**Project Report**

**By Rushikesh Khankar**

**College Admission**

Project 5

DESCRIPTION

**Background and Objective:**  
Every year thousands of applications are being submitted by international students for admission in colleges of the USA. It becomes an iterative task for the Education Department to know the total number of applications received and then compare that data with the total number of applications successfully accepted and visas processed. Hence to make the entire process easy, the education department in the US analyze the factors that influence the admission of a student into colleges. The objective of this exercise is to analyse the same.

**Domain:** Education

**Dataset Description:**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| GRE | Graduate Record Exam Scores |
| GPA | Grade Point Average |
| Rank | It refers to the prestige of the undergraduate institution. The variable rank takes on the values 1 through 4. Institutions with a rank of 1 have the highest prestige, while those with a rank of 4 have the lowest. |
| Admit | It is a response variable; admit/don’t admit is a binary variable where 1 indicates that student is admitted and 0 indicates that student is not admitted. |
| SES | SES refers to socioeconomic status: 1 - low, 2 - medium, 3 - high. |
| Gender\_male | Gender\_male (0, 1) = 0 -> Female, 1 -> Male |
| Race | Race – 1, 2, and 3 represent Hispanic, Asian, and African-American |

**Analysis Tasks:**Analyze the historical data and determine the key drivers for admission.

**Predictive:**

* Find the missing values. (if any, perform missing value treatment)
* Find outliers (if any, then perform outlier treatment)
* Find the structure of the data set and if required, transform the numeric data type to factor and vice-versa.
* Find whether the data is normally distributed or not. Use the plot to determine the same.
* Normalize the data if not normally distributed.
* Use variable reduction techniques to identify significant variables.
* Run logistic model to determine the factors that influence the admission process of a student (Drop insignificant variables)
* Calculate the accuracy of the model and run validation techniques.
* Try other modelling techniques like decision tree and SVM and select a champion model
* Determine the accuracy rates for each kind of model
* Select the most accurate model
* Identify other Machine learning or statistical techniques

My Code : {Solution}

#libraries

> library(readxl)

> library(plyr)

> library(e1071)

> library(rpart)

> library(randomForest)

> library(dplyr)

> library(ggplot2)

> setwd("/Users/rushikeshkhankar/Desktop/R")

> getwd()

[1] "/Users/rushikeshkhankar/Desktop/R"

> #Loading Dataset

> data <- read.csv("~/Desktop/R/Project/Projects for Submission/Education/Project 1\_Dataset.csv")

> View(data)

> #Data Sanity Check

> str(data)

'data.frame': 400 obs. of 7 variables:

$ admit : int 0 1 1 1 0 1 1 0 1 0 ...

$ gre : int 380 660 800 640 520 760 560 400 540 700 ...

$ gpa : num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...

$ ses : int 1 2 2 1 3 2 2 2 1 1 ...

$ Gender\_Male: int 0 0 0 1 1 1 1 0 1 0 ...

$ Race : int 3 2 2 2 2 1 2 2 1 2 ...

$ rank : int 3 3 1 4 4 2 1 2 3 2 ...

> #Checking Null values in Dataset

> is.null(data)

[1] FALSE

> #Descriptive analysis

> summary(data)

admit gre gpa ses

Min. :0.0000 Min. :220.0 Min. :2.260 Min. :1.000

1st Qu.:0.0000 1st Qu.:520.0 1st Qu.:3.130 1st Qu.:1.000

Median :0.0000 Median :580.0 Median :3.395 Median :2.000

Mean :0.3175 Mean :587.7 Mean :3.390 Mean :1.992

3rd Qu.:1.0000 3rd Qu.:660.0 3rd Qu.:3.670 3rd Qu.:3.000

Max. :1.0000 Max. :800.0 Max. :4.000 Max. :3.000

Gender\_Male Race rank

Min. :0.000 Min. :1.000 Min. :1.000

1st Qu.:0.000 1st Qu.:1.000 1st Qu.:2.000

Median :0.000 Median :2.000 Median :2.000

Mean :0.475 Mean :1.962 Mean :2.485

3rd Qu.:1.000 3rd Qu.:3.000 3rd Qu.:3.000

Max. :1.000 Max. :3.000 Max. :4.000

> str(data)

'data.frame': 400 obs. of 7 variables:

$ admit : int 0 1 1 1 0 1 1 0 1 0 ...

