Prediction of Graduate Admissions

Team 10:

Tatiksha Singh

Adam Toy

Erqiang Guo

Shuochen Xu

Stella (Zhuoer) Yang

TABLE OF CONTENTS

01

Project Overview

Project Goal

We wanted to predict the likelihood of admissions into a college for a graduate program

Hypotheses

High scores are something to strive for. However, how much is enough?
Higher Quant scores = higher chance to get in
Higher Undergrad GPA = higher chance to get in

Lower Quant scores = lower chance to get in
Lower Undergrad GPA = lower chance to get in

Overview of Dataset

Source: Github
(https://github.com/deedy/gradcafe_data)

271,807 rows

Dependent Variable (Y)

Decision

9 Independent Variables (X)

Major
University Name
Undergrad GPA
GRE Verbal Score
GRE Quant Score
GRE Writing Score
Status
Season
Degree

Overview of Independent Variables

Variable	Data Type	Description
Major	Qualitative	Subject/Program applied to e.g. Maths, Computer Science
University Name	Qualitative	e.g. UC Irvine, Stanford
Undergrad GPA	Integer	4.0 Scale; Some on 10.0 Scale
GRE Verbal Score	Integer	130-170 New GRE or 200-800 Old GRE
GRE Quant Score	Integer	130-170 New GRE or 200-800 Old GRE
GRE Writing Score	Integer	0 - 6, intervals of 0.5
Status	Qualitative	Applicant Background e.g. International, American
Degree	Qualitative	Either Master's e.g. MS, MBA, MFin or PhD
Season	Qualitative	Fall or Spring Admission

