STUDENT DROPOUT PREDICTION

FINC 530 Machine Learning for Business 1

Professor: J.D Jayaraman

Student: Thi Diem My Nguyen

Outline

1. Introduction
2. Data set and features
3. Exploratory Data Analysis - EDA
4. Data cleaning
5. Methodology and Results
6. Conclusion

1. Introduction

- Dropping out of colleges is considered not just a serious educational problem but also a severe social problem. This project will aim to accurately predict the probability of a student dropping out from the college.
- Measure prediction accuracy and analyze aspects of the students' data so as to recognize the most important factors leading to high dropout rates. Implement several classification algorithms to find the best prediction model

2. Data set and features

+ The data was gathered from New Jersey City University undergraduate student from 2012 to 2017.

+ The data set contains three types of data:

- Student Static Data: contains one record per student
- Student Progress Data: reflecting each student's activity for each term in each academic year
- Student Financial Aid Data: each student for each academic year

2. Data set and features

Import and merge student static data: using rbind

```
StudentStaticData <-
rbind(StaticFall2011,StaticFall2012,StaticFall2013,StaticFall2014,StaticFall2015,StaticFall2016,Static
Spring2012,StaticSpring2013,StaticSpring2014,StaticSpring2015,StaticSpring2016)
```

2. Data set and features

Import and merge student progress data: using rbind and group by

+ Import

+ Create a new column AcademicYearID

```
ProgressFall2011 <- mutate(ProgressFall2011, AcademicYearID = 1)
ProgressSpring2012 <- mutate(ProgressSpring2012, AcademicYearID = 2)
```

...

```
ProgressSpring2017 <- mutate(ProgressSpring2017, AcademicYearID = 17)
ProgressSum2017 <- mutate(ProgressSum2017, AcademicYearID = 18)
```

+ rbin

+ groupby by StudentID with the lastest AcademicYearID

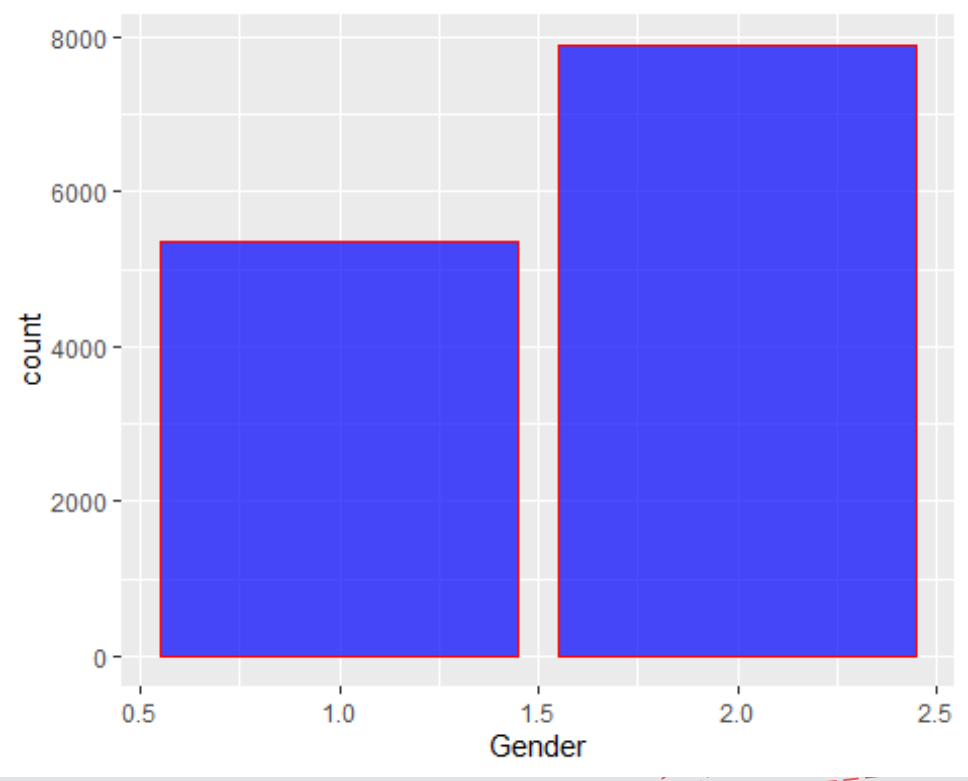
```
ProgressData <- StudentProgressData1 %>% group_by(StudentID) %>% top_n(1, AcademicYearID)
```