$ gre : int 380 660 800 640 520 760 560 400 540 700 ...

$ gpa : num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...

$ ses : int 1 2 2 1 3 2 2 2 1 1 ...

$ Gender\_Male: int 0 0 0 1 1 1 1 0 1 0 ...

$ Race : int 3 2 2 2 2 1 2 2 1 2 ...

$ rank : int 3 3 1 4 4 2 1 2 3 2 ...

> boxplot(data$gre)

> quantile(data$gre, c(0,0.05,0.1,0.25,0.5,0.75,0.90,0.95,0.97,0.98,0.985,0.99,0.995,1))

0% 5% 10% 25% 50% 75% 90% 95% 97% 98% 98.5% 99%

220 399 440 520 580 660 740 800 800 800 800 800

99.5% 100%

800 800

> data<-data[data$gre>=350,]

> boxplot(data$gpa)

> quantile(data$gpa, c(0,0.05,0.1,0.25,0.5,0.75,0.90,0.95,0.97,0.98,0.985,0.99,0.995,1))

0% 5% 10% 25% 50% 75% 90% 95% 97% 98% 98.5%

2.2600 2.7465 2.9010 3.1400 3.4000 3.6700 3.9400 4.0000 4.0000 4.0000 4.0000

99% 99.5% 100%

4.0000 4.0000 4.0000

> data<-data[data$gpa>=2.5,]

> #Logistic REgression on full data -- 70.10% accuracy

> logistic <- glm(admit ~ ., data = data, family = binomial())

> logistic

Call: glm(formula = admit ~ ., family = binomial(), data = data)

Coefficients:

(Intercept) gre gpa ses Gender\_Male

-3.193093 0.002309 0.902663 -0.167424 -0.191090

Race rank

-0.173945 -0.540918

Degrees of Freedom: 387 Total (i.e. Null); 381 Residual

Null Deviance: 486.2

Residual Deviance: 444.6 AIC: 458.6

> #Pedictef Probabilities

> result <- predict(logistic, data)