Preprocessing Data

- Add column names

- Clean the missing & unstandard value

- Change qualitative variables to dummy

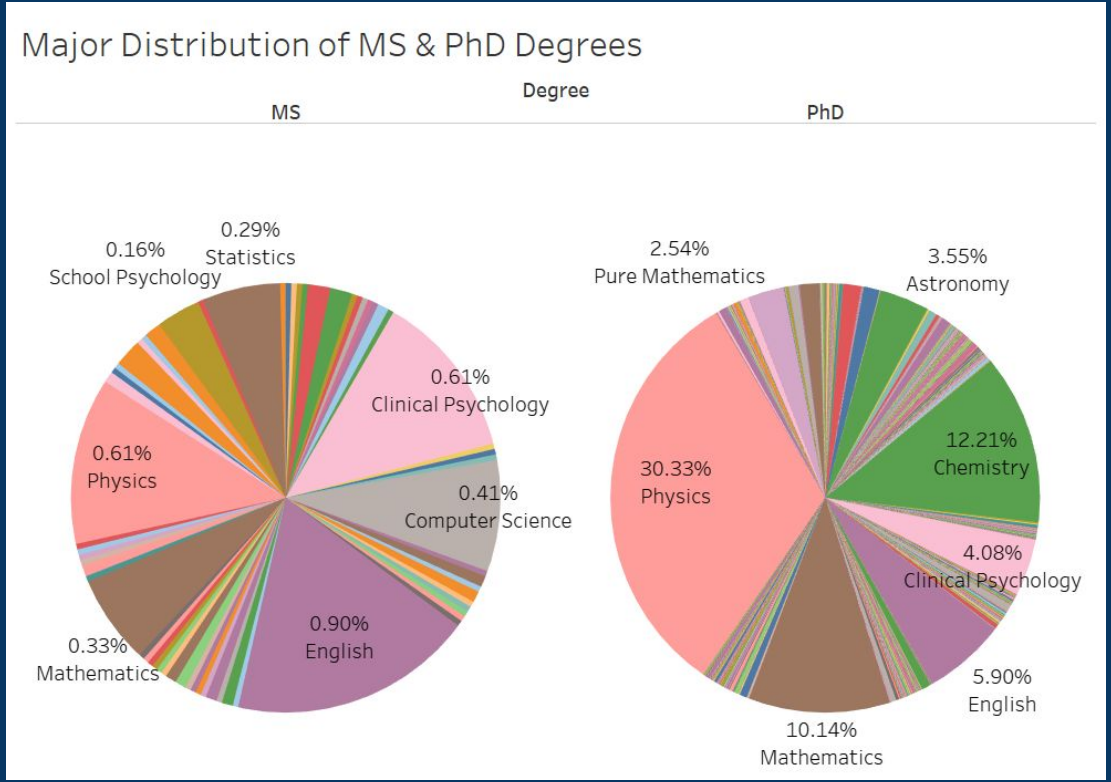
02

Exploratory Data Analysis

Initial Insights

Key Takeaway

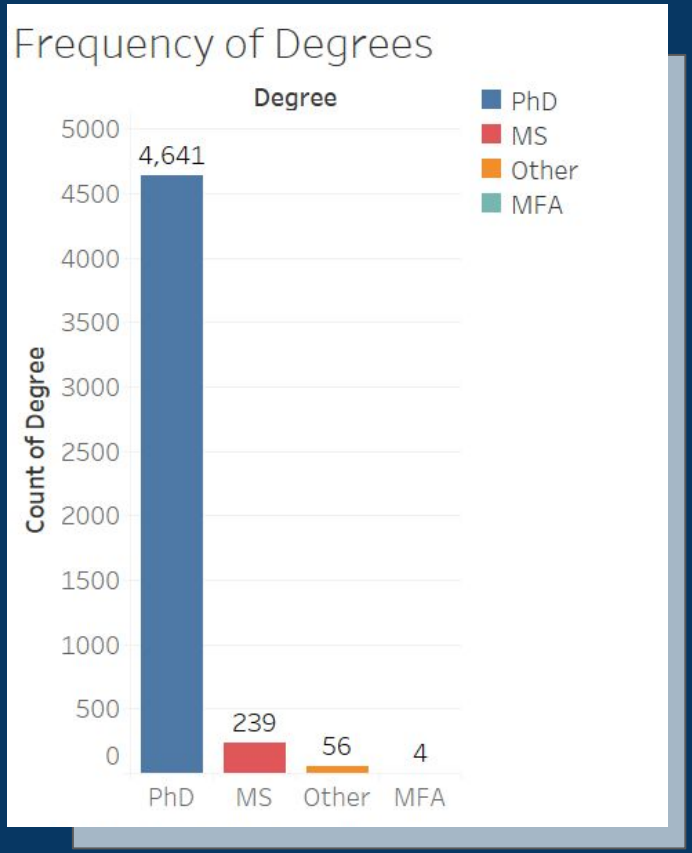
Mostly Science related fields



Initial Insights

Key Takeaways

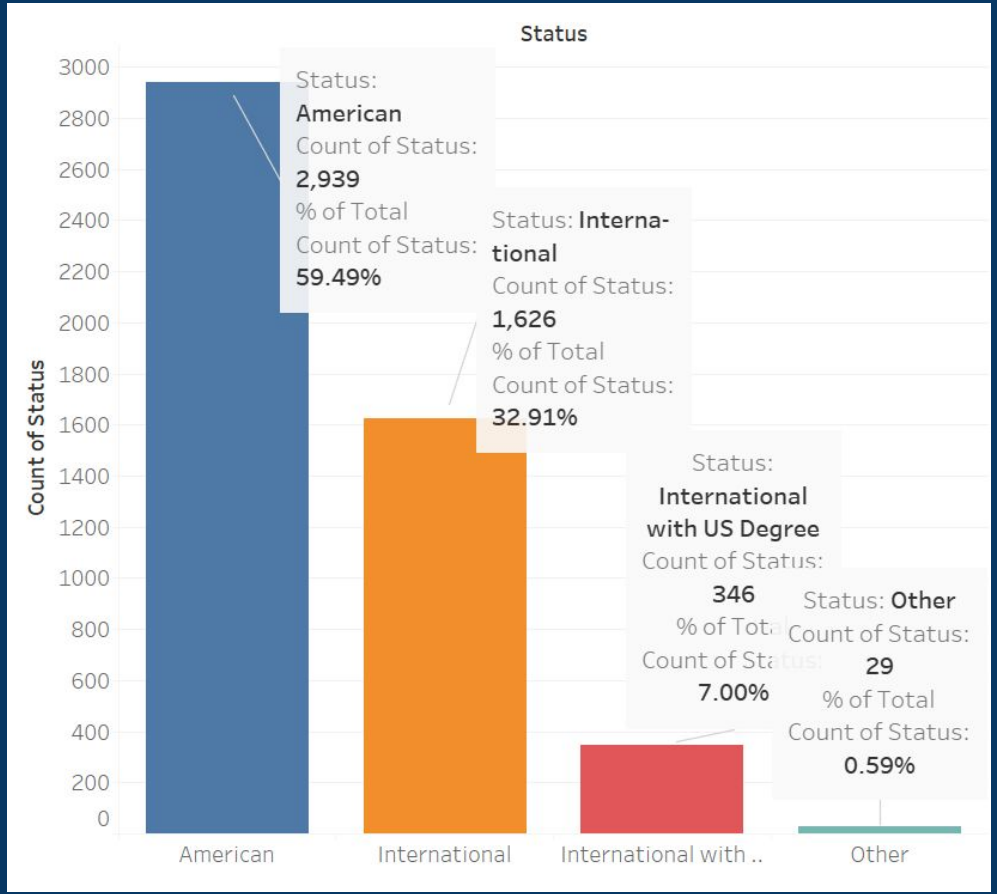
Over 90% PhD programs



Initial Insights

Key Takeaway

~60% American
~30% International



03

Predictive Models

Overview of all Models

- **All-in Regression**
- **Forward stepwise**
- **Backward stepwise**
- **Lasso**
- **Ridge**
- **SVM**
- **KNN**
- **Logistic**

	all-in	forward	backward	lasso
Adjusted R^2	0.0293	0.0310	0.0313	0.0246
RMSE	N.A	1.4750	1.4755	1.4698

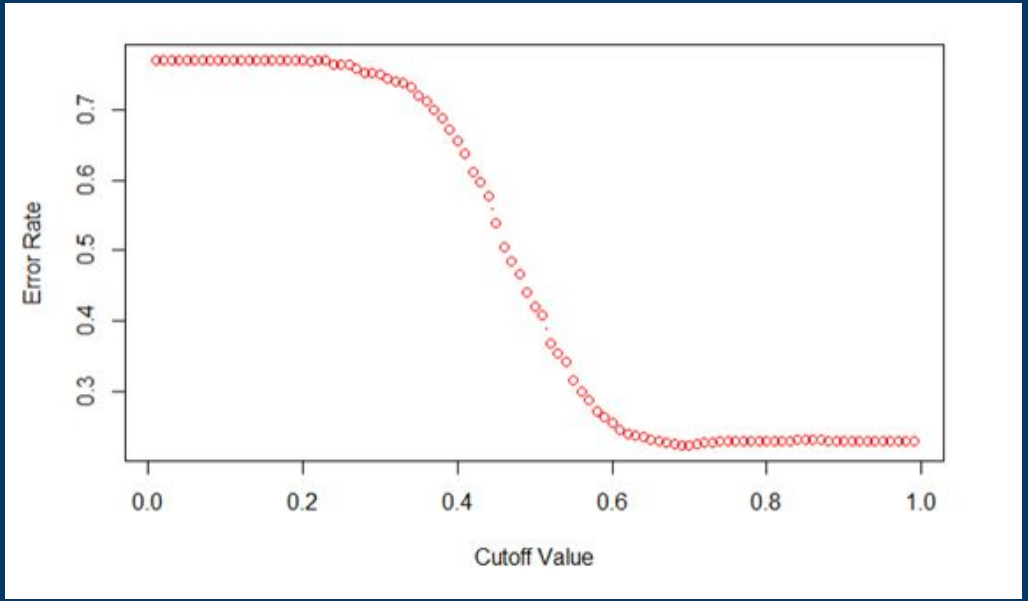
	SVM	KNN	Logistic
Error Rate	0.402	0.398	0.425

Logistic Regression Model (Whole Dataset)

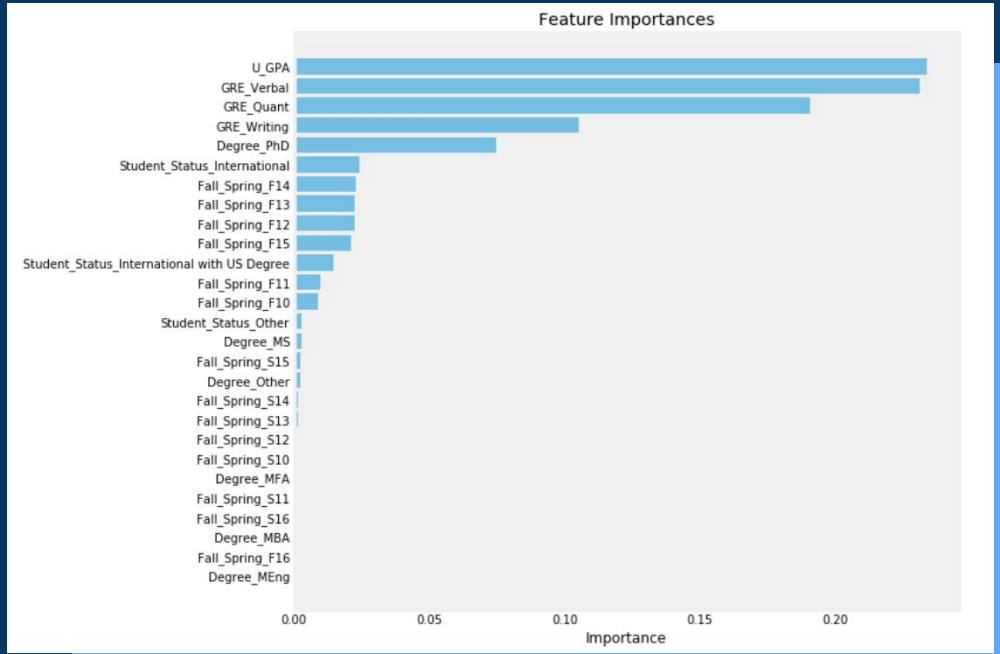

```
#  Run a logistic regression with the three most
#    highly correlated variables
#  Note the modeling syntax
#
logreg <- glm(decision ~ ., data = dat.train,
              family = "binomial")
summary(logreg)
```

```
> mean(yhat.train.class != dat.train$decision)
[1] 0.4388889
> mean(yhat.test.class != dat.test$decision)
[1] 0.4255716
```

