# RDataMining

# Week 5 Lab in R

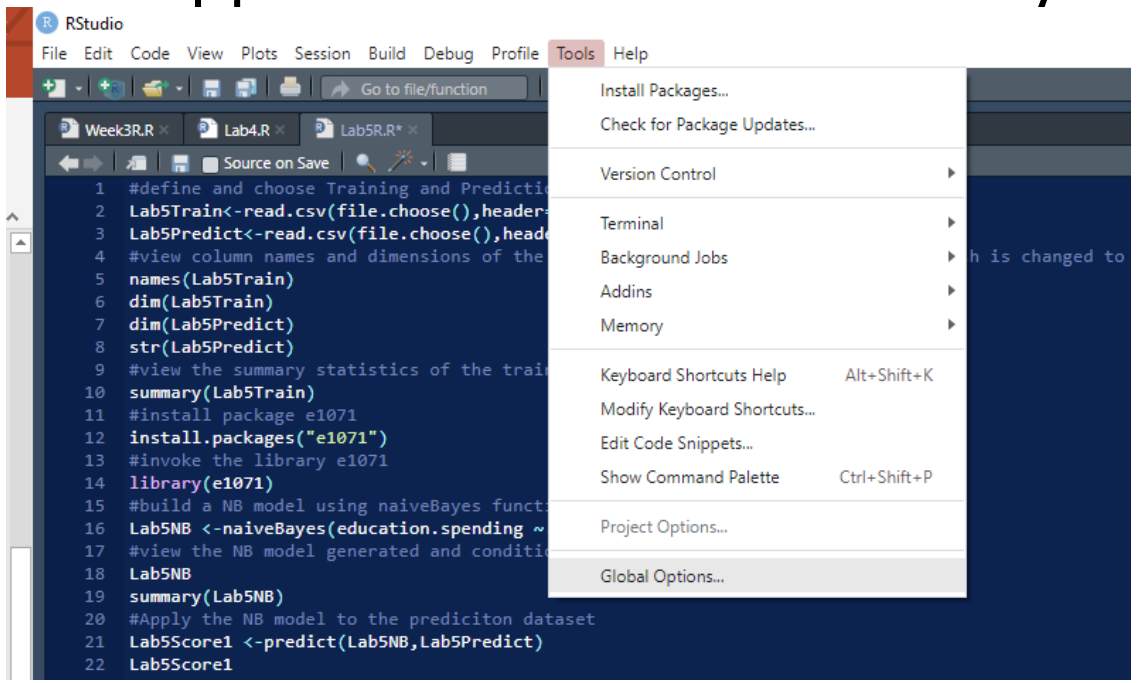
Dr. Liang (Leon) Chen

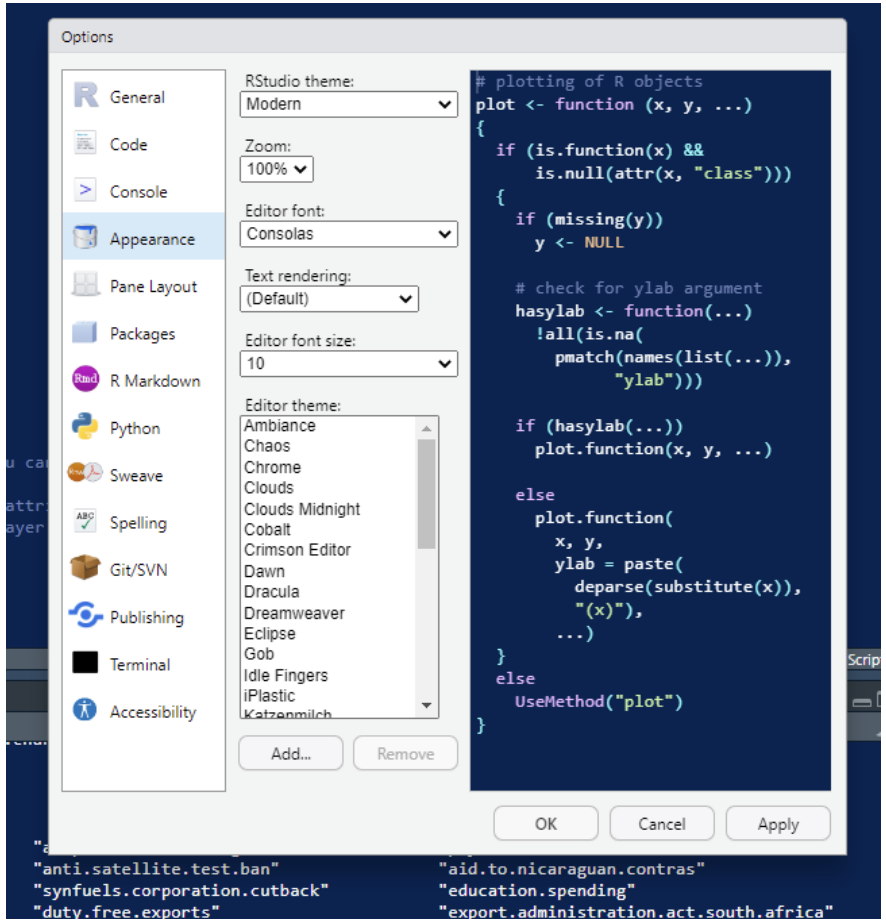
# Instructions

- R scripts for Naïve Bayes classifier and Logistic Regression classifier are provided in the next few slides, please follow them and complete the lab in R.
- You are recommended to type notes (starting at #) as it's a good manner to have them in you code.
- In order to see codes and notes clearly, I show the script in RStudio.
- There are five questions in this lab to answer. Please type your answer via HW2 Submission.

# Change appearance in RStudio

If you prefer a difference appearance, you can change it: Tools → Global Options → Appearance. You can choose your favorite text size, editor theme, etc.



# Naïve Bayes Classifier

Source on Save

```
#define and choose Training and Prediction datasets
Lab5Train<-read.csv(file.choose(),header=T,stringsAsFactors = T)
Lab5Predict<-read.csv(file.choose(),header=T,stringsAsFactors = T)
#view column names and dimensions of the training dataset (please notice that dash is changed to dot in R)
names(Lab5Train)
dim(Lab5Train)
#view the summary statistics of the training dataset
summary(Lab5Train)
#install package e1071
install.packages("e1071")
#invoke the library e1071
library(e1071)
#build a NB model using naiveBayes function and including all attributes
Lab5NB <-naiveBayes(education.spending ~., data=Lab5Train)
#view the NB model generated and conditional proabilities
Lab5NB
#Apply the NB model to the prediciton dataset
Lab5Score <-predict(Lab5NB,Lab5Predict)
#view the prediction result
Lab5Score
#show the summary statistics of Lab5Score
summary(Lab5Score)
#the following three rows of script help us write the prediction value of education.spending to the prediction dataset
Score<-data.frame(Lab5Score)
Lab5Predict$education.spending<-Score$Lab5Score
Lab5Predict
```

Here, we ask R to convert all the strings to factors by adding a new parameter, stringAsFactors = True; otherwise, they will be characters and then you have to convert them to factors later.