> result

1 2 3 4 5

-1.369072848 -0.661871424 1.041102671 -1.705913443 -2.552534114

6 7 8 9 10

-0.511982727 -0.624864051 -1.253865220 -1.041417928 0.364496033

11 12 13 14 15

-0.431370874 -0.661638871 0.590228230 -0.561164901 0.793057707

16 17 18 19 20

-1.278582732 -0.919121756 -2.888987412 0.076907695 0.262735091

21 22 23 24 25

-1.650067141 0.184309195 -2.115113902 -1.248517650 -0.178905853

26 27 28 29 30

0.717052426 0.447389393 -1.636808210 0.091297532 0.095061252

31 32 33 34 35

-1.230296990 -0.736968305 -1.037609120 -0.034211165 -0.774814104

36 37 38 39 40

-1.448369202 0.023549354 -1.680182009 -0.977831968 -1.704822401

42 43 44 45 46

-0.280236391 -0.746023462 -1.356270240 -0.694059059 -1.687293226

47 48 50 51 52

-0.679802077 -2.371517315 -1.544542478 -0.901581244 -2.198207220

53 54 55 56 57

-1.653903555 -0.435838970 -0.618380641 -0.202938791 -1.510618677

58 59 60 61 62

-2.134759736 -1.097861764 -2.293161502 -0.672763553 -1.756142493

63 64 65 66 67

-0.731717655 -0.650823736 -0.913748502 0.009662614 -1.421716477

68 69 70 71 73

-0.371417304 -0.088072604 0.947662924 -0.259893451 -1.871154823

74 75 76 77 78

-0.349164698 -1.453925097 -0.399398048 -1.196263241 -0.382102057

79 80 81 82 83

-0.712095360 0.625482479 -1.805479411 -1.113426014 -1.530895834

84 85 86 87 88

-2.893847522 -0.944218761 -1.067051383 -0.581947262 -0.089630305

89 90 91 92 93

-0.181084267 0.327204165 0.115832277 0.531423941 0.225349834

94 95 96 97 98

-1.158213416 -0.502511744 -0.792894968 -1.749404225 -0.985354711

99 100 102 103 104

-1.106733232 -1.945684745 -0.960524009 -2.497584244 -0.688711216

105 106 107 108 109

0.114646925 -0.417818541 0.389364677 -0.850066543 -2.127793906

110 111 112 113 114

-0.728638509 -2.030549651 -2.356677379 -1.617986348 -0.483289005

115 116 117 118 119

-0.393544876 -0.656469642 -1.025131559 -0.174550947 0.937727894

121 122 123 124 125

-0.381027576 -1.097867385 -1.718793802 -1.821573092 -0.507717597

126 127 128 129 130

-2.083010928 -0.177292018 -0.961403881 -1.028923808 -2.301398035

131 132 133 134 135

-0.671166676 -1.146046159 -0.542871547 -1.754973012 -1.186340180

136 137 138 139 140

-1.486613635 -2.099108910 -0.456201576 -0.450511505 -0.141185502

141 142 143 144 145

-0.123531801 -1.071882965 -1.218854171 -1.455346244 -1.480948851

146 147 148 149 150

-1.351085373 -0.977904662 -1.943896870 -0.866249187 0.286246556

151 152 153 154 155

0.812931668 -1.000211207 -0.107983820 -1.220719671 -0.729656857

156 157 158 159 160

-2.303235008 -1.731285817 0.073383147 -0.457378599 -0.451256804

161 162 163 164 165

-0.596652626 -0.146641099 -0.486401106 -0.951347568 -0.646180542

166 167 168 169 170

0.445166814 -2.631374146 -0.600489164 -0.900856931 -1.043156289

171 172 173 174 175

-1.591206718 -1.708720297 -1.128566895 -0.020854130 -1.548322758

176 177 178 179 181

-0.668681696 -1.415676238 -1.509917543 -1.402506395 -0.822362642

182 183 184 185 186

-1.215245884 0.071673315 -0.251102616 -2.199148449 0.115453706

187 188 189 190 191

-1.120876751 -1.551237454 -1.084541983 -1.144281880 -0.658653677

192 193 194 195 196

-0.288533358 -0.626811784 -1.771243112 -0.272602380 -0.273787732

197 198 199 200 201

-1.209990518 -2.206822324 -1.010463450 -1.655757211 -0.670091272

202 203 204 205 206

-0.729925480 0.468833024 -1.537804899 0.647433633 -0.434869213

207 208 209 210 211

0.661283118 0.153108274 -1.566733727 -0.452605257 -1.315947691

212 213 214 215 216

-1.250919201 -1.311405718 -0.628454147 -0.352265461 -1.347895231

218 219 220 221 222

-0.601376181 -1.029709603 -0.424066969 -1.482080389 -0.691875024

223 224 225 226 227

-0.588908667 -0.525665208 -0.683834421 -0.335414513 -0.127129440

228 229 230 231 232

-2.060078681 -0.411843578 -0.375503587 -1.928132663 -1.176116230

