Midterm-2 Project Portion - Instruction

First and last name: ________ ________ // Pair's first and last name: ______ ________

Submission Date: ____________

Read and Delete This Part Before Submission

- Find a pair, work together, split parts among each of you, explain your findings to each other, make sure you understand all, combine work, and submit separately. It is fine if your codes and results are the same. I expect comments will be your own. If you don't have pair, it is ok.
- Give a name to this rmd file: `Midterm2_Submission_FirstName_LastName.rmd`.
- You will then submit two files to Blackboard: `.rmd` and the knitted `.pdf` files.
- Grading will be based on the pdf file uploaded. Make easy and readable. The grader and the instructor may take a look at the rmd file. The instructor may use your submission fo teaching purposes.
- Always include your comments on results: don't just leave the numbers without explanations. Use full sentences, structured paragraphs if needed, correct grammar, and proofreading.
- Show your knowledge with detailed work in consistency with course materials.
- Show code, however, don't include irrelevant or uncommented outputs. Make the codes and outputs compact.

Midterm-2 Project Instruction

In `Midterm-1 Project`, you have built predictive models using train and test data sets about college students' academic performances and retention status. You fitted four regression models on **Term.GPA** and four classification models on **Persistence.NextYear**. the lowest test score of MSE_{test} achieved on the regression problem was .991 using a simple linear regression, and the highest `accuracy` and `F1` scores obtained were 91.15% and 95.65%, respectively, with the fit of a multiple logistic regression model (equivalently, LDA and QDA give similar performances). Let's call these scores as baseline test scores.

In `Midterm-2 Project`, you will use tree-based methods (trees, random forests, boosting) and artificial neural networks (Modules 5, 6, and 7) to improve the baseline results. There is no any answer key for this midterm: your efforts and justifications will be graded, pick one favorite optimal tree-based method and one optimal ANN architecture for each regression and classification problem (a total of two models for classification and two models for regression), and fit and play with hyperparameters until you get satisfactory improvements in the test data set.

Keep in mind that $Persistence.NextYear$ is not included in as predictor the regression models so use all the predictors except that on the regression. For the classification models, use all the predictors including the term gpa.

First of all, combine the train and test data sets, create dummies for all categorical variables, which include `Entry_Term`, `Gender`, and `Race_Ethc_Visa`, so the data sets are ready to be separated again as train and test. (Expect help on this portion!) You will be then ready to fit models.

A. Improving Regression Models - 15 pts

- Explore tree-based methods, choose the one that is your favorite and yielding optimal results, and then search for one optimal ANN architecture for the regression problem (so two models to report). Fit and make sophisticated decisions by justifying and writing precisely. Report `the test MSE` results in a comparative table along with the methods so the grader can capture all your efforts on building various models in one table.

B. Improving Classification Models - 20 pts

- Explore tree-based methods, choose the one that is your favorite and yielding optimal results, and then search for one optimal ANN architecture for the classification problem (so two models to report). Fit and make sophisticated decisions by justifying and writing precisely. Report `the test accuracy` and `the test F1` results in a comparative table along with the methods so the grader can capture all your efforts in one table.

C. Importance Analyses - 15 pts

- Part a. Perform an importance analysis on the best regression model: which three predictors are most important or effective to explain the response variable? Find the relationship and dependence of these predictors with the response variable. Include graphs and comments.
- Part b. Perform an importance analysis on the best classification model: which three predictors are most important or effective to explain the response variable? Find the relationship and dependence of these predictors with the response variable. Include graphs and comments.

- Part c. Write a conclusion paragraph. Evaluate overall what you have achieved. Did the baselines get improved? Why do you think the best model worked well or the models didn't work well? How did you handle issues? What could be done more to get `better` and `interpretable` results? Explain with technical terms.

Project Evaluation

The submitted project report will be evaluated according to the following criteria:

1. All models in the instruction used correctly
2. Completeness and novelty of the model fitting
3. Techniques and theorems of the methods used accurately
4. Reflection of in-class lectures and discussions
5. Achieved reasonable/high performances; insights obtained (patterns of variables)
6. Clear and minimalist write-ups

If the response is not full or not reflecting the correct answer as expected, you may still earn partial points. For each part or model, I formulated this `partial points` as this:

- 20% of pts: little progress with some minor solutions;
- 40% of pts: major calculation mistake(s), but good work, ignored important pieces;
- 60-80% of pts: correct method used, but minor mistake(s).

