Disease Prediction on the Crohn's Syndrome Dataset-Rcode

2024-10-19

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tidyr)
library(dplyr)
```

```
# load data
load("E:\\qjy\\ecnu\\   \\  \\    1\\CD.Rdata")
load("E:\\qjy\\ecnu\\   \\  \\    1\\CD_P.Rdata")
CD_P<-data.frame(P)
```

```
#P-values were calculated using the chi-square test
n <- ncol(X_train)
chis_p <- rep(0,n)
for (i in c(1:n)) {
  p <- table(X_train[,i],y_train)
  chis_p[i] <- chisq.test(p)$p.value
}
#write.table(chis_p,"pvalue.csv",row.names=FALSE,col.names=TRUE)
```

```
#Variables were selected using bonferroni,FWER
p_value <-read.csv("E://qjy//ecnu//   //  //    1//pvalue.csv")
sum(p_value<0.05/240000)
X_bon <- X_train[,which(p_value<0.05/240000)]
# write.table(X_bon,"X_bon.csv",row.names=FALSE,col.names=TRUE)
```

```
#Variables were selected using adjust bonferroni,FWER
p1<-p.adjust(unlist(p_value), method = "bonferroni")
sum(p1<0.05)
X_bon1 <- X_train[,which(p1<0.05)]
# write.table(X_bon1,"X_bon1.csv",row.names=FALSE,col.names=TRUE)
```

```r
#Variables were selected using Holm,FWER
p2<-p.adjust(unlist(p_value), method = "holm")
sum(p2<0.05)
X_holm <- X_train[,which(p2<0.05)]
# write.table(X_holm,"X_holm.csv",row.names=FALSE,col.names=TRUE)
```

```r
#Variables were selected using BH,FDR
p3<-p.adjust(unlist(p_value), method = "BH")
sum(p3<0.05)
X_BH <- X_train[,which(p3<0.05)]
```

```r
#Variables were selected using above p-value
p_belge<-data.frame(P[,1])
p_cedar2<-data.frame(P[,2])
p_adolescent<-data.frame(P[,3])
p_cedar1<-data.frame(P[,4])
p_niddkj<-data.frame(P[,5])
p_german<-data.frame(P[,6])
p_niddknj<-data.frame(P[,7])
p_belge_adj<-p.adjust(unlist(p_belge), method = "BH")
p_cedar2_adj<-p.adjust(unlist(p_cedar2), method = "BH")
p_adolescent_adj<-p.adjust(unlist(p_adolescent), method = "BH")
p_cedar1_adj<-p.adjust(unlist(p_cedar1), method = "BH")
p_niddkj_adj<-p.adjust(unlist(p_niddkj), method = "BH")
p_german_adj<-p.adjust(unlist(p_german), method = "BH")
p_niddknj_adj<-p.adjust(unlist(p_niddknj), method = "BH")
p_7 <- CD_P[which((p_belge_adj<0.1)|(p_cedar2_adj<0.1)|(p_adolescent_adj<0.1)|(p_cedar1_adj<0.1)|(p_nid
X_train_7 <- X_train[,which((p_belge_adj<0.1)|(p_cedar2_adj<0.1)|(p_adolescent_adj<0.1)|(p_cedar1_adj<0
```

```r
# download IGESS from github
# install.packages("devtools")
# install.packages("rJava")
library(rJava)
library(devtools)
library(usethis)
devtools::install_local("C://Users//Lenovo//Desktop//IGESS-master.zip")  #  all-- 1
```

```r
#IGESS
library(IGESS)
colname<-paste("rs",c(1:317),sep="")
colnames(X_train_7)<-colname
row.names(p_7)<-colname
X_train_7_scale<-scale(X_train_7,center = TRUE,scale = FALSE)
str(X_train_7_scale)
str(p_7)
```

