Student Dropout Prediction Challenge

**Pooja Singh and Manoj Nair**

1/22/21

## INTRODUCTION

Among many different observable phenomena in the student’s career, University dropout is one of the most complex and adverse events, both for students or institutions. A dropout is a potentially devastating event in life of a student, and also impacts negatively the University from economic point of view.

Drop out prediction is a task that can be addresses by exploiting machine learning techniques, which already proved to be effective in field of education. It will help university to know beforehand which student is going to dropout, and hence can help the students to solve their issues.

## OBJECTIVE:

This project aims at predicting Student’s success at completing coursework. This implies whether a student will drop out from enrolled coursework or not. This will help more students especially low-income, first-generation, and students of color – graduate at higher rates, with high-quality credentials, and at affordable prices.

## RESEARCH QUESTION :

Predicting the likelihood of student dropout or not will require to build predictive model to predict student going to drop out. Frontier set wants to automate student drop out based on student detail provided . These details are divided into 3 parts: Static , progress and financial aid . To automate this process, we need to identify student segments, those are going to drop out so that they can specifically target these students, thus making model outcome as binary (0/ 1). Keeping in mind the output and question, Drop out label will be used as target variable.

## TYPES OF DATA

Data collected is for students pursuing Bachelor’s degree during 2012 to 2017. This data set is divided into three parts: 1. Static Data 2. Progress Data 3. Financial Aid Data

## LOAD NECESSARY LIBRARIES

library(caret)  
library(dplyr)  
# for calculating and for decasting  
library(data.table)  
  
  
# VISUALISATION  
library(ggplot2)  
library(CGPfunctions)  
# for descriptive statistics  
library(psych)  
  
# use of combining data set into one (lpdply)  
library(plyr)  
# same for above reading files  
library(readr)   
  
# FOR MISSING VALUES  
library(imputeTS)   
library(mlbench)  
library(Hmisc)  
# for knn imputation  
library(VIM)  
#plot of missing  
library(visdat)  
  
  
  
# For writing data frame into csv  
library("writexl")  
  
  
# MODELS  
# for random forest model  
library(randomForest)  
library(glm2)  
# Stacking  
library(AppliedPredictiveModeling)  
library(caret)  
library(unbalanced)  
set.seed(31)

# READ STUDENT PROGRESS DATA :

This data reflect students’ academic activity, or progress, for each term enrolled. This file contains summer, winter, fall. Fall session has files from 2011 - 2016 , summer file and winter files are from 2012- 2017.

# 

# Reading all student progress files

**PROCESS**:

**STEP1:**

**Since the progress folder contains various files, I applied loop using function ldply() to read and combine all data into one Data Frame.**

#Read data

**CODE:**

# STEP1: READING STUDENT PROGRESS  
  
setwd("C:\\Users\\pooja\\R\_STUDIO\_FILES\\R\_FILES\_PROJECT\_530\\Student Retention Challenge Data\\Student Progress Data")  
# writing name of folder containig multiple files  
myfolder = "Student Progress Data"  
# Listing all files  
allfiles = list.files()  
allfiles

## [1] "Fall 2011\_SP.csv" "Fall 2012\_SP.csv" "Fall 2013\_SP.csv"   
## [4] "Fall 2014\_SP.csv" "Fall 2015\_SP.csv" "Fall 2016\_SP.csv"   
## [7] "Spring 2012\_SP.csv" "Spring 2013\_SP.csv" "Spring 2014\_SP.csv"  
## [10] "Spring 2015\_SP.csv" "Spring 2016\_SP.csv" "Spring 2017\_SP.csv"  
## [13] "Sum 2012.csv" "Sum 2013.csv" "Sum 2014.csv"   
## [16] "Sum 2015.csv" "Sum 2016.csv" "Sum 2017.csv"

list.files()

## [1] "Fall 2011\_SP.csv" "Fall 2012\_SP.csv" "Fall 2013\_SP.csv"   
## [4] "Fall 2014\_SP.csv" "Fall 2015\_SP.csv" "Fall 2016\_SP.csv"   
## [7] "Spring 2012\_SP.csv" "Spring 2013\_SP.csv" "Spring 2014\_SP.csv"  
## [10] "Spring 2015\_SP.csv" "Spring 2016\_SP.csv" "Spring 2017\_SP.csv"  
## [13] "Sum 2012.csv" "Sum 2013.csv" "Sum 2014.csv"   
## [16] "Sum 2015.csv" "Sum 2016.csv" "Sum 2017.csv"

# ldply() function in plyr helps to read all files together and convert to data frame  
student\_progress <- ldply(allfiles, read\_csv)

# STEP2: FORMING SUBSET

# Selecting only 3 important variables(on basis of hypothesis) from student\_progress data: student id, academic year, Term GPA

**CODE:**

#STEP2:   
#student id, academic year, Term GPA represented by their index.  
progress = student\_progress[c(1,5,16)]

# STEP3: DATA TRANSFORMATION:

# A. MEAN BY ACADEMIC YEAR AND STUDENT ID

# This will help to calculate mean of all terms gpa by student id and academic year.

**CODE:**

# using data table to do the process.  
foo <- setDT(progress)[, mean(TermGPA), .(StudentID, AcademicYear)]

## 

## B. DCAST:

## It converts long format to wide format by changing values in academic year as columns . v1 contains values as mean term gpa

# each year range which got transformed gets value = v1 ie mean of termgpa by student id and academic year  
df2 = dcast(foo, StudentID~AcademicYear, value.var = "V1")

# STEP4: CONVERTING VECTOR TO DATA FRAME:

# Converting df2 to a data frame.Since df2 formed by dcast method outputs a vector type. so need to convert to dataframe otherwise produces error.

**CODE:**

df2 = data.frame(df2)  
head(df2)

## StudentID X2011.12 X2012.13 X2013.14 X2014.15 X2015.16 X2016.17  
## 1 20932 NA NA NA 0.000 NA NA  
## 2 21868 NA NA NA 3.635 3.925 3.966667  
## 3 21943 NA 0.00 NA NA NA NA  
## 4 22011 3.59 3.05 3.343333 3.245 NA NA  
## 5 22163 NA NA 3.150000 3.350 3.000 4.000000  
## 6 22672 NA NA 2.100000 0.000 NA 1.850000

# df2 DATAFRAME WILL BE USED FURTHER AS PROGRESS DATA FRAME with 13767 obs and 7 variables

## 

## UNDERSTANDING PROGRESS DATA:

**STEP1: LOOKING FIRST 6 OBSERVATIONS:**

**CODE:**

head(student\_progress)

## StudentID Cohort CohortTerm Term AcademicYear CompleteDevMath  
## 1 285848 2011-12 1 1 2011-12 -2  
## 2 302176 2011-12 1 1 2011-12 -2  
## 3 301803 2011-12 1 1 2011-12 -2  
## 4 302756 2011-12 1 1 2011-12 -2  
## 5 300304 2011-12 1 1 2011-12 0  
## 6 301067 2011-12 1 1 2011-12 -2  
## CompleteDevEnglish Major1 Major2 Complete1 Complete2 CompleteCIP1  
## 1 -2 51.3899 -1.0000 0 0 -2  
## 2 -2 51.3801 -1.0000 0 0 -2  
## 3 -2 51.3899 -1.0000 0 0 -2  
## 4 -2 45.0601 -1.0000 0 0 -2  
## 5 -2 9.0101 -1.0000 0 0 -2  
## 6 -2 23.0101 13.1001 0 0 -2  
## CompleteCIP2 TransferIntent DegreeTypeSought TermGPA CumGPA  
## 1 -2 -1 6 3.25 3.25  
## 2 -2 -1 6 3.00 3.00  
## 3 -2 -1 6 3.91 3.91  
## 4 -2 -1 6 3.78 3.78  
## 5 -2 -1 6 1.57 1.57  
## 6 -2 -1 6 4.00 4.00

# missing values:  
#missing = sapply(student\_progress, function(x) sum(is.na(x)))  
#missing  
  
  
  
# checking for unique values  
unique(student\_progress$CompleteCIP2) # all -2 no use remove

## [1] -2

unique(student\_progress$TransferIntent) # all -1 # no use will remove

## [1] -1

unique(student\_progress$CompleteDevMath) # it has some value as -1, -2 as missing values # all -1 so remove

## [1] -2 0 -1 1

unique(student\_progress$DegreeTypeSought) # all 6 # that means all same bachelor degree

## [1] 6

## 

## CONCLUSION :

1. **Complete CIPI2**: It is of the field in which the student received his or her second highest award. Since this column contains all value = -2 , which means that cip does not apply to any student and so not provide any useful information.
2. **Transfer Intent** : it contains all values equall to -1 , and so is also of not much useful.
3. **Degree Type Sought**: It contains all values as -6 , which means all student with same bachelor degree.

**PART2:**  **READING STATIC FILE:**

It contains information related to demographics and academic background:Data that do not change over time, such as demographics and academic background. It is collected Semi-annually, at each submission cycle.

**PROCESS:**

Since the static folder contains various files, I applied loop using ldply to read and combine all data into one Data Frame.

# READ STATIC DATA

**CODE:**

#STEP1: READING STUDENT\_STATIC DATA  
  
setwd("C:\\Users\\pooja\\R\_STUDIO\_FILES\\R\_FILES\_PROJECT\_530\\Student Retention Challenge Data\\Student Static Data")  
myfolder = "Student Static Data"  
getwd()

## [1] "C:/Users/pooja/R\_STUDIO\_FILES/R\_FILES\_PROJECT\_530/Student Retention Challenge Data/Student Static Data"

allfiles = list.files()  
allfiles

## [1] "Fall 2011\_ST.csv" "Fall 2012.csv" "Fall 2013.csv"   
## [4] "Fall 2014.csv" "Fall 2015.csv" "Fall 2016.csv"   
## [7] "Spring 2012\_ST.csv" "Spring 2013.csv" "Spring 2014.csv"   
## [10] "Spring 2015.csv" "Spring 2016.csv"

list.files()

## [1] "Fall 2011\_ST.csv" "Fall 2012.csv" "Fall 2013.csv"   
## [4] "Fall 2014.csv" "Fall 2015.csv" "Fall 2016.csv"   
## [7] "Spring 2012\_ST.csv" "Spring 2013.csv" "Spring 2014.csv"   
## [10] "Spring 2015.csv" "Spring 2016.csv"

# COMBINING: combined DIIFERENT CSV FILES FROM STUDENT STATIC FOLDER TO ONE df2  
student\_static <- ldply(allfiles, read\_csv)  
head(student\_static)

## StudentID Cohort CohortTerm Campus Address1 Address2 City  
## 1 285848 2011-12 1 NA 328 Adams St Apt 1 <NA> Hoboken  
## 2 302176 2011-12 1 NA 142 Cherry St <NA> Jersey City  
## 3 301803 2011-12 1 NA 12 Rainbow Street <NA> Presque Isle  
## 4 302756 2011-12 1 NA 345 4th St Apt 2 <NA> Jersey City  
## 5 300304 2011-12 1 NA 6600 Broadway Apt 3D West New York  
## 6 301067 2011-12 1 NA 240 3rd St <NA> Jersey City  
## State Zip RegistrationDate Gender BirthYear BirthMonth Hispanic  
## 1 NJ 7030 20110808 2 1978 9 0  
## 2 NJ 7305 20110804 1 1970 4 0  
## 3 ME 4769 20110809 2 1984 4 0  
## 4 NJ 7302 20110823 2 1986 1 0  
## 5 NJ 7093 20110725 1 1992 2 1  
## 6 NJ 7302 20110420 1 1969 4 0  
## AmericanIndian Asian Black NativeHawaiian White TwoOrMoreRace HSDip HSDipYr  
## 1 0 0 0 0 1 0 1 -1  
## 2 0 0 0 0 1 0 1 -1  
## 3 0 0 0 0 1 0 1 -1  
## 4 0 0 0 0 1 0 -1 -1  
## 5 0 0 0 0 0 0 1 2010  
## 6 0 0 0 0 1 0 1 -1  
## HSGPAUnwtd HSGPAWtd FirstGen DualHSSummerEnroll EnrollmentStatus  
## 1 -1.00 -1 -1 0 2  
## 2 -1.00 -1 -1 0 2  
## 3 -1.00 -1 -1 0 2  
## 4 -1.00 -1 -1 0 2  
## 5 3.13 -1 -1 0 1  
## 6 -1.00 -1 -1 0 2  
## NumColCredAttemptTransfer NumColCredAcceptTransfer CumLoanAtEntry HighDeg  
## 1 0 0.0 -1 0  
## 2 96 45.0 -1 0  
## 3 0 0.0 -1 0  
## 4 54 87.5 -1 0  
## 5 -2 -2.0 -2 0  
## 6 70 66.0 -1 2  
## MathPlacement EngPlacement GatewayMathStatus GatewayEnglishStatus  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 1 0 0 0  
## 6 0 0 0 0

# student\_static has 13261, 35 variables

# 

# STEP2: FORMING SUBSET :

# Taking subset of student\_static by removing cohort, cohort term , campus, address2, hsgpawntd, firstgen, dualsummer, enrollmentstatus, cumloanentry, highdeg, gateways, numcolcredattempt:

**CODE:**

# Checking each variable index as will use to remove variables using their index.  
#data.frame(colnames(student\_static))  
  
student\_static\_1 = student\_static[-c(2, 3, 4, 6, 22, 24,25, 26, 27, 30, 33, 34, 35)]  
# removing numcolcredattempt  
student\_static\_1 = student\_static\_1[-c(19,20)]  
  
# STUDENT STATIC1 HAS 20 VARIABLE and will be used further as student static data

## 

## UNDERSTANDING STATIC DATA:

**STEP1: Used unique() to find all unique values .**

**CODE:**

#unique(student\_progress$Hsdip) # has some -1 as missing so this can be useful so will not remove  
#unique(student\_progress$Hsgpaunwtd) # this is fine no remove  
#unique(student\_progress$Hsgpawtd) # all -1 so remove  
#unique(student\_progress$CumloanatEntry) # all -1 or -2 so no use remove  
#unique(student\_progress$CompleteDevEnglish)

## CONCLUSION :

1. **Hsgpawtd** : It has all values as -1.
2. **Cum loan entry** : If available from administrative records, the amount of known debt the student had accumulated at time of entry to the institution. Value -1 = Missing (Known transfer student, previous cumulative loan amount unknown). Since it has all values either as -1 or -2 so -2 = Known not to be a transfer student, does not apply.

## PART 3 : MERGING STATIC FILE AND PROGRESS BY USING INNER JOIN:

## After getting both files (static and progress) , we will merge both of these files using inner join. Since static has all id as unique, so inner join will help to extract all those common id in both datasets.Also, static has only data for fall and spring but no summer , so outer join will of no much use.

