Phase Two: Main Analysis

Mohamed Elmanzalawi

## Introduction

## Previously, we have examined the MidField data and gained an understanding of student demographics. In this report we will build on our previous findings and identify the predictors most signifiant to graduation.

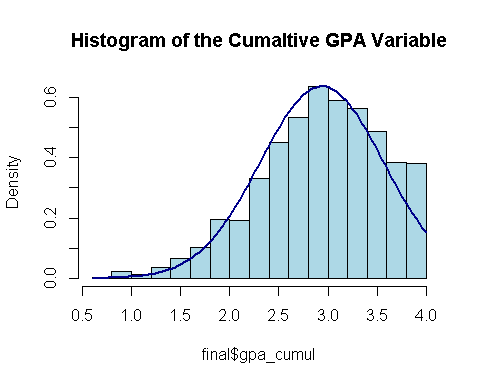
## Domain Insights

### For decades, reserchers have studyed the attirbutes that effect student educational performnce and graduation rate in college. In this report, we leverage such reserch and associted domain knowldge in the feature selection processes.

### The Higher Education Institue at the college of UCLA reports in its study,“Completing College: Assessing Graduation Rates at Four-Year Institutions”, that the most impactful features to gradation include: Student Age, Gender, Ethnicity, Commitment to their institution, and Grade Point Average. This gives a solid foundation of attributes to explore in the MidlField Data.

## Methods

**The goal is to identify the most significant student characteristics with decision trees and ensemble methods. Decision trees are unique because they have an intuitive structure and have very interpretable results. Decision trees are also great for data that does not follow a normal distribution.**



### In the MidField data, the Cumulative GPA variable has a distinctive left skew, indicating that it does not follow the normal distribution. Applying decision tree algorithms to our data should enable higher predictive ability than Logistic regression.

## Data Preprocessing

### A vital aspect of working with decision trees is properly stratifying training and testing data. Ensuring that training data has all levels or feature values for categorical variables improves our model’s generalizability to the testing data.

Relative Frequency of Race & Sex Across Train/Test Data

| **race** | **sex** | **race\_freq\_train** | **sex\_freq\_train.x** | **race\_freq\_test** | **sex\_freq\_train.y** |
| --- | --- | --- | --- | --- | --- |
| Asian | Female | 0.4647059 | 0.042781600 | 0.4642857 | 0.042964281 |
| Asian | Male | 0.5352941 | 0.045128355 | 0.5357143 | 0.045391061 |
| Black | Female | 0.4482759 | 0.018221203 | 0.4489164 | 0.018431422 |
| Black | Male | 0.5517241 | 0.020536756 | 0.5510836 | 0.020716946 |
| International | Female | 0.4236551 | 0.068233945 | 0.4244094 | 0.068514046 |
| International | Male | 0.5763449 | 0.085005834 | 0.5755906 | 0.085079143 |
| Latine | Female | 0.5001364 | 0.058422528 | 0.4994595 | 0.058726325 |
| Latine | Male | 0.4998636 | 0.053471412 | 0.5005405 | 0.053887337 |
| Native American | Female | 0.5350554 | 0.004619011 | 0.5000000 | 0.004448964 |
| Native American | Male | 0.4649446 | 0.003675613 | 0.5000000 | 0.004073557 |
| Other/Unknown | Female | 0.4723404 | 0.045967125 | 0.4721141 | 0.046269226 |
| Other/Unknown | Male | 0.5276596 | 0.047024504 | 0.5278859 | 0.047369646 |
| White | Female | 0.4835109 | 0.761754587 | 0.4836728 | 0.760645735 |
| White | Male | 0.5164891 | 0.745157526 | 0.5163272 | 0.743482309 |
|  |  |  |  |  |  |

### The frequency table above displays the result of our stratification for race and sex, sensitive variables. Despite having an 80/20 training testing ratio, the frequency of each feature combination remains consistent. It’s important to note that our training and testing splits stratify across all categorical features, not just sensitive attributes.