You can also install the package "rminer"(see details here) as an alternative

For details about naïve classifier, please click here

Q1: Please indicate how many of them are predicted to vote as y.

Q2: We practice Naïve Bayes classifier in both R and RapidMiner, do they generate the same prediction for each record in the prediction dataset? (Hint: you can list predictions results generated by the two methods in Excel and then check if the prediction results are different in each row; export the prediction result using write.csv() function.

# Additional Script (Not Required)

```r
#invoke the library e1071
library(e1071)
#build a NB model using naiveBayes function and including all attributes
Lab5NB <-naiveBayes(education.spending ~., data=Lab5Train)
#view the NB model generated and conditional proabilities
Lab5NB
#Apply the NB model to the prediciton dataset
Lab5Score <-predict(Lab5NB,Lab5Predict)
#view the prediction result
Lab5Score
#show the summary statistics of Lab5Score
summary(Lab5Score)
#the following three rows of script help us write the prediction value of education.spending to the prediction dataset
Score<-data.frame(Lab5Score)
Lab5Predict$education.spending<-Score$Lab5Score
Lab5Predict

#Combine predictions with prediciton dataset into one single data frame
Lab5Prediction<-data.frame(Lab5Score,Lab5Predict)
Lab5Prediction
#replace prediction.education by Lab5Score
Lab5Prediction$education.spending <-ifelse(is.na(Lab5Prediction$education.spending),Lab5Prediction$Lab5Score,Lab5Prediction$education.spending)
#delete the column Lab5Score
Lab5Prediction$Lab5Score<-NULL
#view Lab5Prediction; please notice that the values in education.spending are 1,2,and 3, rather than abstain, n, and y
Lab5Prediction

#the following three rows help us delete the orginal education.spending column but use the first column as a new education.spending column
Lab5Prediction2<-data.frame(Lab5Score,Lab5Predict)
Lab5Prediction2$education.spending=NULL
names(Lab5Prediction2)[1]=paste("education.spending")
```