```r
#fit
fit <- IGESS(X_train_7_scale, y_train, SS = p_7)
# auc
auc_scores<-c()
for (i in 1:50){
  set.seed(i)
```

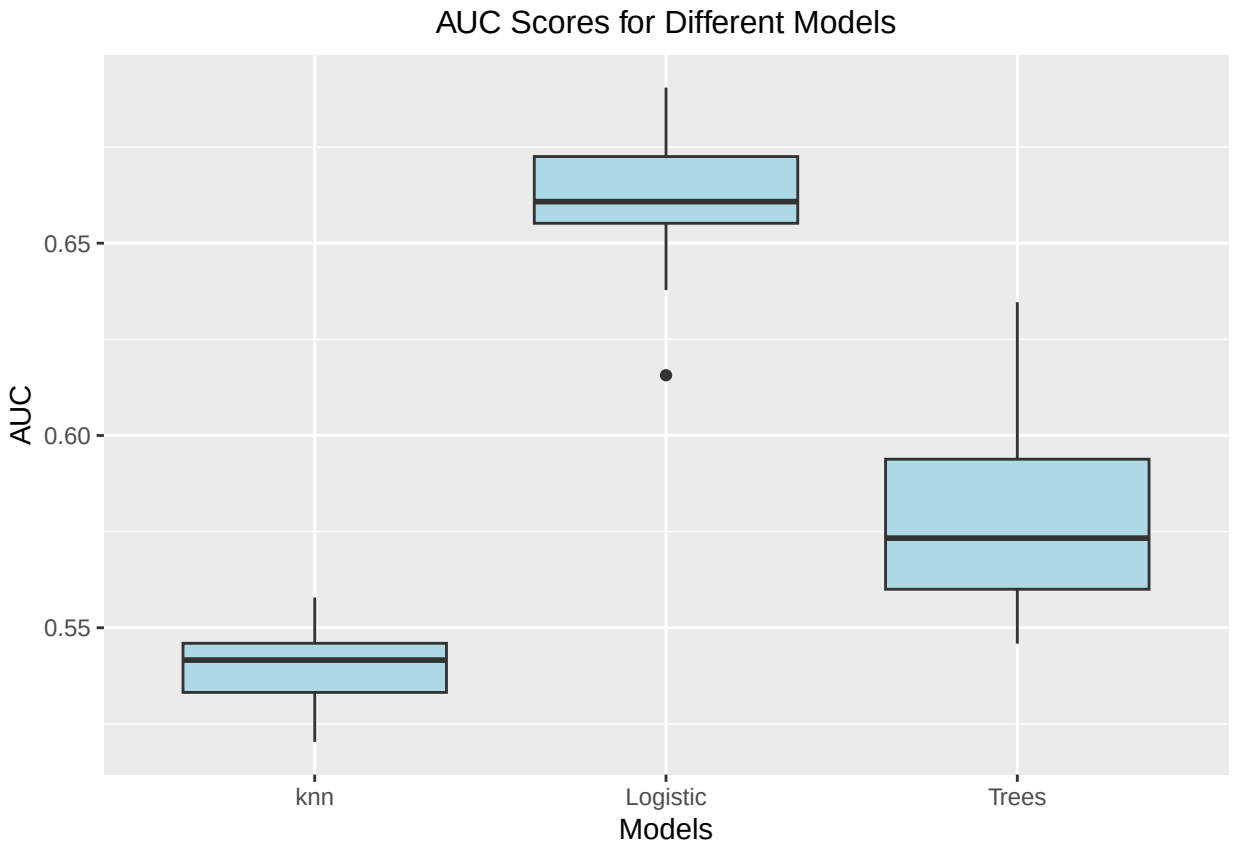
```
  auc_scores[i] <- IGESSCV(X_train_7_scale, y_train, SS = p_7,measure = "auc")
}
# write.table(auc_scores,"auc_scores_IGESS.csv",row.names=FALSE,sep=",")
```

```
#predict
X_test_7 <- X_test[,which((p_belge_adj<0.1)|(p_cedar2_adj<0.1)|(p_adolescent_adj<0.1)|(p_cedar1_adj<0.1
X_test_7_scale<-scale(X_test_7,center = TRUE,scale = FALSE)
yhat <- IGESS_Predict(fit,X_test_7_scale )
ypred <- round(yhat2)
write.table(yhat,"y_pred_IGESS.csv",row.names=FALSE,col.names=TRUE)
```