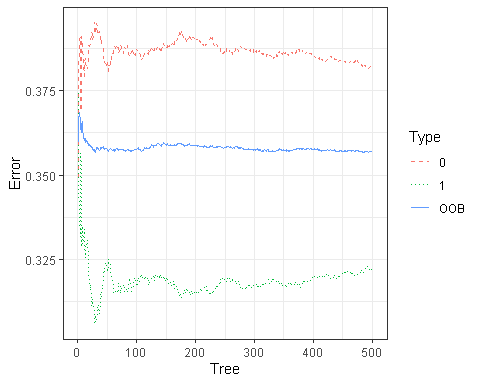
### Despite the unique decision tree characteristics mentioned previously, they have a profound flaw: the performance of a single decision tree yields poor results. However, combining decision trees in an ensemble significantly improves performance.

## Creating an Ensmable

### There are many approaches to creating ensembles. The most robust is known as the Random Forest. This algorithm modifies the bagging ensemble method to generate training splits with random subsets of training features. This diversity in training data diversifies the decision trees while retaining a sense of feature importance.

## Out-of-Bag Sample Validation

### Out-of-bag error provides us with an evaluation of our model from training data left out of our many training samples. The out-of-bag error is somewhat similar to utilizing cross-validation to gauge our models performance.



### The graph above shows that the out-of-bag error stabilizes after approximately 100 trees. The total out-of-bag error rate is 35%, the error classifying graduates is 38%, and the non-graduates is 31%. While these results are higher than expected, they reveal an intriguing insight into our model’s strengths and weaknesses. The 7% difference in error between classifying graduates and non-graduates indicates that it is much easier to predict graduates accurately.

## Predicting & Evaluation

Random Forest Confusion Matrix

| **Prediction** | **Truth** | **Freq** |
| --- | --- | --- |
| 0 | 0 | 5,985 |
| 1 | 0 | 3,568 |
| 0 | 1 | 2,251 |
| 1 | 1 | 4,655 |

## Additional Metrics

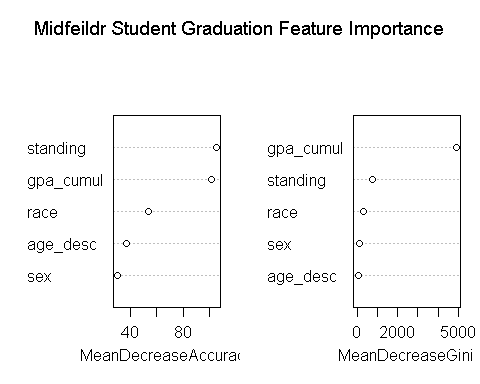
Random Forest Model Accuracy, Sensitivity & Specificity

| **Metric** | **Value** |
| --- | --- |
| Accuracy | 0.6464548 |
| Sensitivity | 0.6265048 |
| Specificity | 0.6740515 |

### The confusion matrix provides us with a general understanding of our model. However, additional metrics such as Accuracy, Sensitivity, and Specificity give us a more complete view of performance. Accuracy is approximately 65%. This metric will be important when compared to the logistic regression model.

Feature Importance Graduation Forest

| **Attributes** | **Non-Graduates** | **Graduates** | **MeanDecreaseAccuracy** | **MeanDecreaseGini** |
| --- | --- | --- | --- | --- |
| standing | 10.110622 | 3.107429 | 105.29840 | 777.80556 |
| gpa\_cumul | 50.143506 | 230.703835 | 101.49473 | 4,920.37664 |
| age\_desc | 22.989483 | 29.657613 | 37.01589 | 84.88823 |
| race | 24.264990 | 49.539976 | 53.31429 | 320.77183 |
| sex | -4.161167 | 31.154316 | 29.89102 | 118.64724 |



### These results indicate that the most significant features in classifying graduation are Academic Standing, GPA, Ethnicity, Age, and Sex. Let’s examine if the results are consistent with the logistic regression model.