Overall Accuracy: 57.45%



Important Features



Subsetting Data

STEM Majors

Confusion matrix

	Predicted		
	0	1	Class Error
0	643	687	0.517
1	485	1362	0.263
	Overall Accuracy: 63.11%		

```
Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.332692   1.525962  -3.495 0.000475 ***
ugrad_gpa         0.847249   0.139001   6.095 1.09e-09 ***
gre_verbal        0.009501   0.006247   1.521 0.128316
gre_quant        -0.001206   0.007612  -0.158 0.874133
gre_writing       0.211321   0.058921   3.587 0.000335 ***
dat2_MS           0.510833   0.221945   2.302 0.021357 *
dat2_American     0.456745   0.140588   3.249 0.001159 **
dat2_International 0.262654   0.140826   1.865 0.062167 .
dat2_Other.1      0.742451   0.884259   0.840 0.401115
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

UC's and Cal States

Accepted = 292
Rejected = 307

UNIVERSITY OF CALIFORNIA

CSU The California State University

Confusion matrix

	0	1	Class Error
0	199	108	0.352
1	123	169	0.421
	Overall Accuracy: 61.44%		

```
Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)       -8.681034   3.348481  -2.593  0.00953 **
ugrad_gpa          0.026370   0.323166   0.082  0.93496
gre_verbal         0.009312   0.015611   0.596  0.55086
gre_quant          0.038710   0.015221   2.543  0.01099 *
gre_writing        0.053840   0.134339   0.401  0.68858
dat2_MS            0.579026   0.440385   1.315  0.18857
dat2_American      0.706824   0.367691   1.922  0.05456 .
dat2_International -0.187769   0.381643  -0.492  0.62272
dat2_Other.1       14.055504 535.411294   0.026  0.97906
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ivy Leagues

PREDICTIVE MODELS

Accepted = 261
Rejected = 328

cients
cept)
gpa
erbal
uant
riting
MFA
MS
America

dat2_International	-0.20782	0.30452	-0.682	0.49495
dat2_Other.1	-0.74149	1.27296	-0.582	0.56024

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

0
1

Overall Accuracy: 65.70%

04

Conclusion

Challenges

- Working with mainly quantitative data
    - Does not take into account Letters of Rec, Statement of Purpose, or other Qualitative attributes
- User - supplied information = a lot of inconsistency
    - E.g. University of California Irvine , UCI , UC Irvine , Irvine
- PhDs have heavy emphasis on faculty, fit, and research focus

Pursuing a PhD/another MS?

If you want to pursue something in STEM...
It helps if you have a high undergrad GPA, have a high GRE Writing score, are American, and pursuing an MS

If you want to pursue a school in California...
It helps if you have a high GRE Quantitative score

If you want to pursue an Ivy League school...
It helps if you have a high undergrad GPA, have a high GRE Quantitative score, and pursuing an MS

Thank You!

APPENDIX

APPENDIX

R packages used:
Tidyverse
Dummy
ggplot2
Dplyr
Class
e1071
Random Forest
Caret (feature importance)

- **Summary of data acquisition to set up work (Steven)**

APPENDIX

```r
```{r}
#import data and add column name
dat <- read.csv("all_clean.csv",header = F)
names(dat) <- c("rowid","uni_name","major","degree","season","decision","decision_method","decision_date","decision_timestamp","ugrad_gpa","gre_verbal","gre_quant","gre_writing","is_new_gre","gre_subject","status","post_data","post_timestamp","comments")
names(dat)
str(dat)


#Prepare the data for analysis
#delete rows with N/A of important variables
dat1 <- na.omit(dat, cols = c("decision","ugrad_gpa","gre_verbal","gre_quant","gre_writing","status"))

#delete rows with old GRE examination and GPA greater than 5
dat2 <- dat1[!(dat1$is_new_gre=="False"| dat1$ugrad_gpa >= 5 | dat1$status ==""),]


#address degree and status
dat3 <- cbind(dat2, dummy(dat2$degree, sep = "_"),  dummy(dat2$status, sep = "_"))
names(dat3)
str(dat3)

dat3$decision <- as.numeric(dat3$decision)

#(we use status_others, degree_ohters as reference
ndat <- dat3[,c(6,10:13,20,21,23:26)]
```
```{r}
names(ndat)
str(ndat)
#head(ndat)
```
```

```
Console  Terminal
~/
> names(ndat)
 [1] "decision"                          "ugrad_gpa"
 [3] "gre_verbal"                        "gre_quant"
 [5] "gre_writing"                       "dat2_MFA"
 [7] "dat2_MS"                           "dat2_PhD"
 [9] "dat2_American"                     "dat2_International"
[11] "dat2_International with US Degree"
> str(ndat)
'data.frame':   4940 obs. of  11 variables:
 $ decision                          : num  5 5 5 5 2 3 5 5 5 5 ...
 $ ugrad_gpa                         : num  3.55 3.55 3.55 3.55 3.55 3.55 3.55 3.55 3.5 3.1 ...
 $ gre_verbal                        : num  157 157 157 157 157 157 157 157 159 146 ...
 $ gre_quant                         : num  163 163 163 163 163 163 163 163 150 170 ...
 $ gre_writing                       : num  4 4 4 4 4 4 4 4 3 3.5 ...
 $ dat2_MFA                          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ dat2_MS                           : int  0 0 0 0 0 0 0 0 0 0 ...
 $ dat2_PhD                          : int  1 1 1 1 1 1 1 1 1 1 ...
 $ dat2_American                     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ dat2_International                : int  1 1 1 1 1 1 1 1 1 1 ...
 $ dat2_International with US Degree: int  0 0 0 0 0 0 0 0 0 0 ...
>
```

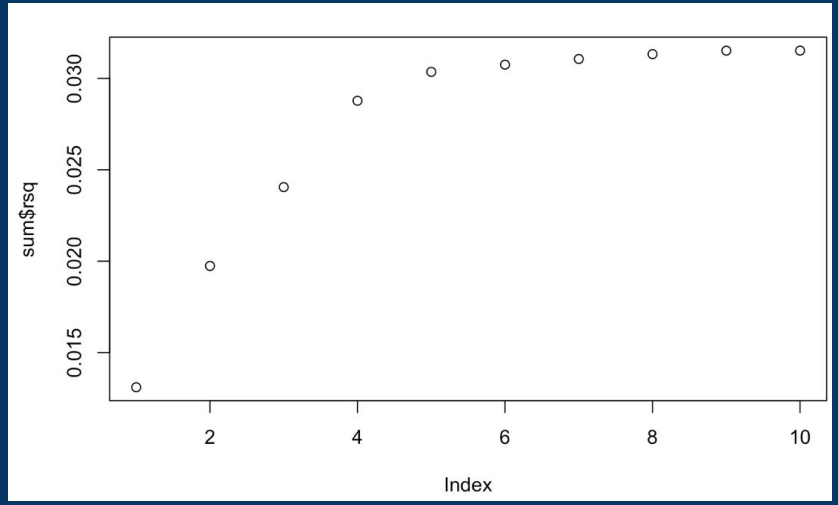
- Add column names
- Delete missing value
- Create dummy variable