**CODE:**

progress\_static\_inner<- merge(x = df2, y = student\_static\_1, by = c("StudentID"))  
  
# progress static inner has 13261 obs and 26 variable

## PART4 : READING FINANCE DATA

**Financial Aid Data was collected for each student for each academic year, and it is stored in different columns for different years. It contains Financial Aid and other related information such as scholarships, loans, gross income etc.**

**PROCESS:**

**STEP1: READ FINANCE DATA**

**CODE:**

# READING FINANCE DATA  
student\_finance = read.csv("C:\\Users\\pooja\\R\_STUDIO\_FILES\\R\_FILES\_PROJECT\_530\\Student Retention Challenge Data\\Student Financial Aid Data\\2011-2017\_Cohorts\_Financial\_Aid\_and\_Fafsa\_Data.csv")  
head(student\_finance)

## ID.with.leading cohort cohort.term Marital.Status Adjusted.Gross.Income  
## 1 297957 2011-12 1 Single 0  
## 2 302040 2011-12 1 Single 18096  
## 3 234532 2011-12 1 Single 12383  
## 4 303486 2011-12 1 Married 59303  
## 5 304316 2011-12 1 Single 25133  
## 6 302808 2011-12 1 Single 15971  
## Parent.Adjusted.Gross.Income Father.s.Highest.Grade.Level  
## 1 0 College  
## 2 0 High School  
## 3 0 High School  
## 4 0 High School  
## 5 0 Unknown  
## 6 0 Middle School  
## Mother.s.Highest.Grade.Level Housing X2012.Loan X2012.Scholarship  
## 1 High School On Campus Housing 3500 NA  
## 2 High School Off Campus 12500 NA  
## 3 High School Off Campus NA NA  
## 4 Middle School Off Campus 4750 NA  
## 5 High School NA NA  
## 6 High School Off Campus 6500 NA  
## X2012.Work.Study X2012.Grant X2013.Loan X2013.Scholarship X2013.Work.Study  
## 1 NA 10714 5500 NA NA  
## 2 NA 3500 6250 NA NA  
## 3 NA 7432 5500 NA NA  
## 4 NA 850 2750 NA NA  
## 5 NA NA NA NA NA  
## 6 NA 5550 8000 NA NA  
## X2013.Grant X2014.Loan X2014.Scholarship X2014.Work.Study X2014.Grant  
## 1 11095 NA NA NA NA  
## 2 NA NA NA NA NA  
## 3 NA NA NA NA NA  
## 4 1650 10500 NA NA 3146  
## 5 NA NA NA NA NA  
## 6 2888 NA NA NA NA  
## X2015.Loan X2015.Scholarship X2015.Work.Study X2015.Grant X2016.Loan  
## 1 NA NA NA NA NA  
## 2 NA NA NA NA NA  
## 3 NA NA NA NA NA  
## 4 5206 NA NA 4580 NA  
## 5 NA NA NA NA NA  
## 6 NA NA NA NA NA  
## X2016.Scholarship X2016.Work.Study X2016.Grant X2017.Loan X2017.Scholarship  
## 1 NA NA NA NA NA  
## 2 NA NA NA NA NA  
## 3 NA NA NA NA NA  
## 4 NA NA 691 8385 NA  
## 5 NA NA NA NA NA  
## 6 NA NA NA NA NA  
## X2017.Work.Study X2017.Grant  
## 1 NA NA  
## 2 NA NA  
## 3 NA NA  
## 4 NA 2233  
## 5 NA NA  
## 6 NA NA

# STEP 2: RENAMING LEADING STUDENT ID WITH STUDENT ID

# Renaming Leading Student id with student id in financial aid data set for inner join to work.

**CODE:**

student\_finance = dplyr::rename(student\_finance, StudentID = ID.with.leading)

# STEP 3: FORMING SUBSET OF FINANCE DATA SET :

# Forming subset by removing variables cohort, cohort term, X2012.Scholarship , X2012.Work.Study ,X2012.Grant, X2013.Loan,X2013.Scholarship X2013.Work.Study, X2013.Grant, X2014.Loan ,X2014.Scholarship, X2014.Work Study .On basis of hypothesis and also, most of these variables does not provide important information, so will not consider them on new subset.

**CODE:**

# It helps to retrieve index for each variable , thus making easy to remove variables by using index.  
#data.frame(colnames(student\_finance))  
  
# This will make subset student\_finance1 by removing all above variables from student finance.  
student\_finance1 = student\_finance[-c(2, 3, 10: 33)]  
  
  
# STUDENT\_FINANCE1 HAS 13769 rows and 7 VARIABLES AND WILL BE USED AS STUDENT FINANCIAL AID INFORMATION

## 

## STEP 4: UNDERSTANDING FINANCE DATA:

head(student\_finance)

## StudentID cohort cohort.term Marital.Status Adjusted.Gross.Income  
## 1 297957 2011-12 1 Single 0  
## 2 302040 2011-12 1 Single 18096  
## 3 234532 2011-12 1 Single 12383  
## 4 303486 2011-12 1 Married 59303  
## 5 304316 2011-12 1 Single 25133  
## 6 302808 2011-12 1 Single 15971  
## Parent.Adjusted.Gross.Income Father.s.Highest.Grade.Level  
## 1 0 College  
## 2 0 High School  
## 3 0 High School  
## 4 0 High School  
## 5 0 Unknown  
## 6 0 Middle School  
## Mother.s.Highest.Grade.Level Housing X2012.Loan X2012.Scholarship  
## 1 High School On Campus Housing 3500 NA  
## 2 High School Off Campus 12500 NA  
## 3 High School Off Campus NA NA  
## 4 Middle School Off Campus 4750 NA  
## 5 High School NA NA  
## 6 High School Off Campus 6500 NA  
## X2012.Work.Study X2012.Grant X2013.Loan X2013.Scholarship X2013.Work.Study  
## 1 NA 10714 5500 NA NA  
## 2 NA 3500 6250 NA NA  
## 3 NA 7432 5500 NA NA  
## 4 NA 850 2750 NA NA  
## 5 NA NA NA NA NA  
## 6 NA 5550 8000 NA NA  
## X2013.Grant X2014.Loan X2014.Scholarship X2014.Work.Study X2014.Grant  
## 1 11095 NA NA NA NA  
## 2 NA NA NA NA NA  
## 3 NA NA NA NA NA  
## 4 1650 10500 NA NA 3146  
## 5 NA NA NA NA NA  
## 6 2888 NA NA NA NA  
## X2015.Loan X2015.Scholarship X2015.Work.Study X2015.Grant X2016.Loan  
## 1 NA NA NA NA NA  
## 2 NA NA NA NA NA  
## 3 NA NA NA NA NA  
## 4 5206 NA NA 4580 NA  
## 5 NA NA NA NA NA  
## 6 NA NA NA NA NA  
## X2016.Scholarship X2016.Work.Study X2016.Grant X2017.Loan X2017.Scholarship  
## 1 NA NA NA NA NA  
## 2 NA NA NA NA NA  
## 3 NA NA NA NA NA  
## 4 NA NA 691 8385 NA  
## 5 NA NA NA NA NA  
## 6 NA NA NA NA NA  
## X2017.Work.Study X2017.Grant  
## 1 NA NA  
## 2 NA NA  
## 3 NA NA  
## 4 NA 2233  
## 5 NA NA  
## 6 NA NA

## 

## CONCLUSION:

1. **All loans column has Nas as value so does not provide us with much information and so removed from data set.**

# 

# PART 5 : MERGING ALL 3 FILES TOGETHER USING INNER JOIN:

# Merging all 3 files will help to get information of each student related to their static, progress and finance.

**CODE:**

# STEP5: COMBINING FINANCIAL DATA SET WITH OTHER2 .   
studentdetail\_inner <- merge(x = student\_finance1, y = progress\_static\_inner, by = c("StudentID"))  
  
  
# studentdetail\_inner has 13261 and 32 variable

## NOTE: MERGED DATA SET IS GIVEN BY studentdetail\_inner data set AND HAS 13261 rows and 32 columns

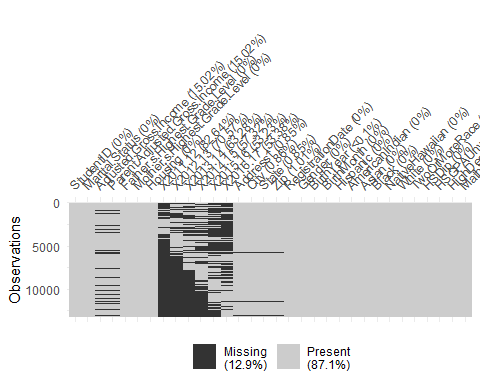
## 

## PART 6 : DATA CLEANING :

## STEP1: MISSING VALUES:

# PLOTTING TO SEE MISSING VALUES USING LIBRARY VISDAT:

vis\_miss(studentdetail\_inner)



**FACTS:**

1. Data set has almost na in all rows , so removing missing values from rows is not possible.

2: **METHODS FOR REMOVING** /**REPLACING MISSING VALUES**

1. Imputation

2. Replace using KNN.

3. Replace numerical values by their mean or median.

4. Replace categorical values by their majority.

5. Delete by columns.

**CATEGORICAL VARIABLES IN DATA SET CONTAINING SOME BLANK SPACES:**

1. **Marital.Status**: Replaced blank space by “Single” as it is in the majority.
2. **Housing** : Replaced by “OffCampus”.
3. **Father.S.Highest.Grade.Level** was replaced by “Unknown” as although most are high school but replacing it by real values(high school) is not good.
4. **Mother.S.Highest.Grade.Level** was replaced by “Unknown”.

III. **NUMERICAL VARIABLES IN DATA SET CONTAINING MISSING VALUES WERE REPLACED BY THEIR MEAN:**

1. Parent Gross Income replaced by mean
2. Adjusted gross income replaced by mean.
3. Other Numerical values was replaced by 0 , using input method

**III. DELETE BY COLUMNS: WHOLE COLUMN WAS DELETED**

1. Degree type sought was deleted as it has all value = 6 , which means it did not provide any new information
2. Transferintent was deleted as has all values as -1
3. CompleteCIP2 was deleted as has all values as -2
4. Most loans variable conatained na in most of their rows , so not useful for our modelling.

**CODE:**

## PART1: NUMERICAL VALUES

## STEP1 : CHECKING IF ANY NA

# STEP 1: checking if any na   
#is.na(studentdetail\_inner)  
  
  
#counting nunber of na values  
sum(is.na(studentdetail\_inner))

## [1] 54906

# STEP2: VIEW ALL MISSING VALUES IN MERGED DATA SET:   
#missing = sapply(studentdetail\_inner, function(x) sum(is.na(x)))  
missing

## function (x) .Primitive("missing")

## 

## REPLACING NA OF NUMERICAL VALUES.

# STEP3: FOR NUMERICAL VALUES   
  
# FOR ADJUSTED GROSS INCOME  
studentdetail\_inner$Adjusted.Gross.Income[is.na(studentdetail\_inner$Adjusted.Gross.Income)] <- mean(studentdetail\_inner$Adjusted.Gross.Income, na.rm = T)  
sum(is.na(studentdetail\_inner$adjustedgrossincome))

## [1] 0

# for parent ADJUSTED gross  
sum(is.na(studentdetail\_inner$Parent.Adjusted.Gross.Income))

## [1] 1992

studentdetail\_inner$Parent.Adjusted.Gross.Income[is.na(studentdetail\_inner$Parent.Adjusted.Gross.Income)] <- mean(studentdetail\_inner$Parent.Adjusted.Gross.Income, na.rm = T)  
sum(is.na(studentdetail\_inner$Parent.Adjusted.Gross.Income))

## [1] 0

# USING IMPUTETS LIBRARY TO REPLACE OTHER NUMERICAL VARIABLE NA BY 0 .  
studentdetail\_inner<- na.replace(studentdetail\_inner, 0)

## 

## FOR CATEGORICAL VALUES:

# BLANK SPACES:  
  
# FOR MARITAL STATUS, checking blank spaces  
  
#sum(is.na(studentdetail\_inner$Marital.Status))  
#unique(studentdetail\_inner$Marital.Status)  
#studentdetail\_inner$Marital.Status   
# CONCLUSION : IT HAS BLANK SPACES   
  
# SOLUTION : REPLACING BLANK WITH each categorical majority   
  
# FOR MARITAL  
studentdetail\_inner$Marital.Status <- sub("^$", "Single", studentdetail\_inner$Marital.Status)  
  
  
  
  
# FOR HOUSING  
studentdetail\_inner$Housing <- sub("^$", "Off Campus", studentdetail\_inner$Housing)  
  
  
  
# for grade level as important factor : so will check if -1 or blank  
#studentdetail\_inner$Father.s.Highest.Grade.Level # empty space  
studentdetail\_inner$Father.s.Highest.Grade.Level<- sub("^$", "Unknown", studentdetail\_inner$Father.s.Highest.Grade.Level)  
  
# For mother grade level: there is blank  
#studentdetail\_inner$Mother.s.Highest.Grade.Level  
studentdetail\_inner$Mother.s.Highest.Grade.Level<- sub("^$", "Unknown", studentdetail\_inner$Mother.s.Highest.Grade.Level)  
  
  
  
# for CATEGORICAL STATE, ADDRESS, CITY  
  
# replace na in city by jersey city USING LIBRARY VIM AND FUNCTION KNN  
  
# FOR CITY  
studentdetail\_inner <- kNN(data = studentdetail\_inner, variable = 'City', k=6, imp\_var=FALSE)  
# FOR STATE  
studentdetail\_inner <- kNN(data = studentdetail\_inner, variable = 'State', k=6, imp\_var=FALSE)  
# FOR ADDRESS  
studentdetail\_inner <- kNN(data = studentdetail\_inner, variable = 'Address1', k=6, imp\_var=FALSE)  
  
  
# RECHECKING ANY MISSING VALUES:   
#missing = sapply(studentdetail\_inner, function(x) sum(is.na(x)))  
#missing

## 

## QUALITY ACESS:

summary(studentdetail\_inner)