233 234 235 236 237

-1.035767722 -2.651154428 0.623372458 -0.790109731 -0.300578011

238 239 240 241 242

-0.429785568 -2.135962046 -0.768097220 -1.624302753 0.223076421

243 244 245 246 247

-1.273726922 -0.960881720 -0.275514757 -0.480486577 -0.366131215

248 249 250 251 252

-1.421635057 -1.007807040 -0.671036529 -1.718837968 -1.716895348

253 254 255 256 257

-0.163480079 -2.120648539 -0.979522922 -0.829479480 -0.436829713

258 259 260 261 262

-0.160116313 -0.605784902 -0.405796795 -0.597409893 -1.295930430

263 264 265 266 267

-1.268130530 -0.842855394 -1.142554856 -1.773694575 -1.838979031

268 269 270 271 272

-0.834012727 -0.438278465 -2.340061537 -0.446848083 -1.430425891

273 274 275 276 277

-0.416143350 0.080365538 -0.934157224 -0.905997507 -1.207350793

278 279 280 281 282

-0.204510381 -1.778538000 -0.790455473 0.105620296 -1.913248561

283 284 285 286 287

-1.568798951 -2.231657326 -1.223049218 -2.031323478 0.139413950

288 289 291 292 293

-1.025429894 -1.366058437 -0.753269252 -0.433639571 -0.064838780

294 295 296 297 298

1.187968230 -1.006638880 -1.190304546 -0.438718518 -1.124351785

299 300 301 302 303

-0.916522402 -0.593104898 -0.866154876 -0.551112381 -1.532048358

304 306 307 308 309

0.020482739 -1.235698772 0.032921725 -0.299820743 -1.250507676

310 311 312 313 314

-1.966634508 -0.601414124 0.052991616 -0.745550580 -1.735192729

315 318 319 320 321

-1.675949707 -0.811536170 -1.599447551 -0.406766553 -1.434658881

322 323 324 325 326

-0.490048258 -2.191652915 -1.576868589 -1.710435353 0.315241854

327 328 329 330 331

-0.567156549 -0.720971544 -0.262183133 -2.240199737 -0.005327134

332 333 334 335 336

-0.740605808 -0.966177458 -1.109410627 -1.324696970 0.370231589

337 338 339 340 341

-1.461037635 -1.986117639 -0.298512787 -1.033494309 -1.962879514

342 343 344 345 346

-1.820009044 -1.145538129 -0.706019050 -1.441407797 -1.609015853

347 348 349 350 351

-0.732914976 -0.385939394 -0.994598255 -0.786073359 -0.078454789

352 353 354 355 356

-1.138294658 -1.175632012 -0.505362764 -0.305262390 -0.316516531

357 358 359 360 361

-0.389658949 -0.124969216 -0.881239624 -0.812072221 0.735951914

362 363 364 365 366

-0.217207343 -0.402905911 -1.155402261 0.250607698 -1.874340248

367 368 369 370 371

-1.641362949 -0.981428287 0.700546531 0.407413510 -0.301246316

372 374 375 376 377

-0.558117951 -0.127764102 -0.450172673 -1.589647095 -0.607934788

378 379 380 381 382

0.326239631 -1.221660900 -1.290761832 -0.053168397 -0.745082233

383 384 385 386 387

-1.074919867 0.520230866 -1.501514929 -0.918539035 0.067848112

388 389 390 391 392

-0.944013814 -0.618465303 -0.844019761 -0.554956338 0.409974920

393 394 395 396 397

-1.403553271 -0.141100840 -1.367279353 0.084564886 -1.287503713

398 399 400

-1.688947557 0.103632191 -0.966861400

> summary(result)

Min. 1st Qu. Median Mean 3rd Qu. Max.

-2.8938 -1.4028 -0.8172 -0.8517 -0.3745 1.1880

> res <- ifelse(result > 0,1,0)

> #Accuracy of the model

> accuracy <- table(res, data[,1])

> sum(diag(accuracy))/sum(accuracy)

[1] 0.6984536

> #Dropping the insignificant variables

> data <- data[, -c(4,5,6)]

> data1 <- data

> #Converting necessary variables into factor

> data$admit <- as.factor(data$admit)

> data$admit

[1] 0 1 1 1 0 1 1 0 1 0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 1 1 1 1 1 0 0 0 0 1 0 0

[37] 0 0 1 1 1 1 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0

[73] 0 0 1 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0

[109] 0 0 0 0 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1

[145] 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0

[181] 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 1 0 1 1 0 1 0 0 0 0 0 0 1 1 1 0 1 0 0 1