All these show you different ways to combine the prediction result and the prediction dataset

# Show Conditional a-posterior probabilities in R (not required)

```
names(Lab5Train)
dim(Lab5Train)
#view the summary statistics of the training dataset
summary(Lab5Train)
#install package e1071
install.packages("e1071")
#invoke the library e1071
library(e1071)
#build a NB model using naiveBayes function and including all attributes
Lab5NB <-naiveBayes(education.spending ~., data=Lab5Train)
#view the NB model generated and conditional proabilities
Lab5NB
#Apply the NB model to the prediciton dataset
Lab5Score <-predict(Lab5NB,Lab5Predict)

#the following two rows of scripts are used to gerneate the conditional a-posterior probabilities for each class, and the class with maximal probability else.
Lab5Score2 <-predict(Lab5NB,Lab5Predict,type = "raw")
Lab5Score2

#view the prediction result
Lab5Score
#show the summary statistics of Lab5Score
summary(Lab5Score)
#the following three rows of script help us write the prediction value of education.spending to the prediction dataset
Score<-data.frame(Lab5Score)
Lab5Predict$education.spending<-Score$Lab5Score
Lab5Predict
```

When you add type="raw" as an additional argument, you can see the conditional a-posterior probabilities for each class, and the class with maximal probability else.

# Logistic Regression Classifier

```
# change the value "abstain" to "n" to make the variable binominal
Lab5Train$education.spending[Lab5Train$education.spending=="abstain"]<-"n"
# develop a logistic regression model using glm function.
LogModel <- glm(education.spending ~ .,family = "binomial", data=Lab5Train)
#view the logistic regression model
summary(LogModel)
#use the LR model to make prediction
Lab5ScoreLR <-predict(LogModel,Lab5Predict,type="response")
#view the prediciton; Round all those predicted probablities to the third decimal place.
round(Lab5ScoreLR, 3)
#check if each predicted probablity is greater than 0.5 (i.e., with y as the predicted class)
Lab5ScoreLR>0.5
# count how many of them are predicted as y (i.e., probability greater than 0.5)
sum(Lab5ScoreLR>0.5)
```

For more details about the glm function, please check this link.

The reason why we chose type="response" is that the default predictions of the binomial Logistic Regression are log-odds, and type = "response" gives the predicted probabilities. For details, please check this link.

Q3: Please indicate how many of them are predicted to vote as y.

Q4: After you run round(Lab5ScoreLR, 3) in R, please compare each record's probability of voting y in R with that in RM (i.e., Confidence(y) in RM prediction result). Do the two tools generate the same probability of voting y?

Q5: Do R and RM generate the same prediction (y or n) for each record in the prediction dataset?