## StudentID Marital.Status Adjusted.Gross.Income  
## Min. : 20932 Length:13261 Min. : -24326   
## 1st Qu.:305254 Class :character 1st Qu.: 0   
## Median :321478 Mode :character Median : 6745   
## Mean :316151 Mean : 13046   
## 3rd Qu.:343511 3rd Qu.: 13265   
## Max. :359783 Max. :2576425   
## Parent.Adjusted.Gross.Income Father.s.Highest.Grade.Level  
## Min. :-62979 Length:13261   
## 1st Qu.: 0 Class :character   
## Median : 20102 Mode :character   
## Mean : 28317   
## 3rd Qu.: 32999   
## Max. :657631   
## Mother.s.Highest.Grade.Level Housing X2011.12   
## Length:13261 Length:13261 Min. :0.0000   
## Class :character Class :character 1st Qu.:0.0000   
## Mode :character Mode :character Median :0.0000   
## Mean :0.4829   
## 3rd Qu.:0.0000   
## Max. :4.0000   
## X2012.13 X2013.14 X2014.15 X2015.16   
## Min. :0.0000 Min. :0.000 Min. :0.000 Min. :0.000   
## 1st Qu.:0.0000 1st Qu.:0.000 1st Qu.:0.000 1st Qu.:0.000   
## Median :0.0000 Median :0.000 Median :0.000 Median :0.000   
## Mean :0.8327 Mean :1.042 Mean :1.223 Mean :1.314   
## 3rd Qu.:1.8000 3rd Qu.:2.655 3rd Qu.:2.940 3rd Qu.:3.000   
## Max. :4.0000 Max. :4.000 Max. :4.000 Max. :4.000   
## X2016.17 Address1 City State   
## Min. :0.000 Length:13261 Length:13261 Length:13261   
## 1st Qu.:0.000 Class :character Class :character Class :character   
## Median :0.000 Mode :character Mode :character Mode :character   
## Mean :1.285   
## 3rd Qu.:2.935   
## Max. :4.000   
## Zip RegistrationDate Gender BirthYear   
## Min. : 0 Min. :20110111 Min. :1.000 Min. : 0   
## 1st Qu.: 7052 1st Qu.:20120710 1st Qu.:1.000 1st Qu.:1986   
## Median : 7304 Median :20140121 Median :2.000 Median :1992   
## Mean : 7712 Mean :20136109 Mean :1.596 Mean :1989   
## 3rd Qu.: 7307 3rd Qu.:20150624 3rd Qu.:2.000 3rd Qu.:1995   
## Max. :98118 Max. :20160912 Max. :2.000 Max. :2000   
## BirthMonth Hispanic AmericanIndian Asian   
## Min. : 1.000 Min. :-1.0000 Min. :-1.00000 Min. :-1.00000   
## 1st Qu.: 4.000 1st Qu.: 0.0000 1st Qu.: 0.00000 1st Qu.: 0.00000   
## Median : 7.000 Median : 0.0000 Median : 0.00000 Median : 0.00000   
## Mean : 6.581 Mean : 0.2568 Mean :-0.06742 Mean : 0.01848   
## 3rd Qu.:10.000 3rd Qu.: 1.0000 3rd Qu.: 0.00000 3rd Qu.: 0.00000   
## Max. :12.000 Max. : 1.0000 Max. : 1.00000 Max. : 1.00000   
## Black NativeHawaiian White TwoOrMoreRace   
## Min. :-1.0000 Min. :-1.00000 Min. :-1.000 Min. :-1.00000   
## 1st Qu.: 0.0000 1st Qu.: 0.00000 1st Qu.: 0.000 1st Qu.: 0.00000   
## Median : 0.0000 Median : 0.00000 Median : 0.000 Median : 0.00000   
## Mean : 0.1447 Mean :-0.06757 Mean : 0.183 Mean :-0.05181   
## 3rd Qu.: 0.0000 3rd Qu.: 0.00000 3rd Qu.: 1.000 3rd Qu.: 0.00000   
## Max. : 1.0000 Max. : 1.00000 Max. : 1.000 Max. : 1.00000   
## HSDip HSGPAUnwtd HighDeg MathPlacement   
## Min. :-1.0000 Min. :-1.0000 Min. :0.0000 Min. :-1.0000   
## 1st Qu.: 1.0000 1st Qu.:-1.0000 1st Qu.:0.0000 1st Qu.: 0.0000   
## Median : 1.0000 Median :-1.0000 Median :0.0000 Median : 0.0000   
## Mean : 0.9643 Mean : 0.1624 Mean :0.5849 Mean : 0.2793   
## 3rd Qu.: 1.0000 3rd Qu.: 2.4000 3rd Qu.:2.0000 3rd Qu.: 1.0000   
## Max. : 4.0000 Max. : 4.0000 Max. :4.0000 Max. : 1.0000

## STEP VI: SPLITTING STUDENT DEATIL INNER DATA SET WITH GIVEN TEST DATA SET :

**We used test data set containing only student id , to split our final merged files from all 3 static, progress, finance data set using inner join. By doing this, all unique ids will be only getting attached, containing same number of variable as merged file. This data set formed is named as Kaggletestdata and will not have any dropout label and will be treated as unknown data set for predicting and checking its accuracy. This will be used when our final model is ready to use. Through model, this data set will get label as 0 or 1.**

**CODE:**

# STEP 1: READ TEST DATA  
  
test\_data = read.csv("C:\\Users\\pooja\\R\_STUDIO\_FILES\\R\_FILES\_PROJECT\_530\\Student Retention Challenge Data\\Test Data\\TestIDs.csv")

# PART 7 : MERGING (INNER JOINING) TEST DATA SET WITH STUDENTDETAIL\_INNER DATA SET(CONTAINING ALL 3 FILES):

**CODE:**

Kaggletest = merge(x = studentdetail\_inner, y = test\_data, by = "StudentID")  
head(Kaggletest)

## StudentID Marital.Status Adjusted.Gross.Income Parent.Adjusted.Gross.Income  
## 1 22011 Single 8845.00 0.0  
## 2 25142 Single 9025.00 0.0  
## 3 26586 Single 0.00 41586.0  
## 4 30958 Single 0.00 0.0  
## 5 34474 Single 19340.00 0.0  
## 6 34937 Single 13045.55 28316.5  
## Father.s.Highest.Grade.Level Mother.s.Highest.Grade.Level Housing  
## 1 College College Off Campus  
## 2 Unknown Unknown Off Campus  
## 3 High School College With Parent  
## 4 Middle School College Off Campus  
## 5 Unknown Unknown Off Campus  
## 6 Unknown Unknown Off Campus  
## X2011.12 X2012.13 X2013.14 X2014.15 X2015.16 X2016.17  
## 1 3.59 3.050000 3.343333 3.245 0 0.000000  
## 2 0.00 0.000000 0.000000 0.000 0 3.026667  
## 3 0.00 0.425000 0.000000 0.000 0 0.000000  
## 4 0.00 3.413333 3.763333 0.000 0 0.000000  
## 5 0.00 0.615000 0.000000 0.000 0 0.000000  
## 6 0.00 4.000000 4.000000 4.000 0 0.000000  
## Address1 City State Zip  
## 1 30 Ave at Port Imperial Apt 410 West New York NJ 7030  
## 2 270 Harrison Ave Apt 207 Jersey City NJ 7305  
## 3 624 3rd St Lyndhurst NJ 7071  
## 4 772 Veterans Pl #1 Cliffside Park NJ 7093  
## 5 196 martin luther king dr. Unit 212 Jersey City NJ 7305  
## 6 22 Stegman Ct Jersey City NJ 7305  
## RegistrationDate Gender BirthYear BirthMonth Hispanic AmericanIndian Asian  
## 1 20110613 2 1982 4 1 0 0  
## 2 20160606 2 1980 8 0 0 0  
## 3 20120813 2 1994 9 0 0 0  
## 4 20120731 2 1980 9 -1 -1 -1  
## 5 20120712 2 1982 5 0 0 0  
## 6 20121220 1 1987 3 0 0 1  
## Black NativeHawaiian White TwoOrMoreRace HSDip HSGPAUnwtd HighDeg  
## 1 0 0 0 0 1 -1.0 0  
## 2 1 0 0 0 1 -1.0 2  
## 3 0 0 1 0 1 2.6 0  
## 4 -1 -1 -1 -1 1 -1.0 2  
## 5 1 0 0 0 1 2.1 0  
## 6 0 0 0 0 1 -1.0 3  
## MathPlacement  
## 1 1  
## 2 0  
## 3 1  
## 4 0  
## 5 1  
## 6 0

# 

# STEP3: CHANGING ALL CHAR IN KAGGLE DATA SET TO FACTORS:

**CODE:**

Kaggletest[sapply(Kaggletest, is.character)] <- lapply(Kaggletest[sapply(Kaggletest, is.character)],   
 as.factor)  
# kaggledata set has now 1000 observations and 32 variable

## 

## PART 8: TRAIN DATA SET:

1. Initially, train data csv file contain only student id and label(dropout).
2. In order to train our model, we need data set with some features.
3. For that, We will get train data set by inner joining final merged file( all 3 files) with train data csv file provided,
4. Now, final train data contain all 32 features and also extra dropout label, making it 33 variable.
5. Since attached data set contains drop out as num type, so will change to factor type

## 

## READING DROPOUT LABEL CSV FILE

**PROCESS**:

**STEP1.: READING DROP OUT FILE**

**CODE:**

#1. STEP1: READ DROPOUT DATA CSV FILE  
dropout\_label = read.csv("C:\\Users\\pooja\\R\_STUDIO\_FILES\\R\_FILES\_PROJECT\_530\\Student Retention Challenge Data\\DropoutTrainLabels.csv")

## 

## STEP 2: USING INNER JOIN TO COMBINE ALL THREE DATA SETS:

**CODE:**

#STEP2: USING INNER JOIN TO COMBINE DROP OUT DATA SET WITH STUDENTDETAIL INNER DATA SET AS FINAL TRAIN DATA SET  
final\_traindata = merge(x = studentdetail\_inner, y = dropout\_label, by = "StudentID")  
head(final\_traindata)

## StudentID Marital.Status Adjusted.Gross.Income Parent.Adjusted.Gross.Income  
## 1 20932 Married 52555.00 0.0  
## 2 21868 Single 30600.00 0.0  
## 3 21943 Single 27879.00 0.0  
## 4 22163 Single 26794.00 0.0  
## 5 22672 Single 13045.55 28316.5  
## 6 23538 Single 28376.00 0.0  
## Father.s.Highest.Grade.Level Mother.s.Highest.Grade.Level Housing X2011.12  
## 1 Unknown Unknown Off Campus 0  
## 2 High School High School Off Campus 0  
## 3 Unknown High School Off Campus 0  
## 4 Unknown College Off Campus 0  
## 5 Unknown Unknown Off Campus 0  
## 6 College High School Off Campus 0  
## X2012.13 X2013.14 X2014.15 X2015.16 X2016.17 Address1  
## 1 0 0.000 0.000000 0.000 0.000000 87 W 28th St  
## 2 0 0.000 3.635000 3.925 3.966667 217 Seeley Ave  
## 3 0 0.000 0.000000 0.000 0.000000 7012 Cottage Avenue , 201  
## 4 0 3.150 3.350000 3.000 4.000000 6 Seaview Ct Apt 1233  
## 5 0 2.100 0.000000 0.000 1.850000 55 Manor Dr Apt 14P  
## 6 0 3.785 3.766667 0.000 0.000000 8 Elm Ct Apt #1  
## City State Zip RegistrationDate Gender BirthYear BirthMonth Hispanic  
## 1 Bayonne NJ 7307 20140825 2 1971 4 0  
## 2 Kearny NJ 7032 20140507 2 1980 8 0  
## 3 North Bergen NJ 7047 20120426 2 1982 7 1  
## 4 Bayonne NJ 7002 20131121 2 1982 4 0  
## 5 Newark NJ 7106 20130827 1 1969 3 0  
## 6 Bayonne NJ 7002 20140113 2 1981 6 0  
## AmericanIndian Asian Black NativeHawaiian White TwoOrMoreRace HSDip  
## 1 0 0 1 0 0 0 1  
## 2 0 0 0 0 1 0 1  
## 3 0 0 0 0 0 0 1  
## 4 0 0 1 0 0 0 1  
## 5 0 0 1 0 0 0 1  
## 6 0 1 0 0 0 0 1  
## HSGPAUnwtd HighDeg MathPlacement Dropout  
## 1 -1 0 0 1  
## 2 -1 0 0 0  
## 3 -1 0 0 1  
## 4 -1 0 0 0  
## 5 -1 0 0 1  
## 6 -1 2 0 0

# STEP3: MAKING DROP OUT COLUMN(TARGET VARIABLE) AS FACTOR

final\_traindata$Dropout = as.factor(final\_traindata$Dropout)

# STEP4: MAKING ALL CHAR VARIABLE IN TRAIN DATA TO FACTOR

final\_traindata[sapply(final\_traindata, is.character)] <- lapply(final\_traindata[sapply(final\_traindata, is.character)],   
 as.factor)

## STEPX: Exploratory Data Analysis -

1. student\_progress
2. student\_static
3. student\_finance
4. final\_traindata

## DESCRIPTIVE STATISTICS:

1. Univariate Analysis
2. Bivariate Analysis

## BASIC DESCRIPTIVE STATISTICS

## FOR STUDENT\_PROGRESS DATA SET:

# for progress data set   
  
  
# INITITAL:  
  
  
# STEP1: displaying first 6 observation of data set   
a = head(student\_progress)  
b = head(df2)  
  
# dimension of data frame   
dim(student\_progress)

## [1] 57945 17

str(df2)

## 'data.frame': 13767 obs. of 7 variables:  
## $ StudentID: num 20932 21868 21943 22011 22163 ...  
## $ X2011.12 : num NA NA NA 3.59 NA NA NA NA NA NA ...  
## $ X2012.13 : num NA NA 0 3.05 NA NA NA 2.62 NA NA ...  
## $ X2013.14 : num NA NA NA 3.34 3.15 ...  
## $ X2014.15 : num 0 3.63 NA 3.25 3.35 ...  
## $ X2015.16 : num NA 3.92 NA NA 3 ...  
## $ X2016.17 : num NA 3.97 NA NA 4 ...

# names of columns in final data set  
names(student\_progress)

## [1] "StudentID" "Cohort" "CohortTerm"   
## [4] "Term" "AcademicYear" "CompleteDevMath"   
## [7] "CompleteDevEnglish" "Major1" "Major2"   
## [10] "Complete1" "Complete2" "CompleteCIP1"   
## [13] "CompleteCIP2" "TransferIntent" "DegreeTypeSought"   
## [16] "TermGPA" "CumGPA"

names(df2)

## [1] "StudentID" "X2011.12" "X2012.13" "X2013.14" "X2014.15" "X2015.16"   
## [7] "X2016.17"

# STEP2: STRUCTURE OF DATA TYPES  
  
str(student\_progress)

## 'data.frame': 57945 obs. of 17 variables:  
## $ StudentID : num 285848 302176 301803 302756 300304 ...  
## $ Cohort : chr "2011-12" "2011-12" "2011-12" "2011-12" ...  
## $ CohortTerm : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ Term : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ AcademicYear : chr "2011-12" "2011-12" "2011-12" "2011-12" ...  
## $ CompleteDevMath : num -2 -2 -2 -2 0 -2 -2 -2 -2 -2 ...  
## $ CompleteDevEnglish: num -2 -2 -2 -2 -2 -2 1 -2 -2 -2 ...  
## $ Major1 : num 51.39 51.38 51.39 45.06 9.01 ...  
## $ Major2 : num -1 -1 -1 -1 -1 ...  
## $ Complete1 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Complete2 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ CompleteCIP1 : num -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 ...  
## $ CompleteCIP2 : num -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 ...  
## $ TransferIntent : num -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...  
## $ DegreeTypeSought : num 6 6 6 6 6 6 6 6 6 6 ...  
## $ TermGPA : num 3.25 3 3.91 3.78 1.57 4 3.86 3.7 1.06 2.1 ...  
## $ CumGPA : num 3.25 3 3.91 3.78 1.57 4 3.86 3.7 1.06 2.1 ...

str(df2)

## 'data.frame': 13767 obs. of 7 variables:  
## $ StudentID: num 20932 21868 21943 22011 22163 ...  
## $ X2011.12 : num NA NA NA 3.59 NA NA NA NA NA NA ...  
## $ X2012.13 : num NA NA 0 3.05 NA NA NA 2.62 NA NA ...  
## $ X2013.14 : num NA NA NA 3.34 3.15 ...  
## $ X2014.15 : num 0 3.63 NA 3.25 3.35 ...  
## $ X2015.16 : num NA 3.92 NA NA 3 ...  
## $ X2016.17 : num NA 3.97 NA NA 4 ...