## Random Forest vs. Logistic Regression

## Confusion Matrix

Logistic Regression Confusion Matrix

| **Prediction** | **Truth** | **Freq** |
| --- | --- | --- |
| 0 | 0 | 5,984 |
| 1 | 0 | 3,569 |
| 0 | 1 | 2,251 |
| 1 | 1 | 4,655 |
|  |  |  |

Logistic Regression Accuracy, Sensitivity & Specificity

| **Metric** | **Value** |
| --- | --- |
| Accuracy | 0.6463941 |
| Sensitivity | 0.6264001 |
| Specificity | 0.6740515 |

### The confusion matrix and additional metrics appear to be similar to the Random Forest model.

## Feature Importance

Feature Importance Graduation Logistic Regression

| **term** | **estimate** |
| --- | --- |
| (Intercept) | -7.9103611 |
| standingAcademic Probation | -0.4879704 |
| standingAcademic Warning | 7.0659343 |
| standingGood Standing | 5.5696637 |
| standingNo Credit Courses Attempted | 3.8896348 |
| gpa\_cumul | 0.5109671 |
| age\_descUnder 25 | 0.6267506 |
| raceBlack | -0.2798053 |
| raceInternational | 0.1277666 |
| raceLatine | -0.3192414 |

### These results indicate that the most significant features in classifying graduation are also Academic Standing, GPA, Ethnicity, Age, and Sex.

## Final Recommendation

### Multiple models have determined that Academic Standing, GPA, Ethnicity, Age, and Sex impact the probability of a student completing a degree. While the logistic regression provides more detailed feature values, both models are consistent with the findings of current research

# Code Appendix

# Setting Document Options  
knitr::opts\_chunk$set(  
 echo = FALSE,  
 warning = FALSE,  
 message = FALSE,  
 fig.align = "center",  
 cache = FALSE   
)  
#install.packages(pkgs = c("tidyverse", "pROC", "broom", "knitr", "flextable", "tree", "rpart", "rpart.plot", "partykit","rattle", "randomForest", "yardstick", "splitTools","randomForestExplainer", "partykit", "DiagrammeR"))  
# Libraries & Imports  
packages <- c("tidyverse", "pROC", "broom", "knitr", "flextable",  
 "tree", "rpart", "rpart.plot", "partykit",  
 "rattle", "randomForest", "yardstick", "splitTools","randomForestExplainer", "partykit", "DiagrammeR")  
  
invisible(  
 lapply(   
 X = packages,  
 FUN = library,  
 character.only = TRUE,  
 quietly = TRUE  
 )  
)  
# Load project 2 data   
final <- read.csv("final\_data.csv")  
# Generate a histogram  
hist(final$gpa\_cumul, probability = TRUE, col = "lightblue", main = "Histogram of the Cumaltive GPA Variable")  
  
# Add a normal distribution curve  
curve(dnorm(x, mean = mean(final$gpa\_cumul), sd = sd(final$gpa\_cumul)), add = TRUE, col = "darkblue", lwd = 2)  
  
# Setting seed for reproducibility  
set.seed(42)  
  
# Stratifying over a variable of combined factor levels  
final$combined\_levels <- interaction(final$standing, final$age\_desc, final$race, final$sex, final$graduation)  
  
index <- partition(final$combined\_levels, p = c(train = 0.8, valid = 0.2), split\_into\_list = T)  
  
# Splitting the data  
train\_data <- final[index$train, ]  
test\_data <- final[index$valid, ]  
# Removing single training outlier  
train\_data <- train\_data %>%  
 filter(!(race == "Other/Unknown" & sex == "Unknown"))  
# Create frequency tables   
sensitive\_freq\_train <- train\_data %>%  
 group\_by(race, sex) %>%  
 summarise(train\_count = n(), .groups = "keep")   
  