Initial Models : Backward & Forward Stepwise


```
bakwd <- regsubsets(decision ~ ., ndat, nvmax = 50,
                    method = "backward", really.big = T)

sum = summary(bakwd)
sum
#sum
#sum$rsq
plot(sum$rsq)
```

```
         ugrad_gpa gre_verbal gre_quant gre_writing dat2_MFA dat2_MS dat2_PhD dat2_American dat2_International
1  ( 1 ) "*"       " "        " "       " "         " "      " "     " "      " "          " "
2  ( 1 ) "*"       " "        " "       " "         " "      " "     "*"      " "          " "
3  ( 1 ) "*"       " "        " "       " "         " "      " "     "*"      "*"          " "
4  ( 1 ) "*"       " "        "*"       " "         " "      " "     "*"      "*"          " "
5  ( 1 ) "*"       "*"        "*"       " "         " "      " "     "*"      "*"          " "
6  ( 1 ) "*"       "*"        "*"       " "         " "      " "     "*"      "*"          " "
7  ( 1 ) "*"       "*"        "*"       " "         "*"      " "     "*"      "*"          " "
8  ( 1 ) "*"       "*"        "*"       "*"         "*"      " "     "*"      "*"          " "
9  ( 1 ) "*"       "*"        "*"       "*"         "*"      "*"     "*"      "*"          " "
10  ( 1 ) "*"      "*"        "*"       "*"         "*"      "*"     "*"      "*"          "*"
         `dat2_International with US Degree`
1  ( 1 )  " "
2  ( 1 )  " "
3  ( 1 )  " "
4  ( 1 )  " "
5  ( 1 )  " "
6  ( 1 )  "*"
7  ( 1 )  "*"
8  ( 1 )  "*"
9  ( 1 )  "*"
10  ( 1 ) "*"
```



We also tried the backward and forward selection model, turns out the result is almost same as the all in model, still, ugrad_gpa,gre_verbal, gre_quant, dat2_PhD,dat2_American,dat2_International with US Degree are significant. But the R^2 is pretty low which means our model couldn't explain most of the observation

Initial Models : Lasso Regression


```
lasso.mod <- glmnet(x.train, y.train, alpha=1, thresh = 1e-12)
plot(lasso.mod, xvar="lambda", label = TRUE)

cv.out1 <- cv.glmnet(x.train, y.train, alpha = 1)
plot(cv.out1)
bestlam1 <- cv.out1$lambda.min
bestlam1
log(bestlam1)

lasso.pred <- predict(lasso.mod, s=bestlam1,
                      newx = x.test)
```

```
[1] "RMSE for back/forward variable selected,"
[1] 1.475142
[1] "RMSE for LASSO selected,"
[1] 1.469829
```

RMSE for backward/forward : 1.475

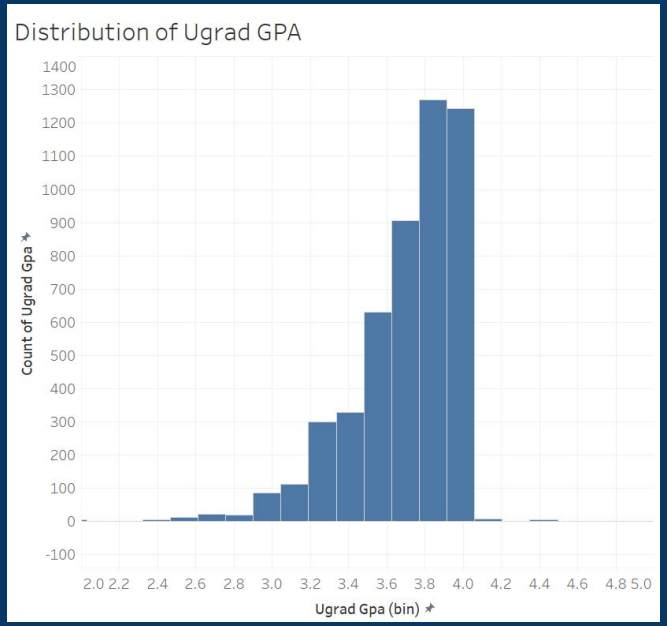
RMSE for LASSO : 1.469

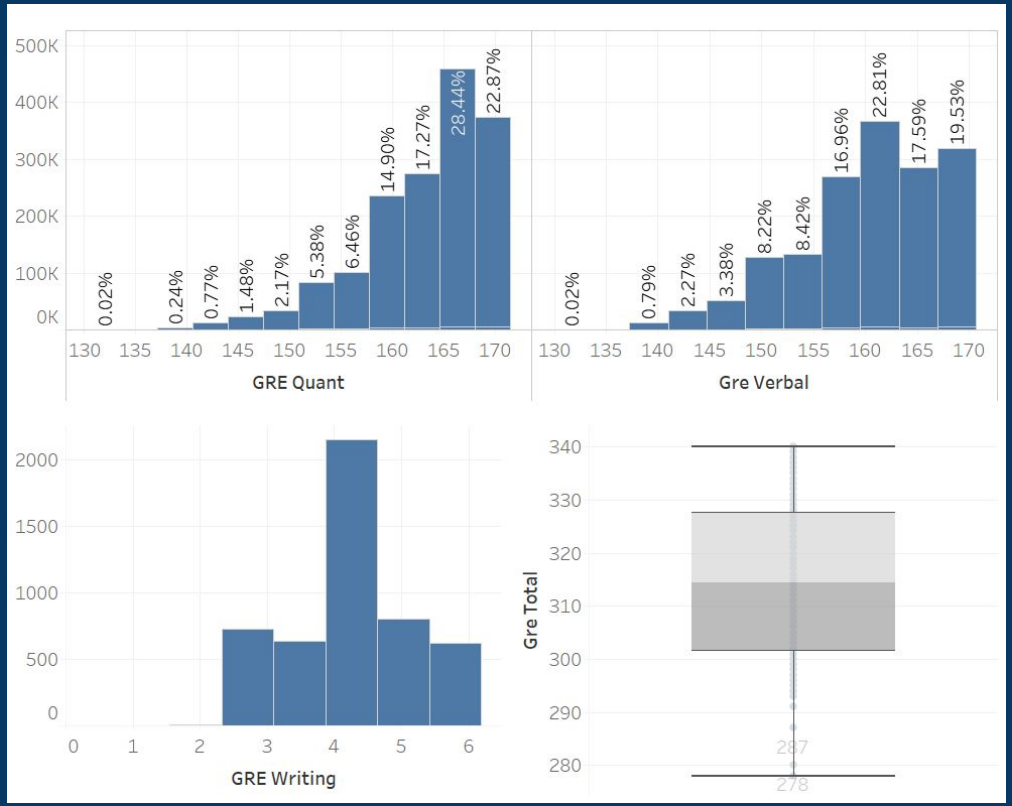
And finally we tried LASSO as our final initial model, turns out the result is the same as backward and forward selection, We are thinking it is because our variables are too few to run LASSO, so LASSO didn't show up its advantage for selecting good variables since the best scenario for LASSO is when the data dimension is higher than data observation, but in our dataset ,we only have less than 10 variables but almost 5000 observations.