# FAQs

- After you import your training and prediction dataset, please check their structure. Because we apply the stringAsFactors = True, all the columns except the target attribute in the prediction dataset should be factors. If you had characters, you may have to convert them to factors; otherwise, it might make your prediction results differently.

```
> str(Lab5Predict)
'data.frame':   35 obs. of  17 variables:
 $ handicapped.infants                  : Factor w/ 3 levels "abstain","n",..: 2 2 1 2 3 2 3 2 3 2 ...
 $ water.project.cost.sharing           : Factor w/ 3 levels "abstain","n",..: 3 3 2 3 3 2 2 2 2 2 ...
 $ adoption.of.the.budget.resolution    : Factor w/ 3 levels "abstain","n",..: 2 2 3 2 2 2 3 2 3 2 ...
 $ physician.fee.freeze                 : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 1 2 1 2 ...
 $ el.salvador.aid                      : Factor w/ 3 levels "abstain","n",..: 3 3 2 3 3 3 3 3 2 3 ...
 $ religious.groups.in.schools          : Factor w/ 3 levels "abstain","n",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ anti.satellite.test.ban              : Factor w/ 3 levels "abstain","n",..: 2 2 3 2 2 2 2 2 3 2 ...
 $ aid.to.nicaraguan.contras            : Factor w/ 3 levels "abstain","n",..: 1 2 3 2 2 2 2 2 3 2 ...
 $ mx.missile                           : Factor w/ 3 levels "abstain","n",..: 2 2 3 2 2 2 3 2 3 2 ...
 $ immigration                          : Factor w/ 3 levels "abstain","n",..: 2 3 3 3 3 2 3 2 2 2 ...
 $ synfuels.corporation.cutback         : Factor w/ 3 levels "abstain","n",..: 1 3 2 2 2 2 2 3 2 2 ...
 $ education.spending                   : logi  NA NA NA NA NA NA ...
 $ superfund.right.to.sue               : Factor w/ 3 levels "abstain","n",..: 1 3 2 1 3 3 3 3 1 3 ...
 $ crime                                : Factor w/ 3 levels "abstain","n",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ duty.free.exports                    : Factor w/ 3 levels "abstain","n",..: 2 2 2 2 2 2 2 2 3 2 ...
 $ export.administration.act.south.africa: Factor w/ 3 levels "abstain","n",..: 1 2 3 2 3 3 3 2 3 2 ...
 $ Party                                : Factor w/ 2 levels "democrat","republican": 2 2 2 2 2 2 1 1 1 2 ...
```

```
> str(Lab5Train)
'data.frame':   400 obs. of  17 variables:
 $ handicapped.infants                  : Factor w/ 3 levels "abstain","n",..: 2 2 1 2 3 2 2 2 2 3 ...
 $ water.project.cost.sharing           : Factor w/ 3 levels "abstain","n",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ adoption.of.the.budget.resolution    : Factor w/ 3 levels "abstain","n",..: 2 2 3 3 3 3 2 2 2 3 ...
 $ physician.fee.freeze                 : Factor w/ 3 levels "abstain","n",..: 3 3 1 2 2 2 3 3 3 2 ...
 $ el.salvador.aid                      : Factor w/ 3 levels "abstain","n",..: 3 3 3 1 3 3 3 3 3 2 ...
 $ religious.groups.in.schools          : Factor w/ 3 levels "abstain","n",..: 3 3 3 3 3 3 3 3 3 2 ...
 $ anti.satellite.test.ban              : Factor w/ 3 levels "abstain","n",..: 2 2 2 2 2 2 2 2 2 3 ...
 $ aid.to.nicaraguan.contras            : Factor w/ 3 levels "abstain","n",..: 2 2 2 2 2 2 2 2 2 3 ...
 $ mx.missile                           : Factor w/ 3 levels "abstain","n",..: 2 2 2 2 2 2 2 2 2 3 ...
 $ immigration                          : Factor w/ 3 levels "abstain","n",..: 3 2 2 2 2 2 2 2 2 2 ...
 $ synfuels.corporation.cutback         : Factor w/ 3 levels "abstain","n",..: 1 2 3 3 3 2 2 2 2 2 ...
 $ education.spending                   : Factor w/ 3 levels "abstain","n",..: 3 3 2 2 1 2 2 2 3 2 ...
 $ superfund.right.to.sue               : Factor w/ 3 levels "abstain","n",..: 3 3 3 3 3 3 1 3 3 2 ...
 $ crime                                : Factor w/ 3 levels "abstain","n",..: 3 3 3 2 3 3 3 3 3 2 ...
 $ duty.free.exports                    : Factor w/ 3 levels "abstain","n",..: 2 2 2 2 3 3 3 1 2 1 ...
 $ export.administration.act.south.africa: Factor w/ 3 levels "abstain","n",..: 3 1 2 3 3 3 3 3 3 1 ...
 $ Party                                : Factor w/ 2 levels "democrat","republican": 2 2 1 1 1 1 1 2 2 1
```

- You can use as.factor() to convert to a factor column, but in our case, we have multiple variables to convert. It will be much easier to use the following codes.
- # convert character columns in the training set and prediction set to factor columns.

```
Lab5Train[sapply(Lab5Train, is.character)] <- lapply(Lab5Train[sapply(Lab5Train, is.character)], as.factor)
Lab5Predict[sapply(Lab5Predict, is.character)] <- lapply(Lab5Predict[sapply(Lab5Predict, is.character)], as.factor)
```

- Then, you can try str() again to see if both of them are converted.
- Further Readings:
  - https://stackoverflow.com/questions/20637360/convert-all-data-frame-character-columns-to-factors
  - https://statisticsglobe.com/convert-character-to-factor-in-r
  - https://stackoverflow.com/questions/30248583/error-could-not-find-function