Additionally, a student who will get the highest performances from both problems in the class (**`minimum test MSE`** from the regression model and **`highest F1`** from the classification model) will get a BONUS (up to +2 pts). Just follow up when you think you did good job!

Tips

- `Term.gpa` is an aggregated gpa up until the current semester, however, this does not include this current semester. In the modeling of `gpa`, include all predictors except `persistent`.
- The data shows the `N.Ws`, `N.DFs`, `N.As` as the number of courses withdrawn, D or Fs, A's respectively in the current semester.
- Some rows were made synthetic so may not make sense: in this case, feel free to keep or remove.
- It may be poor to find linear association between gpa and other predictors (don't include `persistent` in `gpa` modeling).
- Scatterplot may mislead since it doesn't show the density.
- You will use the test data set to asses the performance of the fitted models based on the train data set.
- Implementing 5-fold cross validation method while fitting with train data set is strongly suggested.
- You can use any packs (`caret`, `Superml`, `rpart`, `xgboost`, or visit to search more) as long as you are sure what it does and clear to the grader.
- Include helpful and compact plots with titles.
- Keep at most 4 decimals to present numbers and the performance scores.
- When issues come up, try to solve and write up how you solve or can't solve.
- Check this part for updates: the instructor puts here clarifications as asked.

Your Solutions

Section A.

Section B. ______________________________

Section C.

- Part a.

  ____

- Part b.

  ____

- Part c.

  ____

- BONUS.

DELETE THIS PORTION WHEN SUBMITTION - Setup and Useful Codes

```
# Create a RStudio Project and work under it.

#Download, Import and Assign
train <- read.csv("StudentDataTrain.csv")
test <- read.csv("StudentDataTest.csv")

#Summarize univariately
summary(train)
summary(test)

#Dims
dim(train) #5961x18
dim(test) #1474x18

#Without NA's
dim(na.omit(train)) #5757x18
dim(na.omit(test)) #1445x18

#Perc of complete cases
sum(complete.cases(train))/nrow(train)
sum(complete.cases(test))/nrow(test)

#Delete or not? In general, we don't delete and use Imputation method to fill na's
#However, in midterm, you can omit or use any imputation method
train <- na.omit(train)
test <- na.omit(test)
dim(train)

#Missing columns as percent
san = function(x) sum(is.na(x))
round(apply(train,2,FUN=san)/nrow(train),4) #pers of na's in columns
round(apply(train,1,FUN=san)/nrow(train),4) #perc of na's in rows

##
#you can create new columns based on features

##Make dummy
#make sure each is numerical and dummy, check what dropped

#train data set with dummies, original cols removed,

View(train<-fastDummies::dummy_cols(train,
                                 select_columns=c('Entry_Term', 'Gender', 'Race_Ethc_Visa'),
                                 remove_first_dummy = TRUE, remove_selected_columns=TRUE)
     )
```

```
#test data set with dummies, original cols removed,
View(test<-fastDummies::dummy_cols(test,
                                    select_columns=c('Entry_Term', 'Gender', 'Race_Ethc_Visa'),
                                    remove_first_dummy = TRUE, remove_selected_columns=TRUE)
)

# select columns or select all except some columns
# adapt
train[,c(which(colnames(train)=="desired_colname"),
         which(colnames(train)!="desired_colname"))]

train[, c(which(colnames(train)!="Persistence.NextYear"))]

#drop Persistence.NextYear for regression problem

##check: if col names are same, dim, etc.

#Variable/Column names
colnames(test)

#Response variables
#you may do this
y1=train$Term.GPA #numerical
y2=train$Persistence.NextYear #categorical: 0, 1
#you may do this
z1=test$Term.GPA #numerical
z2=test$Persistence.NextYear #categorical: 0, 1


##Need scaling?
#do for train and test (combined or separate? choose separate. why?)
train_sc <- train
train_sc[,c(which(colnames(train)!="Persistence.NextYear"))] <- scale(train[, c(which(colnames(train)!=
dim(train_sc)
View(train_sc)

test_sc <- test
test_sc[,c(which(colnames(test)!="Persistence.NextYear"))] <- scale(test[, c(which(colnames(test)!="Pers
dim(test_sc)
View(test_sc)
```

I hereby write and submit my solutions without violating the academic honesty and integrity. If not, I accept the consequences.

Write your pair you worked at the top of the page. If no pair, it is ok. List other fiends you worked with (name, last name): . . .

Disclose the resources or persons if you get any help: . . .

How long did the assignment solutions take?: . . .

References

. . .