# Add relative frequency columns for both race and sex  
sensitive\_freq\_train <- sensitive\_freq\_train %>%  
 group\_by(race) %>%  
 mutate(race\_freq\_train = train\_count / sum(train\_count)) %>%  
 group\_by(sex) %>%  
 mutate(sex\_freq\_train = train\_count / sum(train\_count))  
  
sensitive\_freq\_test <- test\_data%>%  
 group\_by(race, sex) %>%  
 summarise(test\_count = n(), .groups = "keep")  
  
# Add relative frequency columns for both race and sex  
sensitive\_freq\_test <- sensitive\_freq\_test %>%  
 group\_by(race) %>%  
 mutate(race\_freq\_test = test\_count / sum(test\_count)) %>%  
 group\_by(sex) %>%  
 mutate(sex\_freq\_train = test\_count / sum(test\_count))  
  
combined\_data <-left\_join(sensitive\_freq\_train, sensitive\_freq\_test, by = c("race","sex"))  
# Professional version of combined frequency table  
combined\_data$train\_count <- NULL  
combined\_data$test\_count <- NULL  
  
combined\_data %>%  
 flextable() %>%  
 set\_caption("Relative Frequency of Race & Sex Across Train/Test Data") %>%  
 set\_table\_properties(width = .6, layout = "autofit") %>%  
 autofit() %>%  
 theme\_zebra() %>%  
 flextable::set\_header\_labels(bold = TRUE)  
  
# show that relative frequency of sensitive attributes are same between training and testing  
# Save and remove the graduation/ combined\_levels columns   
y\_true <- test\_data$graduation  
  
train\_data$graduation <- as.factor(train\_data$graduation)  
test\_data$graduation <- NULL  
  
test\_data$combined\_levels <- NULL  
train\_data$combined\_levels <- NULL  
  
# Using randomForest as an ensemble method  
grad\_rf <- randomForest(  
 formula = graduation ~ standing + gpa\_cumul + age\_desc +  
 race + sex,  
 data = train\_data,  
 ntree = 500,  
 mtry = 3,  
 importance = TRUE,  
 do.trace = FALSE,   
 keep.forest = TRUE  
)  
# save out of bag error rate  
rf\_oob <- as.data.frame(grad\_rf$err.rate)  
  
# Create a line plot of OOB Error and Misclassification Rates  
rf\_oob %>%  
 mutate(  
 Tree = row\_number(),  
 .before = OOB  
 ) %>%  
 pivot\_longer(  
 cols = !Tree,  
 names\_to = "Type",  
 values\_to = "Error"  
 ) %>%  
 ggplot(  
 mapping = aes(  
 x = Tree,  
 y = Error,  
 color = Type,  
 linetype = Type  
 )  
) +  
 geom\_path() +  
 theme\_bw() +  
 scale\_linetype\_manual(values = c("dashed", "dotted", "solid"))  
  
# Predict test data  
test\_data1 <- test\_data  
test\_data1$y\_pred\_rf <- predict(  
 object = grad\_rf,  
 newdata = test\_data,  
 type = "response"  
)  
  
y\_true <- as.factor(y\_true)  
test\_data1$y\_true <- y\_true  
test\_data1$y\_pred\_rf <- as.factor(test\_data1$y\_pred\_rf)  
# confusion matrix data  
rf\_data <- conf\_mat(  
 data = test\_data1,  
 truth = y\_true,  
 estimate = y\_pred\_rf  
)$table  
  
rf\_confusion\_mat <- as.data.frame(rf\_data)  
  
flextable(rf\_confusion\_mat) %>%  
set\_caption("Random Forest Confusion Matrix") %>%  
set\_table\_properties(width = .5) %>%  
theme\_zebra() %>%  
set\_table\_properties(layout = "autofit")  
# Calculate accuracy, sensitivity, and specificity  
accuracy\_val <- accuracy(rf\_data)  
sensitivity\_val <- sens(rf\_data)  
specificity\_val <- spec(rf\_data)  
  