# CONCLUSION : NUMBER OF OBSERVATIONS = NUMBER OF VARIABLES =

##FOR STATIC DATA SET:

# FOR STATIC DATA SET:  
  
  
# STEP1: displaying first 6 observation of data set   
a = head(student\_static)  
b = head(student\_static\_1)  
  
# dimension of data frame   
dim(student\_static)

## [1] 13261 35

# names of columns   
names(student\_static)

## [1] "StudentID" "Cohort"   
## [3] "CohortTerm" "Campus"   
## [5] "Address1" "Address2"   
## [7] "City" "State"   
## [9] "Zip" "RegistrationDate"   
## [11] "Gender" "BirthYear"   
## [13] "BirthMonth" "Hispanic"   
## [15] "AmericanIndian" "Asian"   
## [17] "Black" "NativeHawaiian"   
## [19] "White" "TwoOrMoreRace"   
## [21] "HSDip" "HSDipYr"   
## [23] "HSGPAUnwtd" "HSGPAWtd"   
## [25] "FirstGen" "DualHSSummerEnroll"   
## [27] "EnrollmentStatus" "NumColCredAttemptTransfer"  
## [29] "NumColCredAcceptTransfer" "CumLoanAtEntry"   
## [31] "HighDeg" "MathPlacement"   
## [33] "EngPlacement" "GatewayMathStatus"   
## [35] "GatewayEnglishStatus"

names(student\_static\_1)

## [1] "StudentID" "Address1" "City" "State"   
## [5] "Zip" "RegistrationDate" "Gender" "BirthYear"   
## [9] "BirthMonth" "Hispanic" "AmericanIndian" "Asian"   
## [13] "Black" "NativeHawaiian" "White" "TwoOrMoreRace"   
## [17] "HSDip" "HSGPAUnwtd" "HighDeg" "MathPlacement"

# STEP2: STRUCTURE OF DATA TYPES  
  
str(student\_static)

## 'data.frame': 13261 obs. of 35 variables:  
## $ StudentID : num 285848 302176 301803 302756 300304 ...  
## $ Cohort : chr "2011-12" "2011-12" "2011-12" "2011-12" ...  
## $ CohortTerm : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ Campus : logi NA NA NA NA NA NA ...  
## $ Address1 : chr "328 Adams St Apt 1" "142 Cherry St" "12 Rainbow Street" "345 4th St Apt 2" ...  
## $ Address2 : chr NA NA NA NA ...  
## $ City : chr "Hoboken" "Jersey City" "Presque Isle" "Jersey City" ...  
## $ State : chr "NJ" "NJ" "ME" "NJ" ...  
## $ Zip : num 7030 7305 4769 7302 7093 ...  
## $ RegistrationDate : num 20110808 20110804 20110809 20110823 20110725 ...  
## $ Gender : num 2 1 2 2 1 1 2 2 1 2 ...  
## $ BirthYear : num 1978 1970 1984 1986 1992 ...  
## $ BirthMonth : num 9 4 4 1 2 4 8 8 6 12 ...  
## $ Hispanic : num 0 0 0 0 1 0 0 -1 -1 1 ...  
## $ AmericanIndian : num 0 0 0 0 0 0 0 -1 -1 0 ...  
## $ Asian : num 0 0 0 0 0 0 0 -1 -1 0 ...  
## $ Black : num 0 0 0 0 0 0 1 -1 -1 0 ...  
## $ NativeHawaiian : num 0 0 0 0 0 0 0 -1 -1 0 ...  
## $ White : num 1 1 1 1 0 1 0 -1 -1 0 ...  
## $ TwoOrMoreRace : num 0 0 0 0 0 0 0 -1 -1 0 ...  
## $ HSDip : num 1 1 1 -1 1 1 1 1 1 1 ...  
## $ HSDipYr : num -1 -1 -1 -1 2010 -1 2010 -1 -1 -1 ...  
## $ HSGPAUnwtd : num -1 -1 -1 -1 3.13 -1 3.5 -1 -1 -1 ...  
## $ HSGPAWtd : num -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...  
## $ FirstGen : num -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...  
## $ DualHSSummerEnroll : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ EnrollmentStatus : num 2 2 2 2 1 2 1 2 2 2 ...  
## $ NumColCredAttemptTransfer: num 0 96 0 54 -2 70 -2 62 53 52 ...  
## $ NumColCredAcceptTransfer : num 0 45 0 87.5 -2 66 -2 66 45 66 ...  
## $ CumLoanAtEntry : num -1 -1 -1 -1 -2 -1 -2 -1 -1 -1 ...  
## $ HighDeg : num 0 0 0 0 0 2 0 2 0 0 ...  
## $ MathPlacement : num 0 0 0 0 1 0 0 0 0 0 ...  
## $ EngPlacement : num 0 0 0 0 0 0 1 0 0 0 ...  
## $ GatewayMathStatus : num 0 0 0 0 0 0 1 0 0 0 ...  
## $ GatewayEnglishStatus : num 0 0 0 0 0 0 0 0 0 0 ...

str(student\_static\_1)

## 'data.frame': 13261 obs. of 20 variables:  
## $ StudentID : num 285848 302176 301803 302756 300304 ...  
## $ Address1 : chr "328 Adams St Apt 1" "142 Cherry St" "12 Rainbow Street" "345 4th St Apt 2" ...  
## $ City : chr "Hoboken" "Jersey City" "Presque Isle" "Jersey City" ...  
## $ State : chr "NJ" "NJ" "ME" "NJ" ...  
## $ Zip : num 7030 7305 4769 7302 7093 ...  
## $ RegistrationDate: num 20110808 20110804 20110809 20110823 20110725 ...  
## $ Gender : num 2 1 2 2 1 1 2 2 1 2 ...  
## $ BirthYear : num 1978 1970 1984 1986 1992 ...  
## $ BirthMonth : num 9 4 4 1 2 4 8 8 6 12 ...  
## $ Hispanic : num 0 0 0 0 1 0 0 -1 -1 1 ...  
## $ AmericanIndian : num 0 0 0 0 0 0 0 -1 -1 0 ...  
## $ Asian : num 0 0 0 0 0 0 0 -1 -1 0 ...  
## $ Black : num 0 0 0 0 0 0 1 -1 -1 0 ...  
## $ NativeHawaiian : num 0 0 0 0 0 0 0 -1 -1 0 ...  
## $ White : num 1 1 1 1 0 1 0 -1 -1 0 ...  
## $ TwoOrMoreRace : num 0 0 0 0 0 0 0 -1 -1 0 ...  
## $ HSDip : num 1 1 1 -1 1 1 1 1 1 1 ...  
## $ HSGPAUnwtd : num -1 -1 -1 -1 3.13 -1 3.5 -1 -1 -1 ...  
## $ HighDeg : num 0 0 0 0 0 2 0 2 0 0 ...  
## $ MathPlacement : num 0 0 0 0 1 0 0 0 0 0 ...

# CONCLUSION : NUMBER OF OBSERVATIONS = NUMBER OF VARIABLES =

## FOR FINANCE DATA SET

# FOR STATIC DATA SET:  
  
  
# STEP1: displaying first 6 observation of data set   
a = head(student\_finance)  
b = head(student\_finance1)  
  
# dimension of data frame   
dim(student\_finance)

## [1] 13769 33

dim(student\_finance1)

## [1] 13769 7

# names of columns   
names(student\_finance)

## [1] "StudentID" "cohort"   
## [3] "cohort.term" "Marital.Status"   
## [5] "Adjusted.Gross.Income" "Parent.Adjusted.Gross.Income"  
## [7] "Father.s.Highest.Grade.Level" "Mother.s.Highest.Grade.Level"  
## [9] "Housing" "X2012.Loan"   
## [11] "X2012.Scholarship" "X2012.Work.Study"   
## [13] "X2012.Grant" "X2013.Loan"   
## [15] "X2013.Scholarship" "X2013.Work.Study"   
## [17] "X2013.Grant" "X2014.Loan"   
## [19] "X2014.Scholarship" "X2014.Work.Study"   
## [21] "X2014.Grant" "X2015.Loan"   
## [23] "X2015.Scholarship" "X2015.Work.Study"   
## [25] "X2015.Grant" "X2016.Loan"   
## [27] "X2016.Scholarship" "X2016.Work.Study"   
## [29] "X2016.Grant" "X2017.Loan"   
## [31] "X2017.Scholarship" "X2017.Work.Study"   
## [33] "X2017.Grant"

names(student\_finance1)

## [1] "StudentID" "Marital.Status"   
## [3] "Adjusted.Gross.Income" "Parent.Adjusted.Gross.Income"  
## [5] "Father.s.Highest.Grade.Level" "Mother.s.Highest.Grade.Level"  
## [7] "Housing"

# STEP2: STRUCTURE OF DATA TYPES  
  
str(student\_finance)

## 'data.frame': 13769 obs. of 33 variables:  
## $ StudentID : int 297957 302040 234532 303486 304316 302808 304266 292741 296902 302171 ...  
## $ cohort : chr "2011-12" "2011-12" "2011-12" "2011-12" ...  
## $ cohort.term : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Marital.Status : chr "Single" "Single" "Single" "Married" ...  
## $ Adjusted.Gross.Income : int 0 18096 12383 59303 25133 15971 0 0 19709 42557 ...  
## $ Parent.Adjusted.Gross.Income: int 0 0 0 0 0 0 10551 87131 0 0 ...  
## $ Father.s.Highest.Grade.Level: chr "College" "High School" "High School" "High School" ...  
## $ Mother.s.Highest.Grade.Level: chr "High School" "High School" "High School" "Middle School" ...  
## $ Housing : chr "On Campus Housing" "Off Campus" "Off Campus" "Off Campus" ...  
## $ X2012.Loan : int 3500 12500 NA 4750 NA 6500 12500 10500 2500 3750 ...  
## $ X2012.Scholarship : num NA NA NA NA NA ...  
## $ X2012.Work.Study : num NA NA NA NA NA NA NA NA NA NA ...  
## $ X2012.Grant : num 10714 3500 7432 850 NA ...  
## $ X2013.Loan : num 5500 6250 5500 2750 NA 8000 10500 12500 NA NA ...  
## $ X2013.Scholarship : num NA NA NA NA NA ...  
## $ X2013.Work.Study : int NA NA NA NA NA NA NA NA NA NA ...  
## $ X2013.Grant : num 11095 NA NA 1650 NA ...  
## $ X2014.Loan : num NA NA NA 10500 NA NA 10500 NA NA NA ...  
## $ X2014.Scholarship : num NA NA NA NA NA NA NA NA NA NA ...  
## $ X2014.Work.Study : int NA NA NA NA NA NA NA NA NA NA ...  
## $ X2014.Grant : num NA NA NA 3146 NA ...  
## $ X2015.Loan : int NA NA NA 5206 NA NA NA NA NA NA ...  
## $ X2015.Scholarship : num NA NA NA NA NA NA NA NA NA NA ...  
## $ X2015.Work.Study : num NA NA NA NA NA NA NA NA NA NA ...  
## $ X2015.Grant : num NA NA NA 4580 NA NA NA NA NA NA ...  
## $ X2016.Loan : num NA NA NA NA NA NA NA NA NA NA ...  
## $ X2016.Scholarship : num NA NA NA NA NA NA NA NA NA NA ...  
## $ X2016.Work.Study : int NA NA NA NA NA NA NA NA NA NA ...  
## $ X2016.Grant : num NA NA NA 691 NA NA NA NA NA NA ...  
## $ X2017.Loan : int NA NA NA 8385 NA NA NA NA NA NA ...  
## $ X2017.Scholarship : num NA NA NA NA NA NA NA NA NA NA ...  
## $ X2017.Work.Study : num NA NA NA NA NA NA NA NA NA NA ...  
## $ X2017.Grant : num NA NA NA 2233 NA ...

str(student\_finance1)

## 'data.frame': 13769 obs. of 7 variables:  
## $ StudentID : int 297957 302040 234532 303486 304316 302808 304266 292741 296902 302171 ...  
## $ Marital.Status : chr "Single" "Single" "Single" "Married" ...  
## $ Adjusted.Gross.Income : int 0 18096 12383 59303 25133 15971 0 0 19709 42557 ...  
## $ Parent.Adjusted.Gross.Income: int 0 0 0 0 0 0 10551 87131 0 0 ...  
## $ Father.s.Highest.Grade.Level: chr "College" "High School" "High School" "High School" ...  
## $ Mother.s.Highest.Grade.Level: chr "High School" "High School" "High School" "Middle School" ...  
## $ Housing : chr "On Campus Housing" "Off Campus" "Off Campus" "Off Campus" ...

# CONCLUSION : NUMBER OF OBSERVATIONS = NUMBER OF VARIABLES =

## FINAL TRAIN DATA SET : It is a train data set which has label as Dropout

# STEP1: displaying first 6 observation of data set   
a = head(final\_traindata)  
a

## StudentID Marital.Status Adjusted.Gross.Income Parent.Adjusted.Gross.Income  
## 1 20932 Married 52555.00 0.0  
## 2 21868 Single 30600.00 0.0  
## 3 21943 Single 27879.00 0.0  
## 4 22163 Single 26794.00 0.0  
## 5 22672 Single 13045.55 28316.5  
## 6 23538 Single 28376.00 0.0  
## Father.s.Highest.Grade.Level Mother.s.Highest.Grade.Level Housing X2011.12  
## 1 Unknown Unknown Off Campus 0  
## 2 High School High School Off Campus 0  
## 3 Unknown High School Off Campus 0  
## 4 Unknown College Off Campus 0  
## 5 Unknown Unknown Off Campus 0  
## 6 College High School Off Campus 0  
## X2012.13 X2013.14 X2014.15 X2015.16 X2016.17 Address1  
## 1 0 0.000 0.000000 0.000 0.000000 87 W 28th St  
## 2 0 0.000 3.635000 3.925 3.966667 217 Seeley Ave  
## 3 0 0.000 0.000000 0.000 0.000000 7012 Cottage Avenue , 201  
## 4 0 3.150 3.350000 3.000 4.000000 6 Seaview Ct Apt 1233  
## 5 0 2.100 0.000000 0.000 1.850000 55 Manor Dr Apt 14P  
## 6 0 3.785 3.766667 0.000 0.000000 8 Elm Ct Apt #1  
## City State Zip RegistrationDate Gender BirthYear BirthMonth Hispanic  
## 1 Bayonne NJ 7307 20140825 2 1971 4 0  
## 2 Kearny NJ 7032 20140507 2 1980 8 0  
## 3 North Bergen NJ 7047 20120426 2 1982 7 1  
## 4 Bayonne NJ 7002 20131121 2 1982 4 0  
## 5 Newark NJ 7106 20130827 1 1969 3 0  
## 6 Bayonne NJ 7002 20140113 2 1981 6 0  
## AmericanIndian Asian Black NativeHawaiian White TwoOrMoreRace HSDip  
## 1 0 0 1 0 0 0 1  
## 2 0 0 0 0 1 0 1  
## 3 0 0 0 0 0 0 1  
## 4 0 0 1 0 0 0 1  
## 5 0 0 1 0 0 0 1  
## 6 0 1 0 0 0 0 1  
## HSGPAUnwtd HighDeg MathPlacement Dropout  
## 1 -1 0 0 1  
## 2 -1 0 0 0  
## 3 -1 0 0 1  
## 4 -1 0 0 0  
## 5 -1 0 0 1  
## 6 -1 2 0 0

# displaying last 6 observation  
b = tail(final\_traindata)  
# dimension of data frame   
dim(final\_traindata)

## [1] 12261 33

# names of columns in final data set  
names(final\_traindata)

## [1] "StudentID" "Marital.Status"   
## [3] "Adjusted.Gross.Income" "Parent.Adjusted.Gross.Income"  
## [5] "Father.s.Highest.Grade.Level" "Mother.s.Highest.Grade.Level"  
## [7] "Housing" "X2011.12"   
## [9] "X2012.13" "X2013.14"   
## [11] "X2014.15" "X2015.16"   
## [13] "X2016.17" "Address1"   
## [15] "City" "State"   
## [17] "Zip" "RegistrationDate"   
## [19] "Gender" "BirthYear"   
## [21] "BirthMonth" "Hispanic"   
## [23] "AmericanIndian" "Asian"   
## [25] "Black" "NativeHawaiian"   
## [27] "White" "TwoOrMoreRace"   
## [29] "HSDip" "HSGPAUnwtd"   
## [31] "HighDeg" "MathPlacement"   
## [33] "Dropout"