3. Exploratory Data Analysis - EDA

+ Student Static Data

summary(StudentStaticData)

Distribution of Gender, most students were female



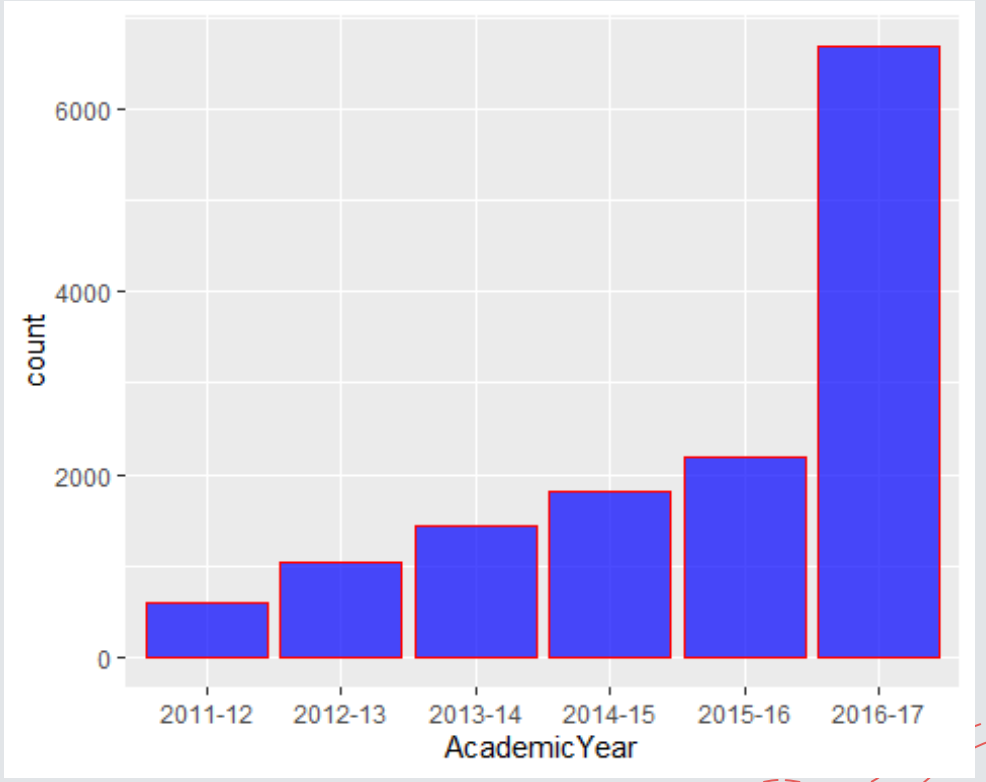
3. Exploratory Data Analysis - EDA

+ Student Progress Data

summary(StudentProgressData)