```
#plot
auc_scores<-read.csv("E:\\qjy\\ecnu\\    \\  \\     1\\auc_scores.csv")
colnames(auc_scores)<-c("RF","Logistic","GBDT","SVM","LightGBM","knn","Trees","XGboost")
```

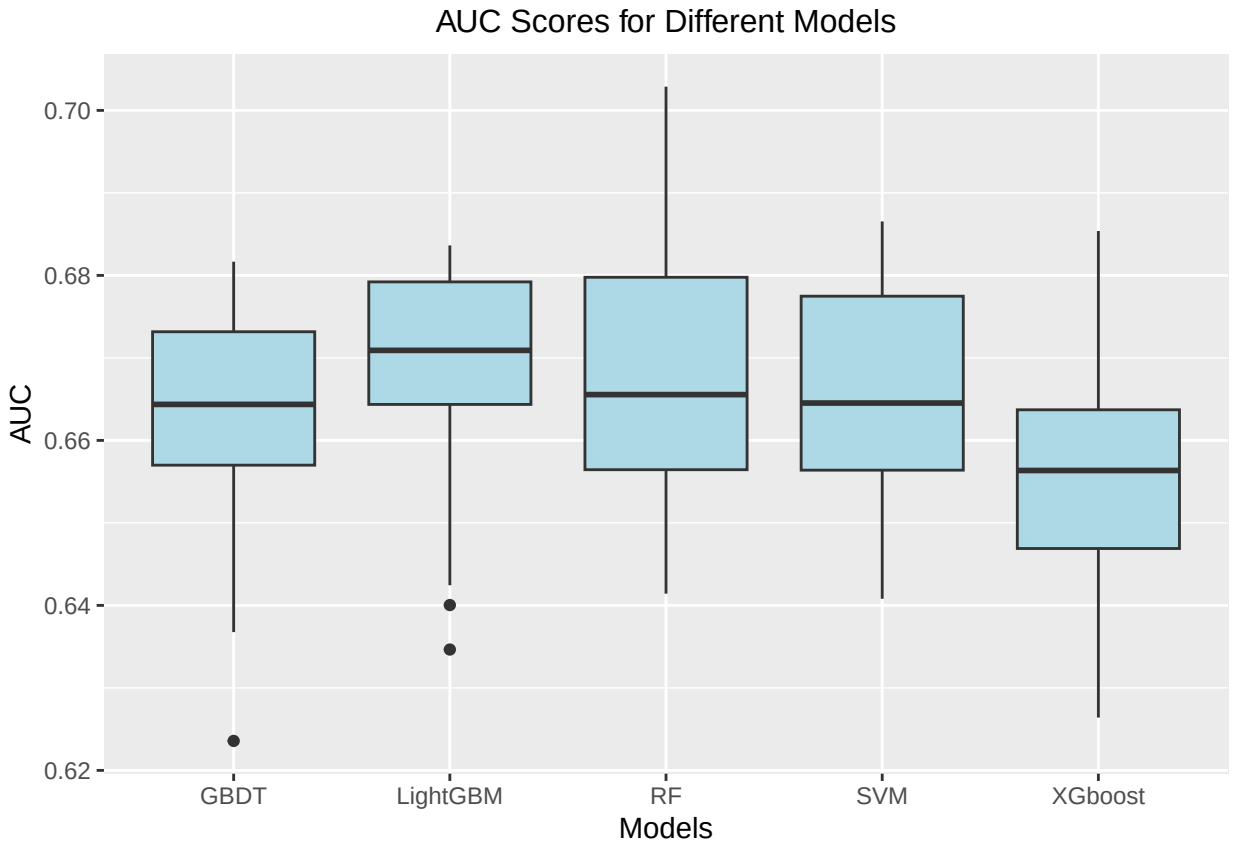
```
# Convert data to long format
auc_scores_long <- tidyr::gather(auc_scores[,c(2,6,7)], key = "Model", value = "AUC")
```