# STEP 2: STRUCTURE OF DATA AND DATA TYPES  
str(final\_traindata)

## 'data.frame': 12261 obs. of 33 variables:  
## $ StudentID : int 20932 21868 21943 22163 22672 23538 23548 23606 23897 25893 ...  
## $ Marital.Status : Factor w/ 4 levels "Divorced","Married",..: 2 4 4 4 4 4 4 4 4 2 ...  
## $ Adjusted.Gross.Income : num 52555 30600 27879 26794 13046 ...  
## $ Parent.Adjusted.Gross.Income: num 0 0 0 0 28317 ...  
## $ Father.s.Highest.Grade.Level: Factor w/ 4 levels "College","High School",..: 4 2 4 4 4 1 2 3 2 3 ...  
## $ Mother.s.Highest.Grade.Level: Factor w/ 4 levels "College","High School",..: 4 2 2 1 4 2 2 3 2 3 ...  
## $ Housing : Factor w/ 3 levels "Off Campus","On Campus Housing",..: 1 1 1 1 1 1 1 1 3 2 ...  
## $ X2011.12 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ X2012.13 : num 0 0 0 0 0 0 2.62 0 0 2.21 ...  
## $ X2013.14 : num 0 0 0 3.15 2.1 ...  
## $ X2014.15 : num 0 3.63 0 3.35 0 ...  
## $ X2015.16 : num 0 3.92 0 3 0 ...  
## $ X2016.17 : num 0 3.97 0 4 1.85 ...  
## $ Address1 : Factor w/ 11781 levels "#apt 5","0-55 Morlot Ave",..: 11076 3998 10043 9200 8801 10632 5565 8993 7458 6227 ...  
## $ City : Factor w/ 652 levels "Aberdeen","Akron",..: 27 294 424 27 418 27 27 27 592 294 ...  
## $ State : Factor w/ 39 levels "AL","AZ","CA",..: 24 24 24 24 24 24 24 24 24 24 ...  
## $ Zip : num 7307 7032 7047 7002 7106 ...  
## $ RegistrationDate : num 20140825 20140507 20120426 20131121 20130827 ...  
## $ Gender : num 2 2 2 2 1 2 1 2 1 2 ...  
## $ BirthYear : num 1971 1980 1982 1982 1969 ...  
## $ BirthMonth : num 4 8 7 4 3 6 12 11 5 4 ...  
## $ Hispanic : num 0 0 1 0 0 0 0 1 1 1 ...  
## $ AmericanIndian : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Asian : num 0 0 0 0 0 1 0 0 0 0 ...  
## $ Black : num 1 0 0 1 1 0 0 0 0 0 ...  
## $ NativeHawaiian : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ White : num 0 1 0 0 0 0 1 0 0 0 ...  
## $ TwoOrMoreRace : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ HSDip : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ HSGPAUnwtd : num -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...  
## $ HighDeg : num 0 0 0 0 0 2 0 0 2 0 ...  
## $ MathPlacement : num 0 0 0 0 0 0 0 0 0 1 ...  
## $ Dropout : Factor w/ 2 levels "0","1": 2 1 2 1 2 1 1 2 1 2 ...

## DESCRIPTIVE STATISTICS: FOR ALL DATA SETS

**USING LIBRARY PSYCH TO GET MORE DETAILED DESCRIPTIVE STATISTICS**:

describe () gives standard deviation also

describe(final\_traindata)

## final\_traindata   
##   
## 33 Variables 12261 Observations  
## --------------------------------------------------------------------------------  
## StudentID   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 12261 1 316079 37582 263458 290916   
## .25 .50 .75 .90 .95   
## 305164 321580 343608 352659 356375   
##   
## lowest : 20932 21868 21943 22163 22672, highest: 359313 359320 359327 359554 359783  
## --------------------------------------------------------------------------------  
## Marital.Status   
## n missing distinct   
## 12261 0 4   
##   
## Value Divorced Married Separated Single  
## Frequency 208 924 185 10944  
## Proportion 0.017 0.075 0.015 0.893  
## --------------------------------------------------------------------------------  
## Adjusted.Gross.Income   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 5343 0.944 13230 18164 0 0   
## .25 .50 .75 .90 .95   
## 0 7007 13563 32113 51400   
##   
## lowest : -24326 -7458 -4503 -2732 -1816  
## highest: 281506 330333 514050 1318546 2576425  
##   
## -20000 (1, 0.000), 0 (6724, 0.548), 20000 (4190, 0.342), 40000 (708, 0.058),  
## 60000 (271, 0.022), 80000 (160, 0.013), 1e+05 (91, 0.007), 120000 (51, 0.004),  
## 140000 (29, 0.002), 160000 (15, 0.001), 180000 (6, 0.000), 2e+05 (5, 0.000),  
## 220000 (2, 0.000), 240000 (3, 0.000), 280000 (1, 0.000), 340000 (1, 0.000),  
## 520000 (1, 0.000), 1320000 (1, 0.000), 2580000 (1, 0.000)  
##   
## For the frequency table, variable is rounded to the nearest 20000  
## --------------------------------------------------------------------------------  
## Parent.Adjusted.Gross.Income   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 5466 0.943 28318 36196 0 0   
## .25 .50 .75 .90 .95   
## 0 19982 32980 76235 111077   
##   
## lowest : -49406 -40260 -20471 -15469 -12240, highest: 389700 395099 415463 442652 657631  
## --------------------------------------------------------------------------------  
## Father.s.Highest.Grade.Level   
## n missing distinct   
## 12261 0 4   
##   
## Value College High School Middle School Unknown  
## Frequency 2916 4578 1201 3566  
## Proportion 0.238 0.373 0.098 0.291  
## --------------------------------------------------------------------------------  
## Mother.s.Highest.Grade.Level   
## n missing distinct   
## 12261 0 4   
##   
## Value College High School Middle School Unknown  
## Frequency 2896 4516 1153 3696  
## Proportion 0.236 0.368 0.094 0.301  
## --------------------------------------------------------------------------------  
## Housing   
## n missing distinct   
## 12261 0 3   
##   
## Value Off Campus On Campus Housing With Parent  
## Frequency 6711 1430 4120  
## Proportion 0.547 0.117 0.336  
## --------------------------------------------------------------------------------  
## X2011.12   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 797 0.416 0.4833 0.8328 0.00 0.00   
## .25 .50 .75 .90 .95   
## 0.00 0.00 0.00 2.91 3.48   
##   
## lowest : 0.000000 0.070000 0.125000 0.165000 0.210000  
## highest: 3.980000 3.983333 3.986667 3.990000 4.000000  
## --------------------------------------------------------------------------------  
## X2012.13   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 1010 0.62 0.8236 1.261 0.000 0.000   
## .25 .50 .75 .90 .95   
## 0.000 0.000 1.740 3.425 3.735   
##   
## lowest : 0.000000 0.100000 0.150000 0.250000 0.285000  
## highest: 3.975000 3.976667 3.980000 3.993333 4.000000  
## --------------------------------------------------------------------------------  
## X2013.14   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 1129 0.725 1.04 1.463 0.000 0.000   
## .25 .50 .75 .90 .95   
## 0.000 0.000 2.647 3.560 3.800   
##   
## lowest : 0.000000 0.100000 0.125000 0.165000 0.190000  
## highest: 3.975000 3.976667 3.980000 3.983333 4.000000  
## --------------------------------------------------------------------------------  
## X2014.15   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 1178 0.795 1.223 1.596 0.000 0.000   
## .25 .50 .75 .90 .95   
## 0.000 0.000 2.940 3.647 3.850   
##   
## lowest : 0.000000 0.100000 0.125000 0.150000 0.165000  
## highest: 3.980000 3.983333 3.985000 3.990000 4.000000  
## --------------------------------------------------------------------------------  
## X2015.16   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 1202 0.83 1.315 1.647 0.00 0.00   
## .25 .50 .75 .90 .95   
## 0.00 0.00 3.00 3.69 3.87   
##   
## lowest : 0.000000 0.110000 0.125000 0.165000 0.190000  
## highest: 3.983333 3.985000 3.986667 3.990000 4.000000  
## --------------------------------------------------------------------------------  
## X2016.17   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 1281 0.829 1.283 1.623 0.000 0.000   
## .25 .50 .75 .90 .95   
## 0.000 0.000 2.935 3.665 3.870   
##   
## lowest : 0.000000 0.100000 0.105000 0.115000 0.125000  
## highest: 3.975000 3.976667 3.980000 3.990000 4.000000  
## --------------------------------------------------------------------------------  
## Address1   
## n missing distinct   
## 12261 0 11781   
##   
## lowest : #apt 5 0-55 Morlot Ave 0-69 Elden Pl 00 McAdoo Ave 039 68th Street Apt 4  
## highest: Village Green Apts Weequahic Towers Wesley Towers Westview Towers Whispering Pines   
## --------------------------------------------------------------------------------  
## City   
## n missing distinct   
## 12261 0 652   
##   
## lowest : Aberdeen Akron Albany Albuquerque Allentown   
## highest: Woodbridge Woodbury Woodcliff Lake Woodland Park Wyckoff   
## --------------------------------------------------------------------------------  
## State   
## n missing distinct   
## 12261 0 39   
##   
## lowest : AL AZ CA CO CT, highest: TX UT VA WA WI  
## --------------------------------------------------------------------------------  
## Zip   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 643 0.998 7723 1425 7002 7002   
## .25 .50 .75 .90 .95   
## 7052 7304 7307 8012 8846   
##   
## lowest : 0 747 1202 1436 1541, highest: 95136 95959 97008 98007 98118  
## --------------------------------------------------------------------------------  
## RegistrationDate   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 1032 1 20136172 19385 20110608 20110809   
## .25 .50 .75 .90 .95   
## 20120710 20140122 20150624 20160531 20160718   
##   
## lowest : 20110111 20110113 20110117 20110118 20110119  
## highest: 20160908 20160909 20160910 20160911 20160912  
##   
## 20110000 (60, 0.005), 20110500 (1034, 0.084), 20111000 (834, 0.068), 20120000  
## (243, 0.020), 20120500 (1223, 0.100), 20121000 (590, 0.048), 20130000 (262,  
## 0.021), 20130500 (1100, 0.090), 20131000 (560, 0.046), 20140000 (318, 0.026),  
## 20140500 (1202, 0.098), 20141000 (543, 0.044), 20150000 (324, 0.026), 20150500  
## (1347, 0.110), 20151000 (560, 0.046), 20160000 (190, 0.015), 20160500 (1446,  
## 0.118), 20161000 (425, 0.035)  
##   
## For the frequency table, variable is rounded to the nearest 500  
## --------------------------------------------------------------------------------  
## Gender   
## n missing distinct Info Mean Gmd   
## 12261 0 2 0.722 1.597 0.4814   
##   
## Value 1 2  
## Frequency 4947 7314  
## Proportion 0.403 0.597  
## --------------------------------------------------------------------------------  
## BirthYear   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 56 0.996 1989 8.925 1969 1977   
## .25 .50 .75 .90 .95   
## 1986 1991 1994 1997 1997   
##   
## lowest : 0 1945 1946 1948 1949, highest: 1996 1997 1998 1999 2000  
##   
## Value 0 1945 1950 1955 1960 1965 1970 1975 1980 1985 1990  
## Frequency 1 2 15 65 129 268 337 471 769 1598 3327  
## Proportion 0.000 0.000 0.001 0.005 0.011 0.022 0.027 0.038 0.063 0.130 0.271  
##   
## Value 1995 2000  
## Frequency 4698 581  
## Proportion 0.383 0.047  
##   
## For the frequency table, variable is rounded to the nearest 5  
## --------------------------------------------------------------------------------  
## BirthMonth   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 12 0.993 6.585 3.959 1 2   
## .25 .50 .75 .90 .95   
## 4 7 10 11 12   
##   
## lowest : 1 2 3 4 5, highest: 8 9 10 11 12  
##   
## Value 1 2 3 4 5 6 7 8 9 10 11  
## Frequency 1058 937 997 900 1009 977 1098 1093 1119 1029 1016  
## Proportion 0.086 0.076 0.081 0.073 0.082 0.080 0.090 0.089 0.091 0.084 0.083  
##   
## Value 12  
## Frequency 1028  
## Proportion 0.084  
## --------------------------------------------------------------------------------  
## Hispanic   
## n missing distinct Info Mean Gmd   
## 12261 0 3 0.743 0.2567 0.567   
##   
## Value -1 0 1  
## Frequency 842 7429 3990  
## Proportion 0.069 0.606 0.325  
## --------------------------------------------------------------------------------  
## AmericanIndian   
## n missing distinct Info Mean Gmd   
## 12261 0 3 0.197 -0.0668 0.1317   
##   
## Value -1 0 1  
## Frequency 842 11396 23  
## Proportion 0.069 0.929 0.002  
## --------------------------------------------------------------------------------  
## Asian   
## n missing distinct Info Mean Gmd   
## 12261 0 3 0.4 0.01974 0.2891   
##   
## Value -1 0 1  
## Frequency 842 10335 1084  
## Proportion 0.069 0.843 0.088  
## --------------------------------------------------------------------------------  
## Black   
## n missing distinct Info Mean Gmd   
## 12261 0 3 0.623 0.1467 0.466   
##   
## Value -1 0 1  
## Frequency 842 8778 2641  
## Proportion 0.069 0.716 0.215  
## --------------------------------------------------------------------------------  
## NativeHawaiian   
## n missing distinct Info Mean Gmd   
## 12261 0 3 0.196 -0.06696 0.1313   
##   
## Value -1 0 1  
## Frequency 842 11398 21  
## Proportion 0.069 0.930 0.002  
## --------------------------------------------------------------------------------  
## White   
## n missing distinct Info Mean Gmd   
## 12261 0 3 0.669 0.1824 0.504   
##   
## Value -1 0 1  
## Frequency 842 8341 3078  
## Proportion 0.069 0.680 0.251  
## --------------------------------------------------------------------------------  
## TwoOrMoreRace   
## n missing distinct Info Mean Gmd   
## 12261 0 3 0.236 -0.05122 0.1622   
##   
## Value -1 0 1  
## Frequency 842 11205 214  
## Proportion 0.069 0.914 0.017  
## --------------------------------------------------------------------------------  
## HSDip   
## n missing distinct Info Mean Gmd   
## 12261 0 4 0.082 0.9647 0.101   
##   
## Value -1 1 2 4  
## Frequency 266 11916 69 10  
## Proportion 0.022 0.972 0.006 0.001  
## --------------------------------------------------------------------------------  
## HSGPAUnwtd   
## n missing distinct Info Mean Gmd .05 .10   
## 12261 0 214 0.644 0.1395 1.664 -1.00 -1.00   
## .25 .50 .75 .90 .95   
## -1.00 -1.00 2.38 3.10 3.42   
##   
## lowest : -1.00 0.90 1.36 1.50 1.53, highest: 3.96 3.97 3.98 3.99 4.00  
## --------------------------------------------------------------------------------  
## HighDeg   
## n missing distinct Info Mean Gmd   
## 12261 0 4 0.62 0.5912 0.8467   
##   
## Value 0 2 3 4  
## Frequency 8710 3406 143 2  
## Proportion 0.710 0.278 0.012 0.000  
## --------------------------------------------------------------------------------  
## MathPlacement   
## n missing distinct Info Mean Gmd   
## 12261 0 3 0.705 0.2742 0.514   
##   
## Value -1 0 1  
## Frequency 520 7859 3882  
## Proportion 0.042 0.641 0.317  
## --------------------------------------------------------------------------------  
## Dropout   
## n missing distinct   
## 12261 0 2   
##   
## Value 0 1  
## Frequency 7527 4734  
## Proportion 0.614 0.386  
## --------------------------------------------------------------------------------