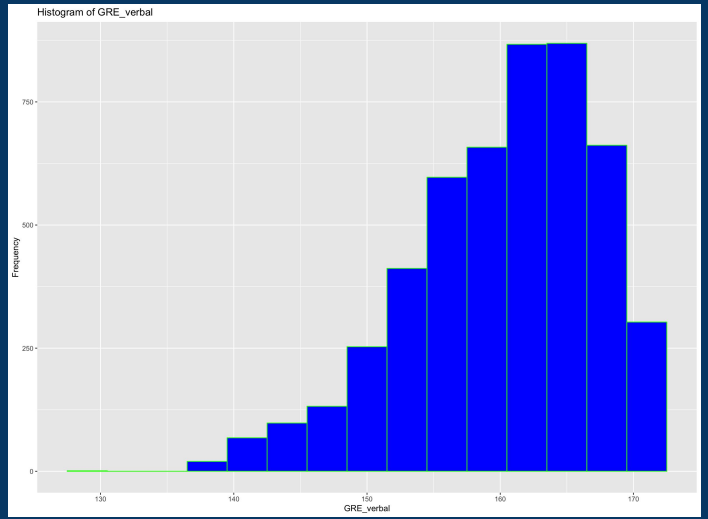
Therefore, we still have to process and transform some data such as our dependent variable into binary, and subsample the data to run machine learning method.