```
library(ggplot2)
# Plot the base model box plot
ggplot(auc_scores_long, aes(x = Model, y = AUC)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "AUC Scores for Different Models", hjust=0.5)+
  labs(x = "Models", y = "AUC") +
  theme(plot.title = element_text(size=12,hjust=0.5))
```



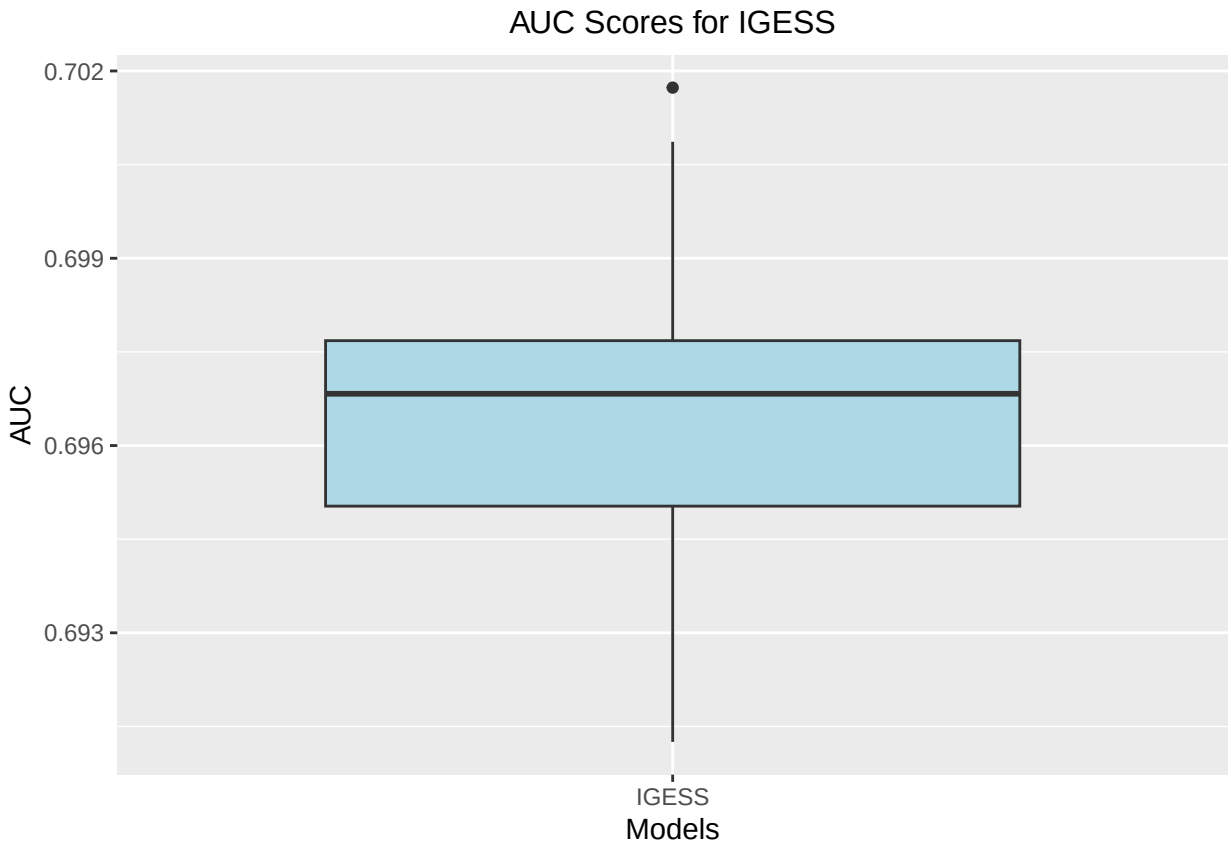
```r
# Convert data to long format
auc_scores_long1 <- tidyr::gather(auc_scores[,c(-2,-6,-7)], key = "Model", value = "AUC")