# STEP1: DESCRIBE THE DATA : UNIVARIATE ANALYSIS(categorical)

## UNIVARIATE ANALYSIS: (categorical)

## FREQUENCY TABLE

# for cohort categorical :   
  
table(student\_progress$Cohort)

##   
## 2011-12 2012-13 2013-14 2014-15 2015-16 2016-17   
## 13126 12466 10408 9513 7769 4663

# conclusion : Most of number of applicants can be seen in 2011-2012   
  
# FOR MARITAL STATUS  
table(final\_traindata$Marital.Status)

##   
## Divorced Married Separated Single   
## 208 924 185 10944

#   
# conclusion:   
# 1. single is the most FOUND then married   
  
# FOR FATHER AND MOTHER GRADE LEVEL  
table(final\_traindata$Father.s.Highest.Grade.Level)

##   
## College High School Middle School Unknown   
## 2916 4578 1201 3566

table(final\_traindata$Mother.s.Highest.Grade.Level)

##   
## College High School Middle School Unknown   
## 2896 4516 1153 3696

# conclusion : OUT OF TOTAL STUDENTS, , MOST OF THE STUDENTS FATHER and mother HAS GONE TO HIGH SCHOOL and same with mother  
  
  
  
  
# DROP OUT : PRPORTION TABLE:  
table1 = table(final\_traindata$Dropout)  
table1

##   
## 0 1   
## 7527 4734

# number of not drop out is more than student drop out.  
  
#proportion table  
prop\_table = prop.table(table1)  
prop\_table

##   
## 0 1   
## 0.6138977 0.3861023

##   
## CONCLUSION :  
#1. Single has the max count as compared to others marital status.  
#2. Most of the father and mother highest grade level is high school  
#3. Most have off campus housing  
#4. proportion table shows that number of drop out(1) is less then not drop out.

## 

## CORRECTING IMBALANCE:

#See imbalance proportion  
# drop out  
table1 = table(final\_traindata$Dropout)  
table1

##   
## 0 1   
## 7527 4734

#proportion table  
prop\_table = prop.table(table1)  
prop\_table

##   
## 0 1   
## 0.6138977 0.3861023

# CONCLUSION: PROPORTION IS 61% AND 38% SO IMBALANCE CORRECTION IS MUST.  
  
#Imbalance correction using caret  
#Create traning test split  
intrain <- createDataPartition(final\_traindata$Dropout,p=0.75,list = FALSE)  
train1 <- final\_traindata[intrain,]  
test1 <- final\_traindata[-intrain,]

## SOLUTION: USING OVERSAMPLING

#Oversampling  
cvctrl <- trainControl(method = "cv", number=10, sampling="up")  
  
modFit <- caret::train(Dropout ~ ., method='rpart', trControl = cvctrl, data=train1)  
predictions <- predict(modFit, newdata = test1)  
confusionMatrix(predictions,test1$Dropout)  
precision <- posPredValue(predictions, test1$Dropout, positive="0")  
recall <- sensitivity(predictions, test1$Dropout, positive="0")  
F1 <- (2 \* precision \* recall) / (precision + recall)  
  
  
# F1 # 79 %

## 

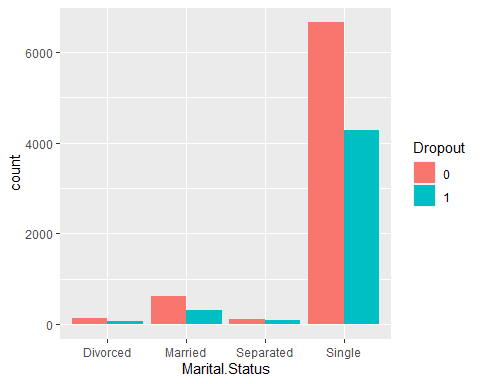
## STEP X1: VISUALISATION;

**BARGRAPH :**

**FOR SINGLE CATEGORICAL VARIABLE DISTRIBUTION Histogram: For single numerical variable Distribution**

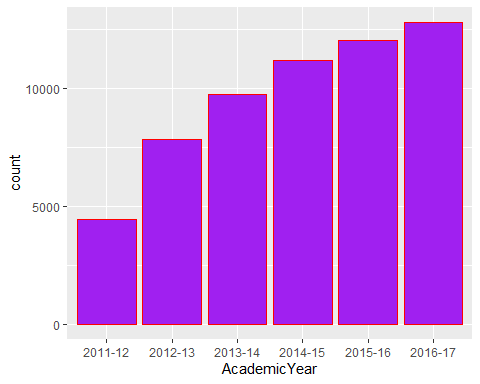
**CODE:**

# for martial status # bar grapgh marital status with color filled by dropout  
# SINGLES ARE MORE NOT TO DROP OUT  
  
ggplot(data= final\_traindata, aes(x= Marital.Status, fill = Dropout)) + geom\_bar(position = "dodge")



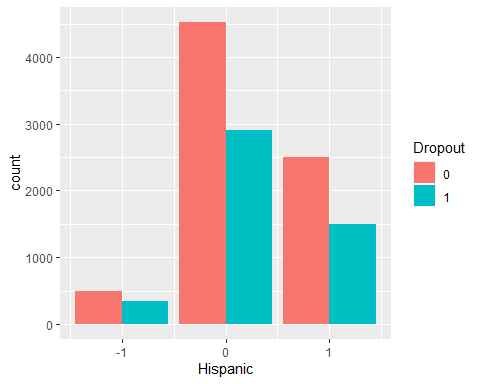
# FOR ACADEMIC YEAR

ggplot(data= student\_progress, aes(x= AcademicYear)) + geom\_bar(color = "Red", fill = "Purple")



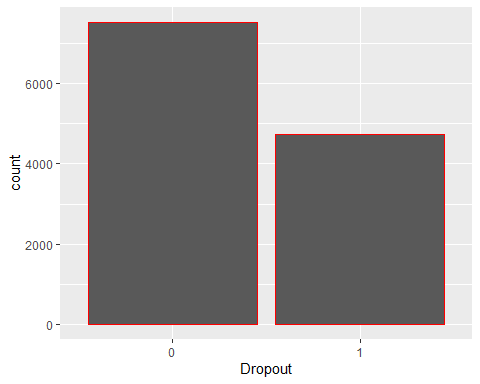
# Hispanic filled by dropouT

ggplot(data= final\_traindata, aes(x= Hispanic , fill = Dropout)) + geom\_bar(position = "dodge")



# for dropout label

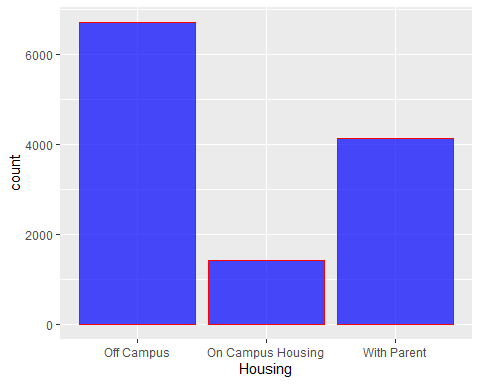
ggplot(data= final\_traindata, aes(x= Dropout)) + geom\_bar(color="red")



# 0 class is very higher then 1

# for housing

ggplot(data= final\_traindata, aes(x= Housing)) + geom\_bar(color="red", fill=rgb(0,0,1,0.7))



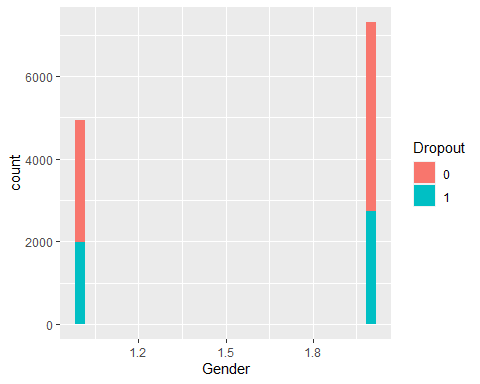
# mos student are off campus then on campus.

**BIVARIATE ANALYSIS FOR CATEGORICAL AND NUMERICAL : USE xtabs(correlation)**

# to see correlation   
# display without na  
xtabs(~ Gender + Dropout, data = final\_traindata) # display without na

## Dropout  
## Gender 0 1  
## 1 2958 1989  
## 2 4569 2745

ggplot(data= final\_traindata, aes(x= Gender, fill = Dropout)) + geom\_histogram()



# total number of females are more than males and number of dropout in female is less   
# IN MALES, ITS ALMOST SAME FOR 0 AND 1, 0 IS LITTLE HIGHER

# box plot:

ggplot(data= final\_traindata, aes(y = Parent.Adjusted.Gross.Income, x = Dropout)) + geom\_boxplot()



# FOR FATHER GRADE LEVEL AND MOTHER GRADE LEVEL

data1 = xtabs(~ Father.s.Highest.Grade.Level + Dropout, data = final\_traindata, addNA = TRUE)  
prop\_table = prop.table(data1)  
prop\_table

## Dropout  
## Father.s.Highest.Grade.Level 0 1  
## College 0.15977490 0.07805236  
## High School 0.23595139 0.13742762  
## Middle School 0.06157736 0.03637550  
## Unknown 0.15659408 0.13424680

# conclusion:   
# conclusion : most student father who went to high school did not drop out and has least drop out number.  
# most drop out happened whose father were college level  
 # parents with high school level shows more 0 (NOT dropout) then dropout.  
 #   
  
# for mother grade level  
data1 = xtabs(~ Mother.s.Highest.Grade.Level + Dropout, data = final\_traindata) # display without na  
xtabs(~ Mother.s.Highest.Grade.Level + Dropout, data = final\_traindata, addNA = TRUE)

## Dropout  
## Mother.s.Highest.Grade.Level 0 1  
## College 1824 1072  
## High School 2827 1689  
## Middle School 698 455  
## Unknown 2178 1518

prop\_table = prop.table(data1)  
prop\_table

## Dropout  
## Mother.s.Highest.Grade.Level 0 1  
## College 0.14876437 0.08743169  
## High School 0.23056847 0.13775385  
## Middle School 0.05692847 0.03710953  
## Unknown 0.17763641 0.12380719

# conclusion :single drop out is highest among all  
# married ones are next which has higher dropout

##STEP 4: NUMERICAL VARIABLE ANALYSIS :

summary(final\_traindata)

## StudentID Marital.Status Adjusted.Gross.Income  
## Min. : 20932 Divorced : 208 Min. : -24326   
## 1st Qu.:305164 Married : 924 1st Qu.: 0   
## Median :321580 Separated: 185 Median : 7007   
## Mean :316079 Single :10944 Mean : 13230   
## 3rd Qu.:343608 3rd Qu.: 13563   
## Max. :359783 Max. :2576425   
##   
## Parent.Adjusted.Gross.Income Father.s.Highest.Grade.Level  
## Min. :-49406 College :2916   
## 1st Qu.: 0 High School :4578   
## Median : 19982 Middle School:1201   
## Mean : 28318 Unknown :3566   
## 3rd Qu.: 32980   
## Max. :657631   
##   
## Mother.s.Highest.Grade.Level Housing X2011.12   
## College :2896 Off Campus :6711 Min. :0.0000   
## High School :4516 On Campus Housing:1430 1st Qu.:0.0000   
## Middle School:1153 With Parent :4120 Median :0.0000   
## Unknown :3696 Mean :0.4833   
## 3rd Qu.:0.0000   
## Max. :4.0000   
##   
## X2012.13 X2013.14 X2014.15 X2015.16   
## Min. :0.0000 Min. :0.000 Min. :0.000 Min. :0.000   
## 1st Qu.:0.0000 1st Qu.:0.000 1st Qu.:0.000 1st Qu.:0.000   
## Median :0.0000 Median :0.000 Median :0.000 Median :0.000   
## Mean :0.8236 Mean :1.040 Mean :1.223 Mean :1.315   
## 3rd Qu.:1.7400 3rd Qu.:2.647 3rd Qu.:2.940 3rd Qu.:3.000   
## Max. :4.0000 Max. :4.000 Max. :4.000 Max. :4.000   
##   
## X2016.17 Address1 City   
## Min. :0.000 1205 7th St : 7 Jersey City :3331   
## 1st Qu.:0.000 10 Kearny Ave : 6 Bayonne :1150   
## Median :0.000 NJCU-Registrar's Office: 6 Newark : 692   
## Mean :1.283 1139 46th St : 5 North Bergen : 564   
## 3rd Qu.:2.935 Summit Apts : 5 Union City : 551   
## Max. :4.000 102 Heller Pkwy : 4 West New York: 418   
## (Other) :12228 (Other) :5555   
## State Zip RegistrationDate Gender   
## NJ :11972 Min. : 0 Min. :20110111 Min. :1.000   
## NY : 120 1st Qu.: 7052 1st Qu.:20120710 1st Qu.:1.000   
## FL : 29 Median : 7304 Median :20140122 Median :2.000   
## CA : 16 Mean : 7723 Mean :20136172 Mean :1.597   
## MD : 15 3rd Qu.: 7307 3rd Qu.:20150624 3rd Qu.:2.000   
## PA : 14 Max. :98118 Max. :20160912 Max. :2.000   
## (Other): 95   
## BirthYear BirthMonth Hispanic AmericanIndian   
## Min. : 0 Min. : 1.000 Min. :-1.0000 Min. :-1.0000   
## 1st Qu.:1986 1st Qu.: 4.000 1st Qu.: 0.0000 1st Qu.: 0.0000   
## Median :1991 Median : 7.000 Median : 0.0000 Median : 0.0000   
## Mean :1989 Mean : 6.585 Mean : 0.2567 Mean :-0.0668   
## 3rd Qu.:1994 3rd Qu.:10.000 3rd Qu.: 1.0000 3rd Qu.: 0.0000   
## Max. :2000 Max. :12.000 Max. : 1.0000 Max. : 1.0000   
##   
## Asian Black NativeHawaiian White   
## Min. :-1.00000 Min. :-1.0000 Min. :-1.00000 Min. :-1.0000   
## 1st Qu.: 0.00000 1st Qu.: 0.0000 1st Qu.: 0.00000 1st Qu.: 0.0000   
## Median : 0.00000 Median : 0.0000 Median : 0.00000 Median : 0.0000   
## Mean : 0.01974 Mean : 0.1467 Mean :-0.06696 Mean : 0.1824   
## 3rd Qu.: 0.00000 3rd Qu.: 0.0000 3rd Qu.: 0.00000 3rd Qu.: 1.0000   
## Max. : 1.00000 Max. : 1.0000 Max. : 1.00000 Max. : 1.0000   
##   
## TwoOrMoreRace HSDip HSGPAUnwtd HighDeg   
## Min. :-1.00000 Min. :-1.0000 Min. :-1.0000 Min. :0.0000   
## 1st Qu.: 0.00000 1st Qu.: 1.0000 1st Qu.:-1.0000 1st Qu.:0.0000   
## Median : 0.00000 Median : 1.0000 Median :-1.0000 Median :0.0000   
## Mean :-0.05122 Mean : 0.9647 Mean : 0.1395 Mean :0.5912   
## 3rd Qu.: 0.00000 3rd Qu.: 1.0000 3rd Qu.: 2.3800 3rd Qu.:2.0000   
## Max. : 1.00000 Max. : 4.0000 Max. : 4.0000 Max. :4.0000   
##   
## MathPlacement Dropout   
## Min. :-1.0000 0:7527   
## 1st Qu.: 0.0000 1:4734   
## Median : 0.0000   
## Mean : 0.2742   
## 3rd Qu.: 1.0000   
## Max. : 1.0000   
##

## STEP 2: UNDERSTANDING OF DATA SET USING DESCRIPTIVE STATISTICS AND VISUALIZATION:

1. Frontier Set cohorts include all undergraduate students who attempted at least one course in a given term, for the first time at your institution. Thus, analysis shows 2011-12 have max students who attempted at least one course in given term and 2016-2017 has minimum number of such student.
2. Among all marital status, most are Single.
3. Singles are less to drop out.
4. Most of the students are living off campus.
5. Total number of females taking admission are more than male
6. Most of father and mother has highest grade level as High School
7. parents with high school grade level are less likely to drop.
8. Number of drop outs are less as compared to not drop out.
9. Academic year; year that applies to the current course file, 2016-17 has maximum student.