Distribution of Academic Year

most students were in the year 2016-2017



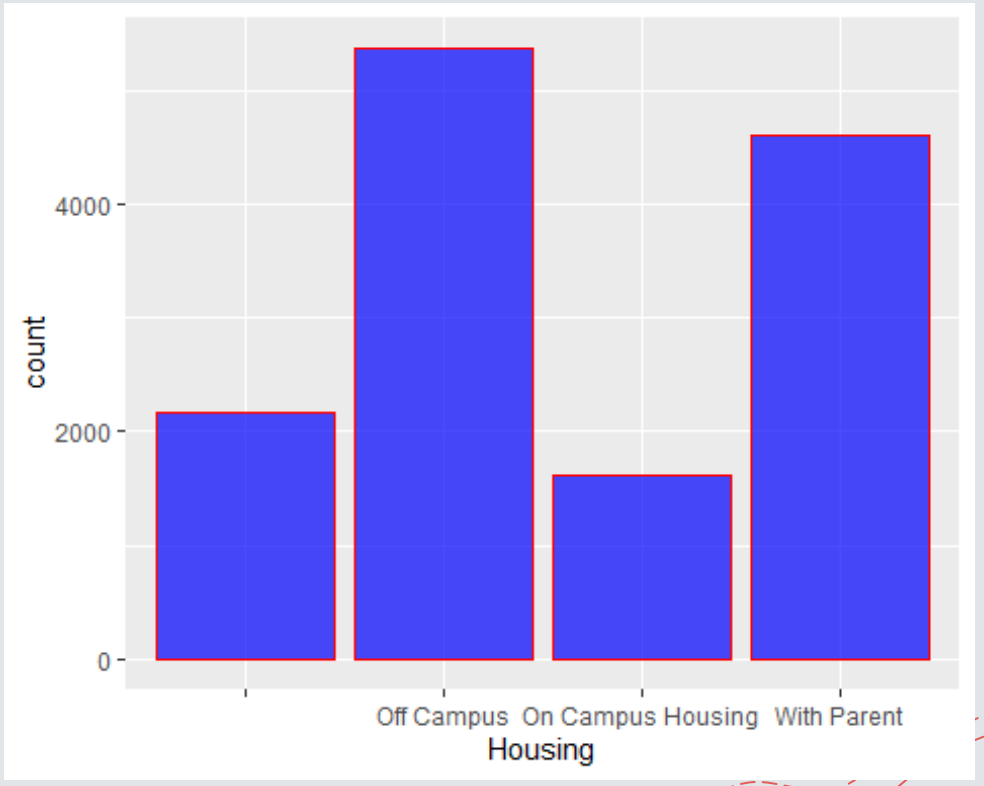
3. Exploratory Data Analysis - EDA

+ **Student Financial Data**

summary(FinancialAid)

Distribution of Housing

most students were living out of campus



4. Data cleaning

+ **Data cleaning for Student Static Data**

+ ```
  #Remove address columns because we won't use them for training and testing data: Address1, Address 2,  City,  State,  Zip,  RegistrationDate
  #Remove columns because of missing most values: Campus, HSDipYr HSGPAWtd, FirstGen, DualHSSummerEnroll, CumLoanAtEntry,
  StudentStatic <- StudentStaticData[-c(4, 5, 6, 7, 8, 9, 10, 22, 24, 25, 26, 30)]
  #Replace value = -1 in Hispanic,  AmericanIndian, Asian,  Black,  NativeHawaiian, White,  TwoOrMoreRace = 0
  StudentStatic["Hispanic"][StudentStatic["Hispanic"] == -1] <- 0
  StudentStatic["AmericanIndian"][StudentStatic["AmericanIndian"] == -1] <- 0
  StudentStatic["Asian"][StudentStatic["Asian"] == -1] <- 0
  StudentStatic["Black"][StudentStatic["Black"] == -1] <- 0
  StudentStatic["NativeHawaiian"][StudentStatic["NativeHawaiian"] == -1] <- 0
  StudentStatic["White"][StudentStatic["White"] == -1] <- 0
  StudentStatic["TwoOrMoreRace"][StudentStatic["TwoOrMoreRace"] == -1] <- 0
  ```
  ```

4. Data cleaning

+ **Data cleaning for Financial Aid Data**

