

legend.justification = c(1, 1)

)

library(interactions)

library(patchwork)

library(ggplot2)

interaction\_model1 <- glm(diagnosis ~ apgar\_1min \* ps\_use\_within\_72h

+ eos,

data = train,

family = binomial())

summary(interaction\_model1)

p1 <- interact\_plot(interaction\_model1,

pred = apgar\_1min,

modx = ps\_use\_within\_72h,

plot.points= TRUE,

interval = TRUE,

int.width = 0.95,

x.label = "Apgar 1-min Score",

y.label = "Predicted Probability of MV") +

ggtitle("A. Apgar × PS Use") +

theme\_minimal(base\_size = 14) +

theme(plot.title = element\_text(face = "bold"))

interaction\_model2 <- glm(diagnosis ~ ps\_use\_within\_72h \* eos

+ apgar\_1min,

data = train,

family = binomial())

summary(interaction\_model2)

p2 <- interact\_plot(interaction\_model2,

pred = ps\_use\_within\_72h,

modx = eos,

plot.points= TRUE,

interval = TRUE,

int.width = 0.95,

x.label = "PS Use within 72h",

y.label = "Predicted Probability of MV") +

ggtitle("B. PS Use × EOS") +

theme\_minimal(base\_size = 14) +

theme(plot.title = element\_text(face = "bold"))

combo\_fig <- p1 + p2 +

plot\_layout(ncol = 2) +

plot\_annotation(

title = "Figure 5. Interaction Effects on Mechanical Ventilation Risk",

theme = theme(plot.title = element\_text(hjust = 0.5, size = 16, face = "bold"))

)

print(combo\_fig)

ggsave("Figure5\_Interaction\_Combined.pdf",

combo\_fig,

width = 12,

height = 5.5,

dpi = 600)