[217] 0 0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0

[253] 0 0 0 1 1 1 0 0 1 1 0 1 0 1 0 0 1 0 1 1 1 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0

[289] 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1

[325] 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1

[361] 1 1 0 0 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 1 1 1 1 0 0 0 0 0

Levels: 0 1

> #Naive Bayes----

> naive\_bayes <- naiveBayes(admit~., data = data)

> summary(naive\_bayes)

Length Class Mode

apriori 2 table numeric

tables 3 -none- list

levels 2 -none- character

isnumeric 3 -none- logical

call 4 -none- call

> Predictions <- predict(naive\_bayes, data)

> Predictions

[1] 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0

[37] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0

[73] 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0

[109] 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0

[145] 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1

[181] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

[217] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

[253] 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 1 1 0 0

[289] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0

[325] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 1 0

[361] 0 1 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 0

Levels: 0 1

> table(Predictions, data$admit)

Predictions 0 1

0 234 88

1 30 36

> #accuracy of model

> table\_mat <- table(Predictions, data$admit)

> accuracy\_Test <- sum(diag(table\_mat))/sum(table\_mat)

> accuracy\_Test

[1] 0.6958763

> #SVM

> SVM <- svm(admit ~., data = data, method = 'class')

> summary(SVM)

Call:

svm(formula = admit ~ ., data = data, method = "class")

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 1

Number of Support Vectors: 263

( 141 122 )

Number of Classes: 2

Levels:

0 1

> Predictions <- predict(SVM, data, type ='class')

> Predictions

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38

1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0

39 40 42 43 44 45 46 47 48 50 51 52 53 54 55 56 57 58 59

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

60 61 62 63 64 65 66 67 68 69 70 71 73 74 75 76 77 78 79

0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0

80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98

1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

99 100 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

119 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138

1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157

0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176

1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0

177 178 179 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215

0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0

216 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235

0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254

0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0

255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 291 292 293

0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0

294 295 296 297 298 299 300 301 302 303 304 306 307 308 309 310 311 312 313

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

314 315 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334

0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353

0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372

0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0

374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392

0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

393 394 395 396 397 398 399 400

0 0 0 0 0 0 0 0

Levels: 0 1

> table(Predictions, data$admit)

Predictions 0 1

0 256 99

1 8 25

> #Accuracy of model

> table\_mat <- table(Predictions, data$admit)

> accuracy\_Test <- sum(diag(table\_mat))/sum(table\_mat)

> accuracy\_Test

[1] 0.7242268

> #Decision Tree

> tree <- rpart(admit~., data = data, method = 'class')

> summary(tree)

Call:

rpart(formula = admit ~ ., data = data, method = "class")

n= 388

CP nsplit rel error xerror xstd

1 0.06451613 0 1.0000000 1.0000000 0.07407561

2 0.02419355 2 0.8709677 0.8790323 0.07139664

3 0.01612903 4 0.8225806 0.9354839 0.07272369

4 0.01000000 6 0.7903226 1.0000000 0.07407561

Variable importance

gpa rank gre

52 35 13

Node number 1: 388 observations, complexity param=0.06451613

predicted class=0 expected loss=0.3195876 P(node) =1

class counts: 264 124

probabilities: 0.680 0.320

left son=2 (196 obs) right son=3 (192 obs)

Primary splits:

gpa < 3.415 to the left, improve=8.783935, (0 missing)

rank < 1.5 to the right, improve=6.907020, (0 missing)

gre < 510 to the left, improve=4.515144, (0 missing)

Surrogate splits:

gre < 610 to the left, agree=0.637, adj=0.266, (0 split)

rank < 3.5 to the right, agree=0.526, adj=0.042, (0 split)

Node number 2: 196 observations

predicted class=0 expected loss=0.2142857 P(node) =0.5051546

class counts: 154 42

probabilities: 0.786 0.214

Node number 3: 192 observations, complexity param=0.06451613

predicted class=0 expected loss=0.4270833 P(node) =0.4948454

class counts: 110 82

probabilities: 0.573 0.427

left son=6 (160 obs) right son=7 (32 obs)