+ ```
  # Most of students are single, so fill the empty values of Marital Status column with Single.
  FinancialAid["MaritalStatus"][FinancialAid["MaritalStatus"] == ""] <- "Single"
  # Most of students live Off campus, so fill the empty values of Housing column with Off Campus.
  FinancialAid["Housing"][FinancialAid["Housing"] == ""] <- "Off Campus"
  # Fill the empty values of parent's Highest Grade level with 'Unknown'.
  FinancialAid["FathersHighestGradeLevel"][FinancialAid["FathersHighestGradeLevel"] == ""] <- "U
  nknown"
  FinancialAid["MotherHighestGradeLevel"][FinancialAid["MotherHighestGradeLevel"] == ""] <- "U
  nknown"
  # Replace all other missing values by 0
  FinancialAid <- na_replace(FinancialAid, 0)StudentStatic["TwoOrMoreRace"][StudentStatic["Two
  OrMoreRace"] == -1] <- 0
  ```
  ```

5. Methodology and results

+ The data set was split in 75% train and 25% test, training the models using grid search and cross-validation on the training set and evaluating them on the test set.

```
+ library(caret)
  intrain <- createDataPartition(DataTrain$Dropout,p=0.75,list = FALSE)

+ train1 <- DataTrain[intrain,]

+ test1 <- DataTrain[-intrain,]

+ #Create cross validation
  trctrl <- trainControl(method = "cv", number = 5)
```

5. Methodology and results

+ **Fit the classification tree model**

+ ```
  model1 <- train(Dropout ~., data = train1, method = "rpart", trControl=trctrl)
  predictions1 <- predict(model1, newdata = test1)
  confusionMatrix(predictions1,test1$Dropout)
  ```

+ ```
  ## Confusion Matrix and Statistics
  ##
  ##          Reference
  ## Prediction    0    1
  ##          0 1845   93
  ##          1   36 1090
  ##
  ##                Accuracy : 0.9579
  ##                  95% CI : (0.9502, 0.9647)
  ##     No Information Rate : 0.6139
  ##     P-Value [Acc > NIR] : < 2.2e-16
  ##
  ```

5. Methodology and results

+ **Fit the Logistic Regression Model**

+ ```
  model2 <- train(Dropout ~., data = train1, method = "glm", trControl=trctrl)
  predictions2 <- predict(model2, newdata = test1)
  confusionMatrix(predictions2,test1$Dropout)
  ```

+ ```
  ## Confusion Matrix and Statistics
  ##
  ##          Reference
  ## Prediction    0    1
  ##          0 1832   90
  ##          1   49 1093
  ##
  ##                Accuracy : 0.9546
  ##                  95% CI : (0.9467, 0.9617)
  ##     No Information Rate : 0.6139
  ##     P-Value [Acc > NIR] : < 2.2e-16
  ##
  ```

5. Methodology and results

Fit the Bagging Model

```
model3 <- train(Dropout ~., data = train1, method = "treebag", trControl=trctrl)
predictions3 <- predict(model3, newdata = test1)
confusionMatrix(predictions3,test1$Dropout)
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    0    1
##          0 1846   78
##          1   35 1105
##
##                Accuracy : 0.9631
##                  95% CI : (0.9558, 0.9695)
##     No Information Rate : 0.6139
##     P-Value [Acc > NIR] : < 2.2e-16
##
```

5. Methodology and results

+ **Fit the SVM Radial Model**

+ ```
  model4 <- train(Dropout ~., data = train1, method = "svmRadial", trControl=trctrl)
  predictions4 <- predict(model4, newdata = test1)
  confusionMatrix(predictions4,test1$Dropout)
  ```

+ ```
  ## Confusion Matrix and Statistics
  ##
  ##          Reference
  ## Prediction    0    1
  ##          0 1860  120
  ##          1   21 1063
  ##
  ##                Accuracy : 0.954
  ##                  95% CI : (0.946, 0.9611)
  ##     No Information Rate : 0.6139
  ##     P-Value [Acc > NIR] : < 2.2e-16
  ##
  ```

5. Methodology and results

Stacking using Random Forest

```
# Construct data frame with predictions
library(caret)
predDF <- data.frame(predictions1, predictions2, predictions3, predictions4, class = test1$D
ropout)
predDF$class <- as.factor(predDF$class)
#Combine models using random forest
combModFit.rf <- train(class ~ ., method = "rf", data = predDF, distribution = 'multinomial')
combPred.rf <- predict(combModFit.rf, predDF)
confusionMatrix(combPred.rf, predDF$class)$overall[1]
```

```
##  Accuracy
## 0.9631201
```