## FEATURE ENGINEERING: FEATURE SELECTION:

1.**EMBEDED METHODS** : These methods perform feature selection as part of the model training process. For example, a decision tree inherently performs feature selection because it selects one feature on which to split the tree at each training step.

2. **Using Baruta.**

**CODE:**

#METHODS1 : DECISION TREE FOR FEATURE SELECTION.  
  
#control <- trainControl(method="repeatedcv", number=10, repeats=3)  
#model.fit <- caret::train(Dropout~., data= final\_traindata, method="rpart", preProcess="scale", trControl=control)  
# Estimate variable importance  
  
#importance <- varImp(model.fit, scale=TRUE)  
# summarize importance  
#print(importance)  
# plot importance  
#plot(importance)  
  
#METHOD2  
#Feature SELECTION USING BAKOTA  
#options(warn = -1)  
# Perform Boruta search  
#boruta\_output <- Boruta(Dropout ~ ., data= final\_traindata, doTrace=0)   
# Get significant variables including tentatives  
#boruta\_signif <- getSelectedAttributes(boruta\_output, withTentative = TRUE)  
#print(boruta\_signif)   
# Variable Importance Scores  
#imps <- attStats(boruta\_output)  
#imps2 = imps[imps$decision != 'Rejected', c('meanImp', 'decision')]  
#a = head(imps2[order(-imps2$meanImp), ]) # descending sort  
#a  
# Plot variable importance  
#plot(boruta\_output, cex.axis=.7, las=2, xlab="", main="Variable Importance")

## CONCLUSION ON FEATURE SELECTION :

1. According to Baruta, x2016-17 are most significant variable.
2. Races like american native, hispanic , city, address have less effect on target.
3. First seven which are x2016-17, 2015-16, registration date, x2014-15, student id, birthyear , x2013.14 are significant to target.
4. Decision tree can also be used to see important variables.
5. Will remove city and address1 from data set as they are not so significant.

## PROBLEMS FACED DURING FEATURE SELECTION:

1. Baruta took a lot of time to show important variables.
2. Removed correlation matrix for 33 variable as not possible to look for p value for each variable.

## METHODOLOGY :

Based on our research question and requirement to automate the student drop out , it will be required to build a predictive model to predict if a student will drop out or not. This is because whenever it is required to predict any future events, predictive model is used. This predictive modelling also called as predictive Analytics, is a mathematical process that seeks to predict future events or outcomes by analyzing patterns that are likely to forecast future results. Once data has been collected, we select and train statistical models, using historical data. It uses data to determine the probable future outcome of an event or likelihood of a situation occurring.

In our case, model will predict likely outcome based on the train data set, which we will provide in order to train the model. Train data set contains all feature variables and target variable. After training the model, we will test the model with test data set, which contains all feature variables and no target as target has to be predicted through the model trained through train data set. Then the model will be evaluated by seeing how much the model correctly predicts the outcome by comparing actual target values.

#Algorithm: Supervised Machine Learning Algorithm: Since we already know what the output has to be and that there is some relationship between output or target with other variables, model will use supervised machine learning algorithm. It helps to build model showing relationship between input and output variables, having idea that there is relationship between input and output. It will develop predictive model based on both output and input data.

##Classification Supervised machine learning Algorithm Based on type of outcome, which is numerical or categorical, type of algorithm is decided for the model. Since our target is categorical type(0 /1), we will use classification algorithm for our model. Classification is the process of recognizing , understanding, and grouping ideas and objects into preset categories or sub population. This algorithm classifies our observation or helps to put data observation into correct group based on feature variable. In our case , it will be classified as 0 / 1.

## MODELLING: VARIABLES HAS BEEN SELECTED BY HYPOTHESIS AND BY HELP OF FEATURE SELECTION.

1 Our train data set train1 contains 31 variables(removed city and address as not so significant according to boruta) and test contains 31 variable 2. We are going to fit our model using all these 31 variables 3. we will use cross validation to reduce over fitting issue. 4. Kaggletest with no labels will be used to predict the outcome and f1 score is given by kaggle.

## TYPES OF MODEL USED:

1. CLASSIFICATION TREE
2. BAGGING
3. SVM
4. RANDOM FOREST
5. STACKING
6. LOGISTIC REGRESSION

## PREDICTION MEASURE:

The confusion matrix is a two by two table that contains four outcomes produced by a binary classifier. Various measures, such as error-rate, accuracy, specificity, sensitivity, and precision, are derived from the confusion matrix. Moreover, several advanced measures, such as ROC and precision-recall, are based on them.

## FOR KAGGLE :

1. PREDICTED DIFFERENT OUTCOMES USING KAGGLE TEST DATA SET WITH NO LABELS AS UNKNOWN DATA SET
2. FOR EACH MODEL, THEN COMBINED EACH PREDICTIONS WITH KAGGLE DATA FRAME USING CBIND
3. THEN, MADE OTHER NEW DATA FRAME CONTAING ONLY STUDENT ID AND PREDICTIONS FROM BLENDED DATA FRAME.

# DATA PARTITION:

# SPLITTING MY DATA SET AGAIN BY 75% AND 25%   
  
set.seed(31)  
intrain <- caret::createDataPartition(final\_traindata$Dropout,p=0.75,list = FALSE)  
data.frame(colnames(final\_traindata))

## colnames.final\_traindata.  
## 1 StudentID  
## 2 Marital.Status  
## 3 Adjusted.Gross.Income  
## 4 Parent.Adjusted.Gross.Income  
## 5 Father.s.Highest.Grade.Level  
## 6 Mother.s.Highest.Grade.Level  
## 7 Housing  
## 8 X2011.12  
## 9 X2012.13  
## 10 X2013.14  
## 11 X2014.15  
## 12 X2015.16  
## 13 X2016.17  
## 14 Address1  
## 15 City  
## 16 State  
## 17 Zip  
## 18 RegistrationDate  
## 19 Gender  
## 20 BirthYear  
## 21 BirthMonth  
## 22 Hispanic  
## 23 AmericanIndian  
## 24 Asian  
## 25 Black  
## 26 NativeHawaiian  
## 27 White  
## 28 TwoOrMoreRace  
## 29 HSDip  
## 30 HSGPAUnwtd  
## 31 HighDeg  
## 32 MathPlacement  
## 33 Dropout

train1 <- final\_traindata[intrain, ]  
test1 <- final\_traindata[-intrain,]  
  
# REMOVING CITY AND ADDRESS1 FROM THE DATA SETS as not so important  
train1 =train1[-c(14,15)]  
test1 = test1[-c(14,15)]  
Kaggletest = Kaggletest[-c(14,15)]  
  
# Setup the cross validation. method is the type of cross validation and number is the number of folds  
trctrl <- trainControl(method = "cv", number = 10)  
  
  
# AFTER REMOVING CITY AND ADRESS1 , AS NOT SO SIGNIFICANT ,TOTAL VARIABLES IN TRAIN AND TEST = 31 AND IN KAGGKETEST = 30(WITHOUT LABEL)

## MODELS:

1. **CLASSIFICATION TREE MODEL**

tree\_fit <- caret::train(Dropout ~ ., data = train1, method = "rpart",trControl=trctrl)  
+#To see the tuned complexity parameter   
tree\_fit$bestTune # 0.02377084

## cp  
## 1 0.01520698

#Predict  
predictions\_tree <- predict(tree\_fit, newdata = test1) # 79%  
  
#Performance metrics  
 confusionMatrix(predictions\_tree,test1$Dropout) # 79%

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1602 345  
## 1 279 838  
##   
## Accuracy : 0.7963   
## 95% CI : (0.7816, 0.8105)  
## No Information Rate : 0.6139   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5659   
##   
## Mcnemar's Test P-Value : 0.009266   
##   
## Sensitivity : 0.8517   
## Specificity : 0.7084   
## Pos Pred Value : 0.8228   
## Neg Pred Value : 0.7502   
## Prevalence : 0.6139   
## Detection Rate : 0.5228   
## Detection Prevalence : 0.6354   
## Balanced Accuracy : 0.7800   
##   
## 'Positive' Class : 0   
##

# var imp  
treeImp <- varImp(tree\_fit, scale = TRUE)  
treeImp

## rpart variable importance  
##   
## only 20 most important variables shown (out of 74)  
##   
## Overall  
## X2016.17 100.00  
## RegistrationDate 43.62  
## X2013.14 36.69  
## StudentID 31.50  
## X2014.15 29.81  
## X2015.16 29.65  
## BirthYear 28.20  
## X2012.13 25.30  
## MathPlacement 15.59  
## HSGPAUnwtd 0.00  
## TwoOrMoreRace 0.00  
## StateIL 0.00  
## StateKS 0.00  
## Black 0.00  
## StateFL 0.00  
## Marital.StatusMarried 0.00  
## StateWA 0.00  
## Zip 0.00  
## StateWI 0.00  
## StateDE 0.00

## CONCLUSION ON CLASSIFICATION TREE:

1. ACCURACY ON TRAIN DATA SET IS 79% WHEREAS FOR KAGGLE TEST DATA SET IS 78%.Although the difference in accuracy is less, but there may be chance of over fitting as accuracy on test data set is less than on train data set.
2. MODEL SHOWS THAT ONLY 20 VARIABLE OUT OF 31 ARE IMPORTANT AND AMONG THEM X2016 ACADEMIC YEAR IS MOST important.

# 2: RANDOM FOREST

#Fit the random forest (method = "rf"). Set importance = TRUE to have the variable importance calculated.  
  
forest\_fit <- caret::train(Dropout ~., data = train1, method = "rf",importance = T, trControl=trctrl)  
#To see model details  
  
#To see the the % variance explained  
forest\_fit$finalModel

##   
## Call:  
## randomForest(x = x, y = y, mtry = min(param$mtry, ncol(x)), importance = ..1)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 38  
##   
## OOB estimate of error rate: 11.18%  
## Confusion matrix:  
## 0 1 class.error  
## 0 5309 337 0.05968827  
## 1 691 2860 0.19459307

#Predict  
predictions\_rf <- predict(forest\_fit, newdata = test1)  
  
  
## performance matrics on test1  
confusionMatrix(predictions\_rf,test1$Dropout) # 89%

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1782 230  
## 1 99 953  
##   
## Accuracy : 0.8926   
## 95% CI : (0.8811, 0.9034)  
## No Information Rate : 0.6139   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7687   
##   
## Mcnemar's Test P-Value : 7.659e-13   
##   
## Sensitivity : 0.9474   
## Specificity : 0.8056   
## Pos Pred Value : 0.8857   
## Neg Pred Value : 0.9059   
## Prevalence : 0.6139   
## Detection Rate : 0.5816   
## Detection Prevalence : 0.6567   
## Balanced Accuracy : 0.8765   
##   
## 'Positive' Class : 0   
##

## oredictions on kaggletest  
predictions1 <- predict(forest\_fit, newdata = Kaggletest)  
  
# gave accuracy = 76%  
  
  
  
#To see the importance of the variables  
forestImp <- varImp(forest\_fit)  
forestImp

## rf variable importance  
##   
## only 20 most important variables shown (out of 74)  
##   
## Importance  
## X2016.17 100.000  
## RegistrationDate 68.213  
## X2015.16 67.944  
## X2014.15 36.503  
## StudentID 35.003  
## BirthYear 29.305  
## MathPlacement 24.531  
## X2013.14 24.394  
## Adjusted.Gross.Income 20.357  
## X2012.13 16.957  
## HighDeg 15.744  
## Parent.Adjusted.Gross.Income 13.522  
## HSGPAUnwtd 12.852  
## Mother.s.Highest.Grade.LevelUnknown 11.660  
## Father.s.Highest.Grade.LevelUnknown 10.422  
## HousingWith Parent 9.792  
## Zip 9.576  
## HSDip 8.114  
## Marital.StatusSingle 7.829  
## Marital.StatusMarried 7.600

## CONCLUSION ON RANDOM FOREST:

1. ACCURACY O TRAIN WAS 89% WHILE ON KAGGLE TEST THE ACCURACY IS 76% WHICH SHOWS THAT THERE IS OVERFITTING ISSUE. It can be resolved by taking less variables.
2. The model showed x2016-17 as most important while housing, marital,father grade ,zip are not so important and can be removed to reduce over fitting issue.