Primary splits:

rank < 1.5 to the right, improve=8.0083330, (0 missing)

gre < 450 to the left, improve=1.1737770, (0 missing)

gpa < 3.945 to the left, improve=0.5037879, (0 missing)

Node number 6: 160 observations, complexity param=0.02419355

predicted class=0 expected loss=0.3625 P(node) =0.4123711

class counts: 102 58

probabilities: 0.638 0.362

left son=12 (89 obs) right son=13 (71 obs)

Primary splits:

rank < 2.5 to the right, improve=1.4024450, (0 missing)

gre < 650 to the left, improve=1.1404990, (0 missing)

gpa < 3.945 to the left, improve=0.7032468, (0 missing)

Surrogate splits:

gpa < 3.515 to the right, agree=0.594, adj=0.085, (0 split)

Node number 7: 32 observations

predicted class=1 expected loss=0.25 P(node) =0.08247423

class counts: 8 24

probabilities: 0.250 0.750

Node number 12: 89 observations

predicted class=0 expected loss=0.3033708 P(node) =0.2293814

class counts: 62 27

probabilities: 0.697 0.303

Node number 13: 71 observations, complexity param=0.02419355

predicted class=0 expected loss=0.4366197 P(node) =0.1829897

class counts: 40 31

probabilities: 0.563 0.437

left son=26 (55 obs) right son=27 (16 obs)

Primary splits:

gpa < 3.495 to the right, improve=2.600032, (0 missing)

gre < 500 to the left, improve=1.751006, (0 missing)

Node number 26: 55 observations, complexity param=0.01612903

predicted class=0 expected loss=0.3636364 P(node) =0.1417526

class counts: 35 20

probabilities: 0.636 0.364

left son=52 (26 obs) right son=53 (29 obs)

Primary splits:

gpa < 3.73 to the left, improve=2.8948640, (0 missing)

gre < 690 to the right, improve=0.3878788, (0 missing)

Surrogate splits:

gre < 610 to the left, agree=0.582, adj=0.115, (0 split)

Node number 27: 16 observations

predicted class=1 expected loss=0.3125 P(node) =0.04123711

class counts: 5 11

probabilities: 0.312 0.688

Node number 52: 26 observations

predicted class=0 expected loss=0.1923077 P(node) =0.06701031

class counts: 21 5

probabilities: 0.808 0.192

Node number 53: 29 observations, complexity param=0.01612903

predicted class=1 expected loss=0.4827586 P(node) =0.07474227

class counts: 14 15

probabilities: 0.483 0.517

left son=106 (9 obs) right son=107 (20 obs)

Primary splits:

gre < 690 to the right, improve=0.8827586, (0 missing)

gpa < 3.945 to the left, improve=0.5029606, (0 missing)

Node number 106: 9 observations

predicted class=0 expected loss=0.3333333 P(node) =0.02319588

class counts: 6 3

probabilities: 0.667 0.333

Node number 107: 20 observations

predicted class=1 expected loss=0.4 P(node) =0.05154639

class counts: 8 12

probabilities: 0.400 0.600

> Predictions <- predict(tree, data, type = 'class')

> Predictions

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38

1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0

39 40 42 43 44 45 46 47 48 50 51 52 53 54 55 56 57 58 59

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

60 61 62 63 64 65 66 67 68 69 70 71 73 74 75 76 77 78 79

0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0

80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98

1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 1 0 0 0

99 100 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118

0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0

119 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138

1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157

0 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0

158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176

1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0

177 178 179 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196

0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0

197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215

0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0

216 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235

0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1

236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254

0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0

255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273

0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0

274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 291 292 293

0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0

294 295 296 297 298 299 300 301 302 303 304 306 307 308 309 310 311 312 313

1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

314 315 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334

0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0

335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353

0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0

354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372

0 1 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0

374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392

0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1

393 394 395 396 397 398 399 400

0 1 0 1 0 0 0 0

Levels: 0 1

> table(Predictions, data$admit)