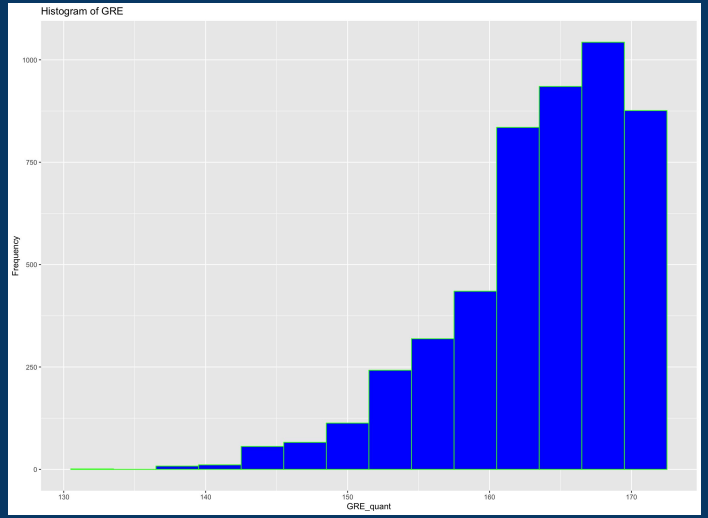
APPENDIX

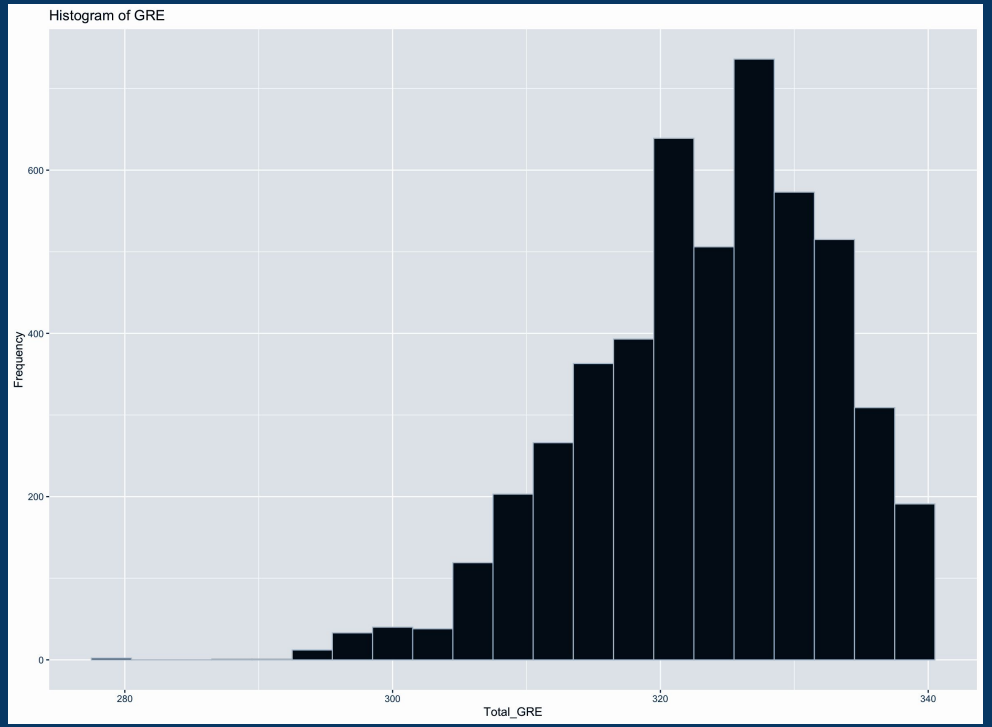
more visualization plots







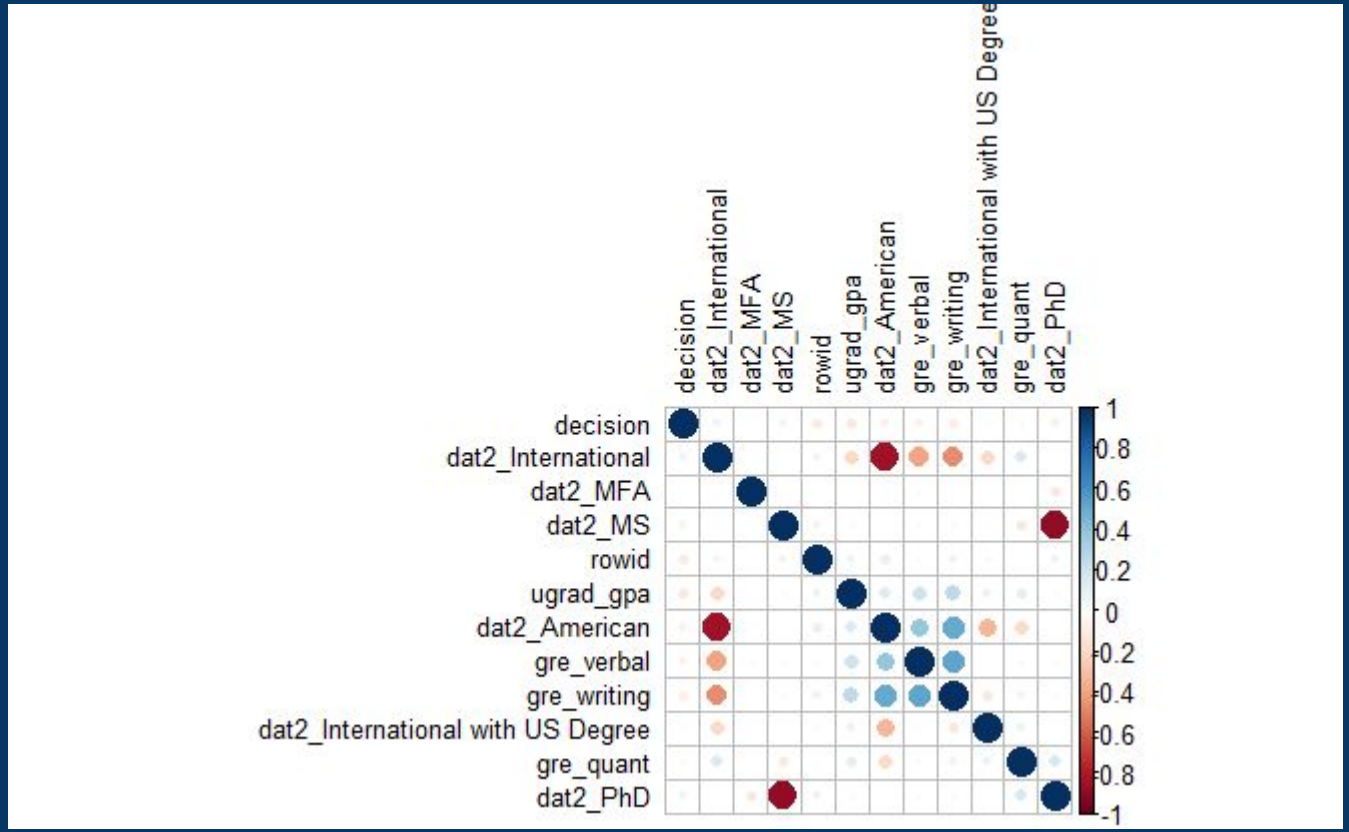


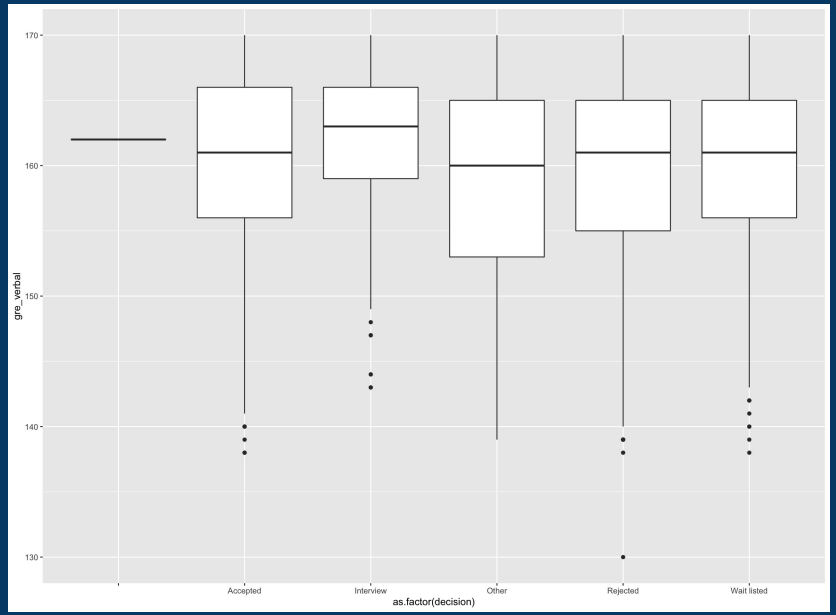


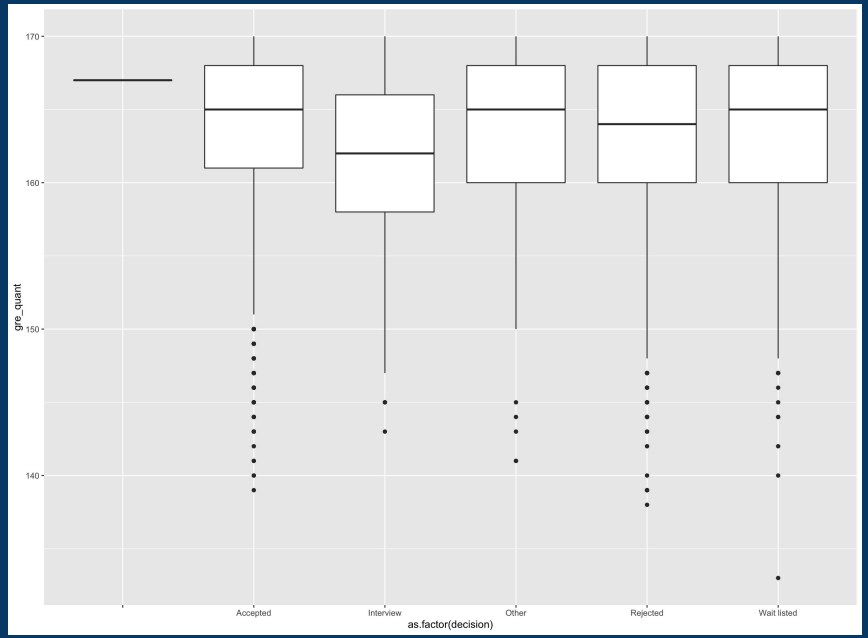
EXPLORATORY ANALYSIS

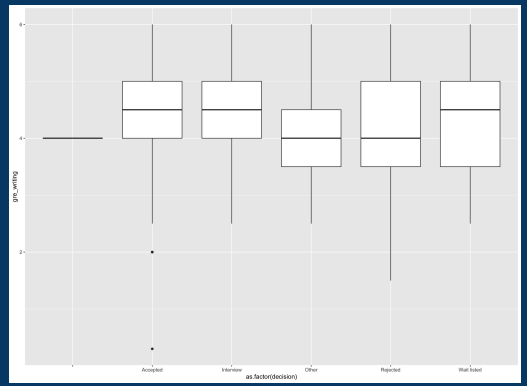
the GRE score distributions are pretty skewed. People are still getting rejected to schools even if they have super high scores.

Initial Insights




170
160
150
140
130
gre_verbal
Accepted
Interview
Other
Rejected
Wait listed
as.factor(decision)


170
160
150
140
gre_quant
Accepted
Interview
Other
Rejected
Wait listed
as.factor(decision)


6
4
2
gre_writing
Accepted
Interview
Other
Rejected
Wait listed
as.factor(decision)

Initial Models : All - In Regression

Decision (Y)

- Accepted
- Waitlisted
- Interviewed
- Rejected
- Other

$R^2 = 0.03$

```
Call:
lm(formula = decision ~ ., data = ndat)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7404 -1.3904 -0.4819  1.4773  3.4243

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                      9.056653   0.847581  10.685  < 2e-16 ***
ugrad_gpa                       -0.489357   0.079769  -6.135 9.20e-10 ***
gre_verbal                      -0.008222   0.003678  -2.235  0.02543 *
gre_quant                       -0.017865   0.003566  -5.010 5.65e-07 ***
gre_writing                     -0.038239   0.032399  -1.180  0.23795
dat2_MFA                         1.121989   0.765299   1.466  0.14269
dat2_MS                          0.215999   0.220363   0.980  0.32704
dat2_PhD                         0.770005   0.201087   3.829  0.00013 ***
dat2_American                   -0.183585   0.275769  -0.666  0.50562
dat2_International              -0.036141   0.277494  -0.130  0.89638
`dat2_International with US Degree`  0.092259   0.285780   0.323  0.74684
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.475 on 4929 degrees of freedom
Multiple R-squared:  0.03152,   Adjusted R-squared:  0.02955
F-statistic: 16.04 on 10 and 4929 DF,  p-value: < 2.2e-16
```