5. Methodology and results

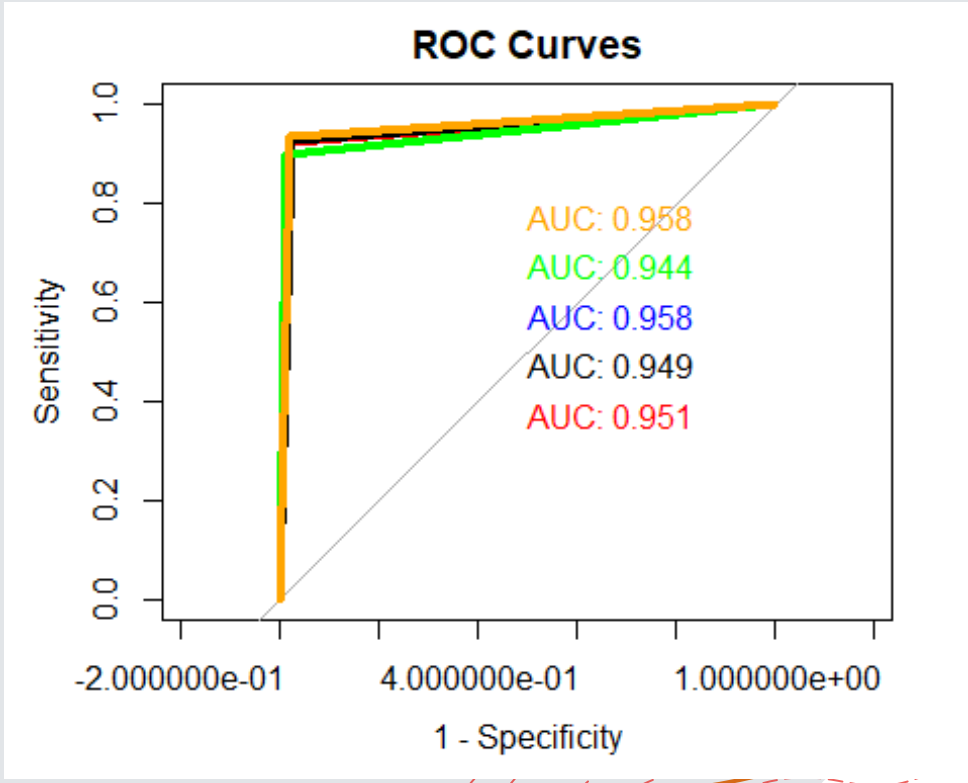
- **Compare the accuracy of each model**
- The performance of the classifiers is assessed using the standard measure of accuracy.
- Model Accuracy Score
  - Classification tree 95.79%
  - Logistic Regression 95.46%
  - Bagging 96.31%
  - SVM Radial 95.54%
  - Stacking with Random Forest 96.31%

5. Methodology and results

ROC Curve

The plot presents the ROC curves for the fine binary classifiers used.

The the bagging model, and Stacking model using Random Forest performed the same AUC and better than the Classification Tree, Logistic Regression and SVM.



6. Conclusion

These models achieves high predictive power with accuracy score over 96%.

The result was that the Bagging and Stacking with Random Forest performed the best, followed by the Classification Tree, Logistic Regression and SVM.

Some improvements that can be made to the experiment include a more advanced solution dealing with missing values rather than replacing missing values to 0 or the majority value.

Demands analytics and coding skills.

Issues

- Data wrangling: spent more than 80% time.
- Storage memories:

+ Increase memory size:

memory.size()

memory.limit()

memory.limit(size=500000)

+ Remove unused data

Rm() and gc()

- Taking time to run the code:

Using R Markdown instead of R Studio

Question and Answer