Predictions 0 1

0 243 77

1 21 47

> #Accuracy of model

> table\_mat <- table(Predictions, data$admit)

> accuracy\_Test <- sum(diag(table\_mat))/sum(table\_mat)

> accuracy\_Test

[1] 0.7474227

> #Random Forest

> forest <- randomForest(x = data, y = data$admit, ntree = 800)

> summary(forest)

Length Class Mode

call 4 -none- call

type 1 -none- character

predicted 388 factor numeric

err.rate 2400 -none- numeric

confusion 6 -none- numeric

votes 776 matrix numeric

oob.times 388 -none- numeric

classes 2 -none- character

importance 4 -none- numeric

importanceSD 0 -none- NULL

localImportance 0 -none- NULL

proximity 0 -none- NULL

ntree 1 -none- numeric

mtry 1 -none- numeric

forest 14 -none- list

y 388 factor numeric

test 0 -none- NULL

inbag 0 -none- NULL

> Predictions <- predict(forest, data)

> Predictions

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

0 1 1 1 0 1 1 0 1 0 0 0 1 0 1 0 0 0 0

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38

1 0 1 0 0 1 1 1 1 1 0 0 0 0 1 0 0 0 0

39 40 42 43 44 45 46 47 48 50 51 52 53 54 55 56 57 58 59

1 1 1 1 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0

60 61 62 63 64 65 66 67 68 69 70 71 73 74 75 76 77 78 79

0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0

80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98

1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0 0 0

99 100 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118

0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0

119 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138

1 1 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0

139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157

0 1 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 1 0

158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176

1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1

177 178 179 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196

0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0

197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215

0 1 0 0 0 1 1 0 1 1 0 1 0 0 0 0 0 0 1

216 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235

1 1 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1

236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254

0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1

255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273

1 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 1 0 1

274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 291 292 293

0 0 1 0 1 1 1 0 0 0 0 1 0 1 1 0 1 0 0

294 295 296 297 298 299 300 301 302 303 304 306 307 308 309 310 311 312 313

0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0

314 315 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334

1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353

1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1

354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372

0 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 1

374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392

1 0 0 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 1

393 394 395 396 397 398 399 400

1 1 1 0 0 0 0 0

Levels: 0 1

> table(Predictions, data$admit)

Predictions 0 1

0 264 0

1 0 124

> #Accuracy of model

> table\_mat <- table(Predictions, data$admit)

> accuracy\_Test <- sum(diag(table\_mat))/sum(table\_mat)

> accuracy\_Test

[1] 1

> Aptitude\_Descriptive = transform(data1,

+ GreLevels = ifelse(gre<439, "Low", ifelse(gre<579,"Medium","High")))

> str(Aptitude\_Descriptive)

'data.frame': 388 obs. of 5 variables:

$ admit : int 0 1 1 1 0 1 1 0 1 0 ...

$ gre : int 380 660 800 640 520 760 560 400 540 700 ...

$ gpa : num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...

$ rank : int 3 3 1 4 4 2 1 2 3 2 ...

$ GreLevels: chr "Low" "High" "High" "High" ...

> Sum\_Apt = aggregate(admit~GreLevels, data = Aptitude\_Descriptive, FUN = sum)

> lenght\_Apt = aggregate(admit~GreLevels, Aptitude\_Descriptive, FUN = length)

> Probability\_Table = cbind(Sum\_Apt, Recs = lenght\_Apt[,2])

> Probability\_Table\_Final = transform(Probability\_Table, Probability\_Admission = admit/Recs)

> Probability\_Table\_Final

GreLevels admit Recs Probability\_Admission

1 High 83 225 0.36888889

2 Low 2 29 0.06896552

3 Medium 39 134 0.29104478

> ggplot(Probability\_Table\_Final, aes(x=GreLevels, y=Probability\_Admission))+geom\_point()

> table(Aptitude\_Descriptive$admit, Aptitude\_Descriptive$GreLevels)

High Low Medium

0 142 27 95

1 83 2 39

>Chart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart

Description automatically generated