K-Nearest Neighbor (kNN) (Whole Dataset)


```
#  Now set up a loop to run a bunch kNNs
#    with the normalized data
#  knn.err keeps track of the errors
#
knn.err <- 1:50
xrange <- 1:50
for (j in 1:99) {
  if (j %% 2 != 0) {
    xrange[(j+1)/2] <- j
    out <- knn(dat.train.x.n, dat.test.x.n,
               dat.train.y, j)
    knn.err[(j+1)/2] <- mean(out != dat.test.y)
  }
}
```


```
#  Now set up a loop to run a bunch kNNs
#    with the normalized data
#  knn.err keeps track of the errors
#
knn.err <- 1:50
xrange <- 1:50
for (j in 1:99) {
  if (j %% 2 != 0) {
    xrange[(j+1)/2] <- j
    out <- knn(dat.train.x.n, dat.test.x.n,
               dat.train.y, j)
    knn.err[(j+1)/2] <- mean(out != dat.test.y)
  }
}
```


```
> tab.knn59
      Predicted
Actual   0   1
     0 622 335
     1 520 666
> knn59.err <- mean(dat.test.y != out59)
> knn59.err
[1] 0.3989734
>
```

First, we normalize the data. And then set up a loop to run a bunch kNNs and choose k = 59 with the lowest error rate. Here we can see the error rate is 39% which is slightly better than logistic regression.

Support Vector Machine (Whole Dataset)


```
> svmfit2.1 <- svm(decision ~ .-dat2_MFA,
+                  data = dat.train, kernel = "radial", scale = T,
+                  gamma = 1, cost = 1)
> summary(svmfit2.1)

Call:
svm(formula = decision ~ . - dat2_MFA, data = dat.train, kernel = "radial", gamma = 1, cost = 1,
    scale = T)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  1554

 ( 763 791 )


Number of Classes:  2

Levels:
 0 1
```

```
> table(truth = dat.test$decision,
+       predict = ypred.best2)
     predict
truth   0   1
    0 562 395
    1 468 718
> mean(dat.test$decision != ypred.best2)
[1] 0.4027065
```

```
#  Run cross validation
#
set.seed(123321)
tune.out2 <- tune(svm, decision ~ .-dat2_MFA, data = dat.train,
                  kernel = "radial", scale = T,
                  ranges = list(cost = c(0.01,
                                         0.1, 1, 10,
                                         100, 1000),
                                gamma = c(0.5, 1, 2, 3, 4)))
summary(tune.out2)
#
```

First we Run the SVM process on randomly selected values of cost and gamma

After that, we run cross validation and select best model

We got an error rate of 0.402

APPENDIX

Entire Dataset -> Logistic Regression - Decision (Accepted/Rejected)

Confusion matrix

	0	1	Class Error
0	1064	792	0.427
1	759	1531	0.331
Overall Accuracy: 62.59%			

```
Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)       -9.654302   1.245666  -7.750 9.17e-15 ***
ugrad_gpa          0.766527   0.121789   6.294 3.10e-10 ***
gre_verbal         0.012557   0.005501   2.283 0.022437 *
gre_quant          0.027074   0.005430   4.986 6.16e-07 ***
gre_writing        0.050917   0.048640   1.047 0.295186
dat2_MFA           0.401460   1.227834   0.327 0.743693
dat2_MS            0.835100   0.158392   5.272 1.35e-07 ***
dat2_Other         1.704088   0.453941   3.754 0.000174 ***
dat2_American      0.404088   0.129981   3.109 0.001878 **
dat2_International 0.251848   0.131583   1.914 0.055622 .
dat2_Other.1       0.037549   0.560055   0.067 0.946546
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
No. of variables tried at each split: 3

        OOB estimate of  error rate: 37.41%
Confusion matrix:
     0    1 class.error
0 1064  792   0.4267241
1  759 1531   0.3314410
```

APPENDIX

Since the dataset was not standardized, majors didn't follow the same format. We used Regex to find majors that matched words like Engineer, Math, Sci, etc.

patterns = c('Engineer', 'Math', 'Sci', 'Bio', 'Chem', 'Physics')

```
No. of variables tried at each split: 3

        OOB estimate of  error rate: 36.89%
Confusion matrix:
    0    1 class.error
0 643  687   0.5165414
1 485 1362   0.2625880
```

STEM Majors -- Logistic Regression

```
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -5.332692   1.525962  -3.495 0.000475 ***
ugrad_gpa        0.847249   0.139001   6.095 1.09e-09 ***
gre_verbal       0.009501   0.006247   1.521 0.128316
gre_quant       -0.001206   0.007612  -0.158 0.874133
gre_writing      0.211321   0.058921   3.587 0.000335 ***
dat2_MS          0.510833   0.221945   2.302 0.021357 *
dat2_American    0.456745   0.140588   3.249 0.001159 **
dat2_International  0.262654   0.140826   1.865 0.062167 .
dat2_Other.1     0.742451   0.884259   0.840 0.401115
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model Usage

If you used the STEM Majors model, you'd be accepted if you have the following attributes:
GPA = 3.75
GRE Writing = 5
MS = True (1)
American = True (1)

You'd be rejected if you have the following attributes
GPA = 3.53
GRE Writing = 3.5
MS = True (1)
American = False (0)

Just to give an idea of how you'd use the model. Plug in your attributes, and the model will output the predicted outcome status.

```
> predict1 = data.frame(ugrad_gpa = 3.75, gre_writing = 5, dat2_MS = 1, dat2_American = 1)
> predict(log_mod, predict1, interval = 'prediction', level = .95)
        1
1.213908
> predict1 = data.frame(ugrad_gpa = 3., gre_writing = 5, dat2_MS = 1, dat2_American = 1)
> predict(log_mod, predict1, interval = 'prediction', level = .95)
        1
0.5980034
> predict1 = data.frame(ugrad_gpa = 3.75, gre_writing = 3.5, dat2_MS = 1, dat2_American = 1)
> predict(log_mod, predict1, interval = 'prediction', level = .95)
        1
0.8526048
>
```

APPENDIX

UC's and Cal State