# Plot the embedding model box plot
ggplot(auc_scores_long1, aes(x = Model, y = AUC)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "AUC Scores for Different Models", hjust=0.5)+
  labs(x = "Models", y = "AUC") +
  theme(plot.title = element_text(size=12,hjust=0.5))
```



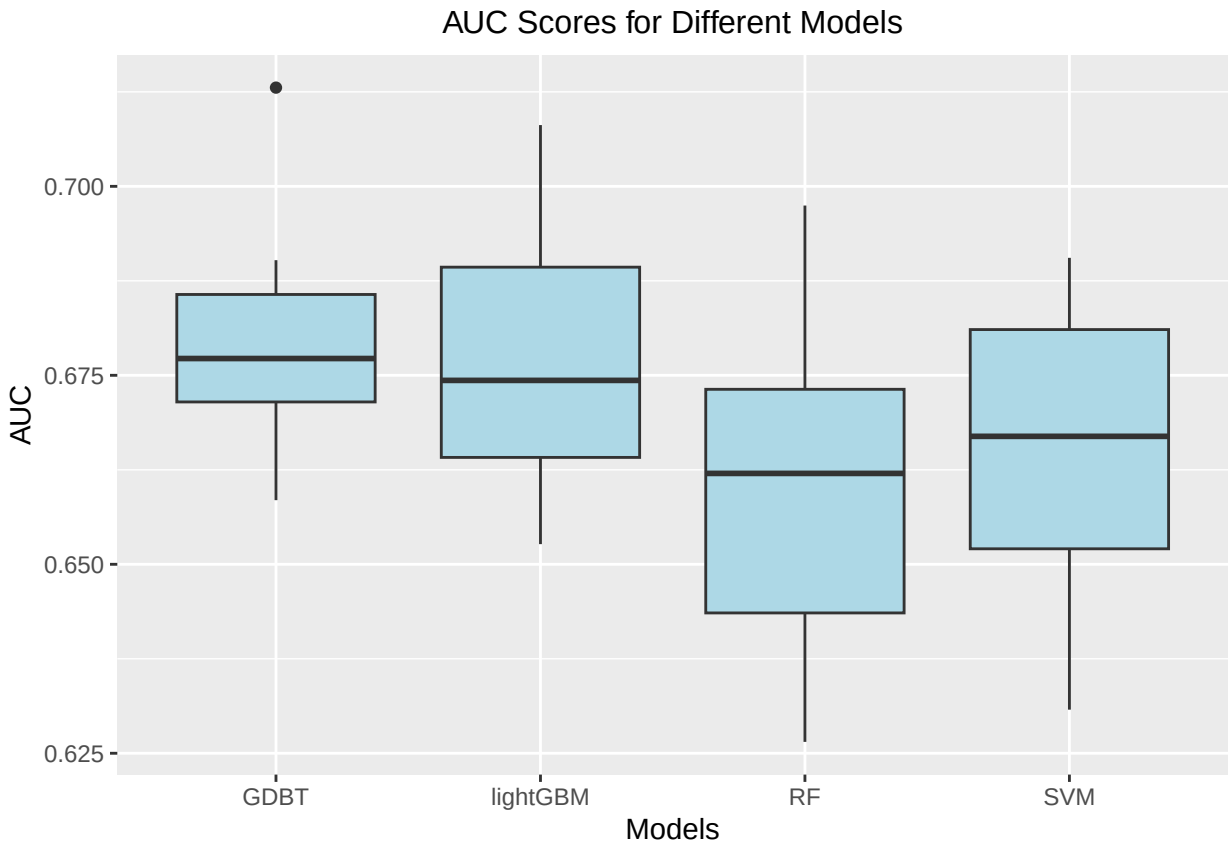

```r
auc_scores_IGESS<-read.csv("E:\\qjy\\ecnu\\   \\  \\     1\\auc_scores_IGESS.csv",header = T)
auc_scores_IGESS <- tidyr::gather(auc_scores_IGESS, key = "Model", value = "AUC")

# Plot the IGESS box plot
ggplot(auc_scores_IGESS, aes(x = Model, y = AUC)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "AUC Scores for IGESS", hjust=0.5)+
  labs(x = "Models", y = "AUC") +
  theme(plot.title = element_text(size=12,hjust=0.5))
```




```
auc_scores_bys<-read.csv("E:\\qjy\\ecnu\\   \\  \\    1\\auc_scores_BH_bys.csv",header = T)
auc_scores_bys <- tidyr::gather(auc_scores_bys, key = "Model", value = "AUC")

# Plot the embedding model box plot with Bayesian optimization
ggplot(auc_scores_bys, aes(x = Model, y = AUC)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "AUC Scores for Different Models", hjust=0.5)+
  labs(x = "Models", y = "AUC") +
  theme(plot.title = element_text(size=12,hjust=0.5))
```




```
y_pred<-read.csv("E:\\qjy\\ecnu\\   \\  \\    1\\y_pred.csv")
# save(y_pred,file="y_pred")
```