# Create a data frame for the results  
result\_table <- data.frame(  
 Metric = c("Accuracy", "Sensitivity", "Specificity"),  
 Value = c(accuracy\_val$.estimate, sensitivity\_val$.estimate, specificity\_val$.estimate),  
 stringsAsFactors = FALSE  
)  
  
flextable(result\_table) %>%  
 set\_caption("Random Forest Model Accuracy, Sensitivity & Specificity ") %>%  
 set\_table\_properties(width = .5) %>%  
 theme\_zebra() %>%  
 set\_table\_properties(layout = "autofit")  
  
# Display attribute importance  
importance\_df <- as.data.frame(importance(grad\_rf))  
  
importance\_df <- importance\_df %>%  
 rename("Non-Graduates" = "0", "Graduates" = "1")  
  
importance\_df <- importance\_df %>%  
 mutate(Attributes = rownames(.)) %>%  
 select(Attributes, everything())  
  
flextable(importance\_df) %>%  
 set\_caption("Feature Importance Graduation Forest") %>%  
 set\_table\_properties(width = .5) %>%  
 theme\_zebra() %>%  
 set\_table\_properties(layout = "autofit")  
# Variable importance plots  
varImpPlot(  
 x = grad\_rf,  
 main = "Midfeildr Student Graduation Feature Importance"  
)  
# Full Attribute Logistic Regression Model  
graduation\_mdl <- glm(  
 formula = graduation ~ standing + gpa\_cumul + age\_desc + race + sex,  
 data = train\_data,  
 family = binomial(link = "logit"),  
 na.action = "na.omit"  
)  
# Predict test data  
test\_data2 <- test\_data  
test\_data2$y\_pred\_lr <- predict(  
 object = grad\_rf,  
 newdata = test\_data,  
 type = "response"  
)  
  
test\_data2$y\_true <- y\_true  
# Build confusion matrix  
lr\_data <- conf\_mat(  
 data = test\_data2,  
 truth = y\_true,  
 estimate = y\_pred\_lr  
)$table  
  
lr\_confusion\_mat <- as.data.frame(lr\_data)  
  
flextable(lr\_confusion\_mat) %>%  
set\_caption("Logistic Regression Confusion Matrix") %>%  
set\_table\_properties(width = .5) %>%  
theme\_zebra() %>%  
set\_table\_properties(layout = "autofit")  
  
# Calculate accuracy, sensitivity, and specificity  
accuracy\_val2 <- accuracy(lr\_data)  
sensitivity\_val2 <- sens(lr\_data)  
specificity\_val2 <- spec(lr\_data)  
  
# Create a data frame for the results  
result\_table2 <- data.frame(  
 Metric = c("Accuracy", "Sensitivity", "Specificity"),  
 Value = c(accuracy\_val2$.estimate, sensitivity\_val2$.estimate, specificity\_val2$.estimate),  
 stringsAsFactors = FALSE  
)  
  
flextable(result\_table2) %>%  
 set\_caption("Logistic Regression Accuracy, Sensitivity & Specificity ") %>%  
 set\_table\_properties(width = .5) %>%  
 theme\_zebra() %>%  
 set\_table\_properties(layout = "autofit")  
  
# Display implied attribute importance # make neat  
summary\_table <- tidy(graduation\_mdl)  
feature\_importance\_lr <- summary\_table[, c("term", "estimate")]  
feature\_lr\_5 <- feature\_importance\_lr %>% slice\_head(n = 10)  
  
flextable(feature\_lr\_5) %>%  
 set\_caption("Feature Importance Graduation Logistic Regression") %>%  
 set\_table\_properties(width = .5) %>%  
 theme\_zebra() %>%  
 set\_table\_properties(layout = "autofit")