```
Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       -8.681034   3.348481  -2.593  0.00953 **
ugrad_gpa          0.026370   0.323166   0.082  0.93496
gre_verbal         0.009312   0.015611   0.596  0.55086
gre_quant          0.038710   0.015221   2.543  0.01099 *
gre_writing        0.053840   0.134339   0.401  0.68858
dat2_MS            0.579026   0.440385   1.315  0.18857
dat2_American      0.706824   0.367691   1.922  0.05456 .
dat2_International -0.187769   0.381643  -0.492  0.62272
dat2_Other.1       14.055504 535.411294   0.026  0.97906
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
        OOB estimate of  error rate: 38.56%
Confusion matrix:
    0   1 class.error
0 199 108   0.3517915
1 123 169   0.4212329
```

APPENDIX

Ivy Leagues

Accepted = 261
Rejected = 328

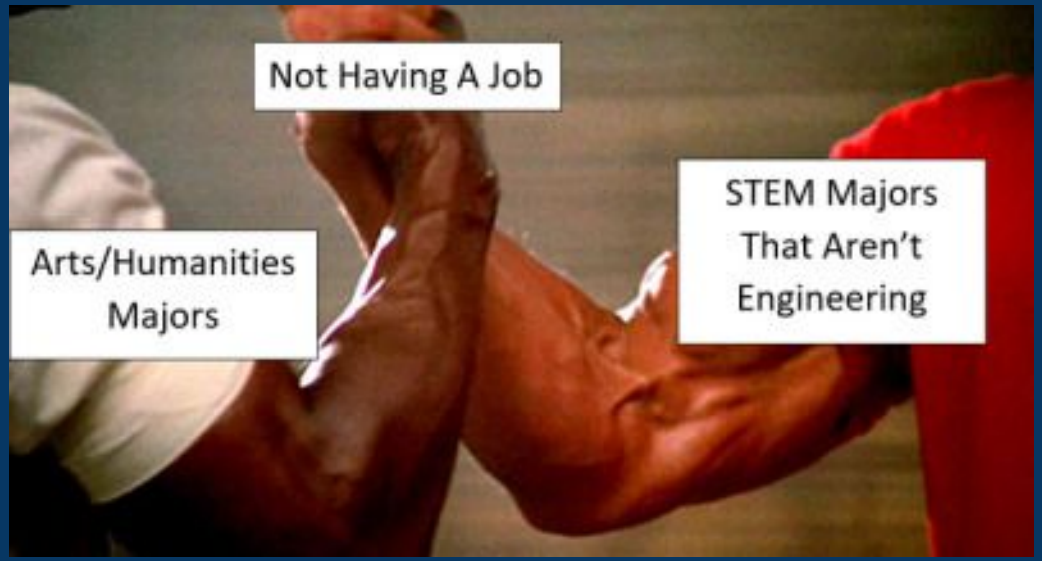
```
Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       -18.79505    4.07358  -4.614 3.95e-06 ***
ugrad_gpa           1.23198    0.44152   2.790  0.00527 **
gre_verbal          0.03181    0.01646   1.933  0.05320 .
gre_quant           0.05061    0.01651   3.066  0.00217 **
gre_writing         0.15673    0.13479   1.163  0.24493
dat2_MFA            1.30489    1.25141   1.043  0.29707
dat2_MS             1.44257    0.47597   3.031  0.00244 **
dat2_American      -0.40815    0.30226  -1.350  0.17691
dat2_International -0.20782    0.30452  -0.682  0.49495
dat2_Other.1       -0.74149    1.27296  -0.582  0.56024
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
No. of variables tried at each split: 3

        OOB estimate of  error rate: 34.3%
Confusion matrix:
    0   1 class.error
0 246  82   0.2500000
1 120 141   0.4597701
>
```

Overall Accuracy: 65.70%

UCLA:
Arts & Architecture vs. Computer Science

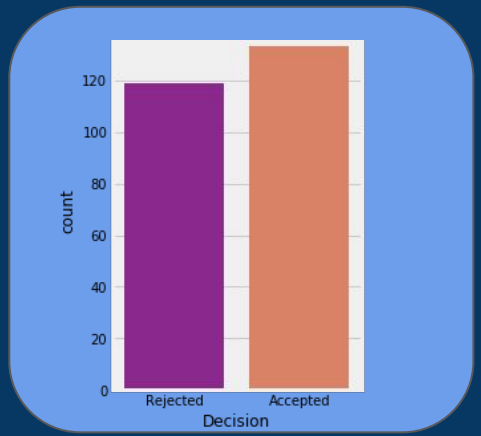


UCLA:
Arts & Architecture vs. Computer Science

PREDICTIVE MODELS

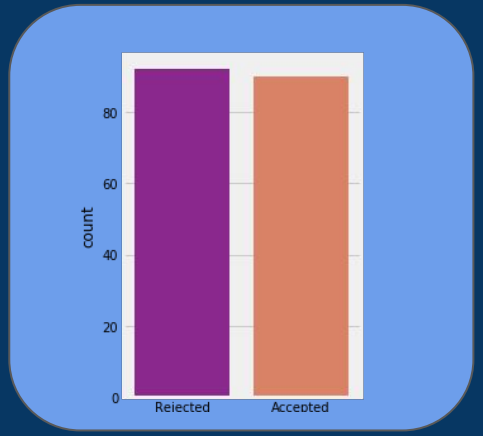
UCLA ARTS

#2 Among Fine Arts Programs

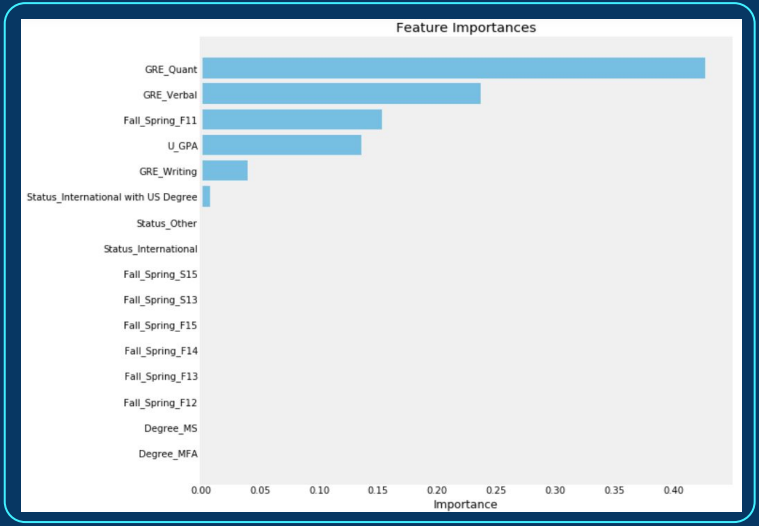


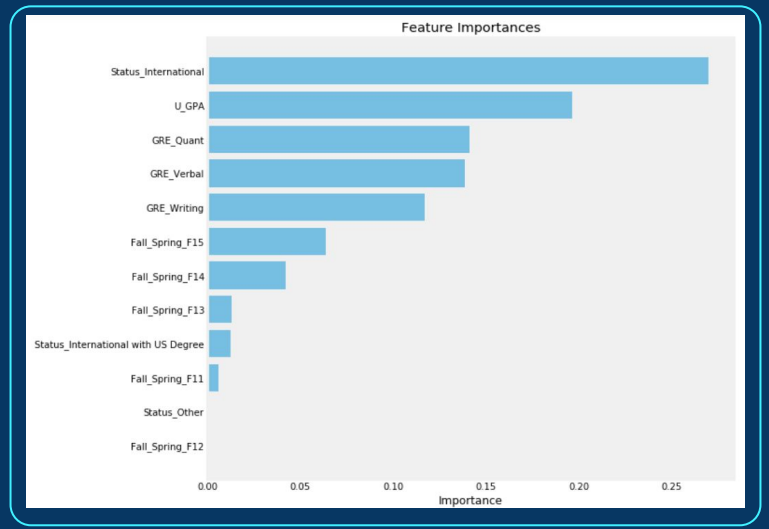
UCLA Samueli School of Engineering

#13 Among Engineering Schools



UCLA:
Arts & Architecture vs. Computer Science





- Both Departments value GRE Quant over GRE Verbal
- Primarily International Students applying to CS than Arts
- Fall > Spring

Arts Model : Decision Tree (R2 = 0.25)

CS Model : Ada Boosting (R2 = 0.57)*

More Challenges

- Data Loss - ~270,000 rows turned into ~5,000 rows
- No way of validating information
    - Misleading data could mean inaccurate predictions