## 3: BAGGING

**CODE:**

# FITTING   
  
bag\_fit <- caret::train(Dropout ~., data = train1, method = "treebag",trControl=trctrl)  
bag\_fit

## Bagged CART   
##   
## 9197 samples  
## 30 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 8277, 8276, 8277, 8277, 8278, 8277, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.8786564 0.7399361

# PREDICTION (TEST1)  
prediction\_bagging <- predict(bag\_fit, newdata = test1)  
  
  
# PERFORMANCE MATRIX  
confusionMatrix(prediction\_bagging ,test1$Dropout)# 88 %

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1764 224  
## 1 117 959  
##   
## Accuracy : 0.8887   
## 95% CI : (0.877, 0.8996)  
## No Information Rate : 0.6139   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7612   
##   
## Mcnemar's Test P-Value : 9.455e-09   
##   
## Sensitivity : 0.9378   
## Specificity : 0.8107   
## Pos Pred Value : 0.8873   
## Neg Pred Value : 0.8913   
## Prevalence : 0.6139   
## Detection Rate : 0.5757   
## Detection Prevalence : 0.6488   
## Balanced Accuracy : 0.8742   
##   
## 'Positive' Class : 0   
##

# PREDICTON USING KAGGLE:   
prediction\_bagging <- predict(bag\_fit, newdata = Kaggletest)  
  
# IMPORTANCE OF VARIABLES  
#To see the importance of the variables  
bagImp <- varImp(bag\_fit, scale=TRUE)  
bagImp

## treebag variable importance  
##   
## only 20 most important variables shown (out of 80)  
##   
## Overall  
## RegistrationDate 100.000  
## StudentID 83.648  
## X2016.17 78.681  
## X2015.16 71.535  
## X2014.15 68.787  
## BirthYear 61.157  
## X2012.13 55.811  
## X2013.14 52.218  
## MathPlacement 38.073  
## Adjusted.Gross.Income 35.396  
## X2011.12 25.988  
## Zip 25.775  
## Parent.Adjusted.Gross.Income 21.408  
## BirthMonth 15.920  
## HSGPAUnwtd 15.344  
## HighDeg 7.856  
## Father.s.Highest.Grade.LevelUnknown 6.803  
## Hispanic 5.989  
## Mother.s.Highest.Grade.LevelUnknown 5.544  
## Asian 5.052

## CONCLUSION ON BAGGING:

1. ACCURACY ON TRAIN IS 88% WHILE ACCURACY ON KAGGLETEST IS 85.9 SO THAT MEANS OVERFITTING ISSUE.
2. IMPORTANT VARIABLE IS X2016-17 AND REGISTRATION DATE 3, LEAST IMOORTANT VARIABLES ARE ASIAN, MARITAIAL, PARENT GROSS INCOME, ZIP, FATHERS GRADE LEVL AND SO CAN BE REMOVED TO OVERCOME OVERFITIING ISSUE

## 4: SVM(LINEAR)

#Fit the SVM model to training data  
#svmLinear uses the linear Kernel.   
modSVMFit <- caret::train(Dropout ~ .,method="svmLinear",trControl=trctrl,data=train1, scale=FALSE)  
SVMpredict <- predict(modSVMFit,test1)  
  
confusionMatrix(SVMpredict,test1$Dropout) # ONLY 64 %

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1469 691  
## 1 412 492  
##   
## Accuracy : 0.64   
## 95% CI : (0.6227, 0.657)  
## No Information Rate : 0.6139   
## P-Value [Acc > NIR] : 0.00153   
##   
## Kappa : 0.2059   
##   
## Mcnemar's Test P-Value : < 2e-16   
##   
## Sensitivity : 0.7810   
## Specificity : 0.4159   
## Pos Pred Value : 0.6801   
## Neg Pred Value : 0.5442   
## Prevalence : 0.6139   
## Detection Rate : 0.4794   
## Detection Prevalence : 0.7050   
## Balanced Accuracy : 0.5984   
##   
## 'Positive' Class : 0   
##

## CONCLUSION ON SVM (LINEAR)

1. ACCURACY ON SVM IS 64% WHICH MEANS DISTRIBUTION OF DATA IS NOT LINEAR.

**5: LOGISTIC REGRESSION**

**CODE:**

# logistic regression  
options(warn = -1)  
logistic <- caret::train(Dropout ~ ., data= train1, method = "glm", family="binomial", trControl=trctrl)  
  
summary(logistic)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7191 -0.6314 -0.3114 0.6147 3.5237   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error z value  
## (Intercept) 2.544e+03 8.562e+01 29.714  
## StudentID 5.306e-07 6.890e-07 0.770  
## Marital.StatusMarried 4.253e-02 2.548e-01 0.167  
## Marital.StatusSeparated 4.760e-02 3.073e-01 0.155  
## Marital.StatusSingle 6.175e-01 2.377e-01 2.597  
## Adjusted.Gross.Income -9.508e-07 1.225e-06 -0.776  
## Parent.Adjusted.Gross.Income -1.617e-07 8.116e-07 -0.199  
## `Father.s.Highest.Grade.LevelHigh School` 1.396e-01 8.142e-02 1.715  
## `Father.s.Highest.Grade.LevelMiddle School` 5.751e-02 1.230e-01 0.468  
## Father.s.Highest.Grade.LevelUnknown 4.439e-01 1.019e-01 4.357  
## `Mother.s.Highest.Grade.LevelHigh School` 8.703e-02 8.191e-02 1.063  
## `Mother.s.Highest.Grade.LevelMiddle School` 2.579e-01 1.228e-01 2.100  
## Mother.s.Highest.Grade.LevelUnknown 1.230e-01 1.023e-01 1.202  
## `HousingOn Campus Housing` -1.954e-01 1.088e-01 -1.795  
## `HousingWith Parent` -1.045e-01 7.511e-02 -1.391  
## X2011.12 -7.616e-01 3.719e-02 -20.480  
## X2012.13 -6.352e-01 3.045e-02 -20.856  
## X2013.14 -5.026e-01 2.493e-02 -20.158  
## X2014.15 -2.686e-01 2.144e-02 -12.527  
## X2015.16 -1.607e-01 2.125e-02 -7.564  
## X2016.17 -6.623e-01 2.567e-02 -25.802  
## StateAZ -1.112e+00 2.539e+00 -0.438  
## StateCA -7.873e-01 2.019e+00 -0.390  
## StateCO 1.218e+01 5.354e+02 0.023  
## StateCT 5.117e-01 3.042e+00 0.168  
## StateDC -1.394e+01 5.354e+02 -0.026  
## StateDE -1.439e+01 5.354e+02 -0.027  
## StateFL -8.903e-01 1.948e+00 -0.457  
## StateGA -1.343e+00 2.200e+00 -0.610  
## StateIA 1.142e-01 2.495e+00 0.046  
## StateIL -1.055e+00 2.593e+00 -0.407  
## StateIN -2.333e-01 2.292e+00 -0.102  
## StateKS -1.415e+01 5.354e+02 -0.026  
## StateKY 1.246e+01 5.354e+02 0.023  
## StateLA 1.150e+01 5.354e+02 0.021  
## StateMA 8.258e-02 2.142e+00 0.039  
## StateMD -1.271e+00 2.125e+00 -0.598  
## StateME -1.621e+01 5.354e+02 -0.030  
## StateMI -1.311e-01 2.370e+00 -0.055  
## StateMN -4.177e-01 2.347e+00 -0.178  
## StateMO -1.253e+01 5.354e+02 -0.023  
## StateNC -1.952e+00 2.326e+00 -0.839  
## StateNE -1.072e+01 5.354e+02 -0.020  
## StateNJ -5.190e-01 1.885e+00 -0.275  
## StateNM -2.409e+00 2.352e+00 -1.024  
## StateNV -1.409e+01 3.756e+02 -0.038  
## StateNY -8.935e-01 1.905e+00 -0.469  
## StateOH -6.647e-01 2.548e+00 -0.261  
## StateOK -1.260e+01 5.354e+02 -0.024  
## StatePA -4.812e-01 2.040e+00 -0.236  
## StateRI 9.759e+00 5.354e+02 0.018  
## StateSC 1.539e-01 2.376e+00 0.065  
## StateSD 1.143e+01 5.354e+02 0.021  
## StateTN -1.373e+01 5.354e+02 -0.026  
## StateTX -1.641e+00 2.201e+00 -0.746  
## StateUT -1.173e+01 5.354e+02 -0.022  
## StateVA -4.449e-01 2.366e+00 -0.188  
## StateWA NA NA NA  
## StateWI 3.904e-01 2.175e+00 0.180  
## Zip 1.350e-05 7.875e-06 1.714  
## RegistrationDate -1.240e-04 4.246e-06 -29.201  
## Gender -1.342e-01 5.867e-02 -2.288  
## BirthYear -2.244e-02 4.725e-03 -4.749  
## BirthMonth -6.900e-03 8.292e-03 -0.832  
## Hispanic -9.113e-03 1.212e-01 -0.075  
## AmericanIndian 6.574e-01 5.179e-01 1.269  
## Asian -2.940e-01 1.423e-01 -2.066  
## Black -9.278e-02 1.252e-01 -0.741  
## NativeHawaiian -4.806e-01 7.420e-01 -0.648  
## White -1.184e-01 1.227e-01 -0.965  
## TwoOrMoreRace 3.839e-01 2.310e-01 1.662  
## HSDip -5.264e-01 9.457e-02 -5.567  
## HSGPAUnwtd 1.300e-01 2.272e-02 5.722  
## HighDeg -1.079e-01 3.347e-02 -3.223  
## MathPlacement -1.326e-01 7.294e-02 -1.818  
## Pr(>|z|)   
## (Intercept) < 2e-16 \*\*\*  
## StudentID 0.44127   
## Marital.StatusMarried 0.86745   
## Marital.StatusSeparated 0.87691   
## Marital.StatusSingle 0.00939 \*\*   
## Adjusted.Gross.Income 0.43757   
## Parent.Adjusted.Gross.Income 0.84207   
## `Father.s.Highest.Grade.LevelHigh School` 0.08632 .   
## `Father.s.Highest.Grade.LevelMiddle School` 0.63999   
## Father.s.Highest.Grade.LevelUnknown 1.32e-05 \*\*\*  
## `Mother.s.Highest.Grade.LevelHigh School` 0.28796   
## `Mother.s.Highest.Grade.LevelMiddle School` 0.03570 \*   
## Mother.s.Highest.Grade.LevelUnknown 0.22950   
## `HousingOn Campus Housing` 0.07259 .   
## `HousingWith Parent` 0.16413   
## X2011.12 < 2e-16 \*\*\*  
## X2012.13 < 2e-16 \*\*\*  
## X2013.14 < 2e-16 \*\*\*  
## X2014.15 < 2e-16 \*\*\*  
## X2015.16 3.91e-14 \*\*\*  
## X2016.17 < 2e-16 \*\*\*  
## StateAZ 0.66139   
## StateCA 0.69660   
## StateCO 0.98186   
## StateCT 0.86642   
## StateDC 0.97923   
## StateDE 0.97856   
## StateFL 0.64764   
## StateGA 0.54179   
## StateIA 0.96350   
## StateIL 0.68392   
## StateIN 0.91892   
## StateKS 0.97891   
## StateKY 0.98143   
## StateLA 0.98287   
## StateMA 0.96925   
## StateMD 0.54970   
## StateME 0.97585   
## StateMI 0.95586   
## StateMN 0.85878   
## StateMO 0.98133   
## StateNC 0.40146   
## StateNE 0.98402   
## StateNJ 0.78310   
## StateNM 0.30588   
## StateNV 0.97007   
## StateNY 0.63906   
## StateOH 0.79417   
## StateOK 0.98122   
## StatePA 0.81351   
## StateRI 0.98546   
## StateSC 0.94836   
## StateSD 0.98297   
## StateTN 0.97954   
## StateTX 0.45587   
## StateUT 0.98252   
## StateVA 0.85089   
## StateWA NA   
## StateWI 0.85751   
## Zip 0.08654 .   
## RegistrationDate < 2e-16 \*\*\*  
## Gender 0.02214 \*   
## BirthYear 2.04e-06 \*\*\*  
## BirthMonth 0.40532   
## Hispanic 0.94005   
## AmericanIndian 0.20437   
## Asian 0.03881 \*   
## Black 0.45862   
## NativeHawaiian 0.51714   
## White 0.33475   
## TwoOrMoreRace 0.09648 .   
## HSDip 2.60e-08 \*\*\*  
## HSGPAUnwtd 1.05e-08 \*\*\*  
## HighDeg 0.00127 \*\*   
## MathPlacement 0.06906 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 12268.3 on 9196 degrees of freedom  
## Residual deviance: 7748.3 on 9123 degrees of freedom  
## AIC: 7896.3  
##   
## Number of Fisher Scoring iterations: 12

**PREDICTION: LOGISTIC REGRESSION**

prediction\_reg = predict(logistic, newdata = test1)  
confusionMatrix(prediction\_reg, test1$Dropout)$overall[1]# 81.527

## Accuracy   
## 0.8152742

# prediction using kaggletest data set  
prediction\_reg = predict(logistic, newdata = Kaggletest) # 79.6%

# CONCLUSION ON LOGISTIC REGRESSION:

1. Accuracy for the train data set is 81.52% while accuracy for kaggletest data set is 79.6% which means over fitting issue.
2. summary of logistic regression helps to figure out most significant variable using p value. A significance level of 0.05 indicates a 5% risk of concluding that an association exists when there is no actual association. If the p-value is less than or equal to the significance level, you can conclude that there is a statistically significant association between the response variable and the term.
3. The significant factor to predict our target is: (using summary() and its p value.
4. X2011.12 - x2016-17 are significant to dropout
5. Registration data is significant
6. hsdip,highdeg are also significant.
7. All variables are positively correlated to dropout as for all p value is positive.

## 6: STACKING

**CODE:**

options(warn = -1)  
modelFitRF <- caret::train(Dropout ~ ., data = train1, method = "rf")  
modelFitGBM <- caret::train(Dropout ~ ., data = train1, method = "gbm",verbose=F)  
modelFitKNN <- caret::train(Dropout ~ ., data = train1, method = "knn")  
  
predRF <- predict(modelFitRF,newdata=test1)  
predGBM <- predict(modelFitGBM, newdata = test1)  
predKNN <- predict(modelFitKNN, newdata = test1)  
  
confusionMatrix(predRF, test1$Dropout)$overall[1]

## Accuracy   
## 0.8939295

confusionMatrix(predGBM, test1$Dropout)$overall[1]

## Accuracy   
## 0.8616188

confusionMatrix(predKNN, test1$Dropout)$overall[1]

## Accuracy   
## 0.6357702

predDF <- data.frame(predRF, predGBM,predKNN, Dropout =test1$Dropout, stringsAsFactors = F)  
modelStack <- caret::train(Dropout ~ ., data = predDF, method = "rf")

## note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .

combPred <- predict(modelStack, predDF)  
confusionMatrix(combPred, test1$Dropout)$overall[1] # 89.6

## Accuracy   
## 0.8939295

##########################################################  
# USING KAGGLETEST TO PREDICT:  
  
#predRF <- predict(modelFitRF,newdata=Kaggletest)  
#predGBM <- predict(modelFitGBM, newdata = Kaggletest)  
#predKNN <- predict(modelFitKNN, newdata = Kaggletest)  
#predDF <- data.frame(predRF, predGBM,predKNN, stringsAsFactors = F)  
#modelStack <- caret::train(predRF ~ ., data = predDF, method = "rf")  
#combPred2 <- predict(modelStack, Kaggletest)  
#combPred2 ## 85%

##CONCLUSION FOR STACKING: 1. It has accuracy of 89% on train while it has accuracy of 85% on kaggle test so showing over fitting issues.

## MODEL SELECTION

1. Bagging with 31 variables has accuracy score of 85.9% and is the best model among all.
2. But since the model has over fitting issue, so will resolve this problem by removing some unimportant variables.

## OVERFITTING:

1. All Models showed over fitting issues as their accuracy on kaggle test are lower than train data set.
2. Removing some variables can help to reduce over fitting.

# OTHER ISSUES:

1. Computer got slow while runnig models especially stacking and random forest.
2. Combining files together from each folder using rbind was very time taking and required repetative codes to be written for all files and so used plyr library to use function lpdy to combine all files into one data frame.
3. While doing feature engineering, Boruta took a lot of time while correlation matrix with such large number of variables was difficult to work.
4. Cleaning of data took a lot of time.

## CONCLUSION:

1. Our bagging model with 31 variable which predicts the student dropout with accuracy score of 85% on new data set implies that we have identified 85% of student drop out accurately.
2. With the help of model, we were able to find the most significant factors that can help in predicting model.
3. Regarding the methodology, data wrangling and feature analysis plays a crucial role in model selection.
4. Feature selection helped to let us know that city and address are not so significant so removed them from train, test and kaggle data set.