Assignment 2

2025-02-22

```
view(admit)
```

```
summary(admit)
```

```
##      admit             gre             gpa             rank      
##  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.:2.000  
##  Median :0.0000   Median :580.0   Median :3.395   Median :2.000  
##  Mean   :0.3175   Mean   :587.7   Mean   :3.390   Mean   :2.485  
##  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.   :4.000
```

#comments Admission looks imbalanced, GPA and GRE scores seem to be associated with higher chances of admission, these are the most important predictors.

For GRE our mean and median have a symmetric distribution and not a possible chance of a lot of outliers. there might be a possible light right skew due to the 660 extending towards the 800 but overall it seems like a normal distribution

```
colSums(is.na(admit))
```

```
## admit   gre   gpa  rank 
##     0     0     0     0
```

#comments we see no missing values

```
admit$admit <- as.factor(admit$admit)
```

```
table(admit$admit)
```

```
## 
##   0   1 
## 273 127
```

```
prop.table(table(admit$admit))
```

```
## 
##      0      1 
## 0.6825 0.3175
```

#comments The data set is imbalanced, with more rejections than acceptances.

```r
set.seed(1)
split <- sample.split(admit$admit, SplitRatio = 0.7)
train_data <- subset(admit, split == TRUE)
test_data <- subset(admit, split == FALSE)
```

```r
dim(train_data)
```

```
## [1] 280   4
```

```r
dim(test_data)
```

```
## [1] 120   4
```

```r
log_model <- glm(admit ~ ., data = train_data, family = binomial)
```

```r
summary(log_model)
```

```
##
## Call:
## glm(formula = admit ~ ., family = binomial, data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.091615   1.296665  -2.384 0.017112 *
## gre          0.000809   0.001253   0.646 0.518578
## gpa          0.903378   0.384148   2.352 0.018691 *
## rank        -0.502850   0.151783  -3.313 0.000923 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 350.14  on 279  degrees of freedom
## Residual deviance: 328.00  on 276  degrees of freedom
## AIC: 336
##
## Number of Fisher Scoring iterations: 4
```

#comments GPA and Rank are the most significant values, higher GPA improves your chances of getting accepted and the lower the rank of the school the lower the admission chances. Gre does not seem too significant with a value of 0.518 and does not influence the chances of admission.

```r
pred_probs <- predict(log_model, test_data, type = "response")
```

```r
pred_classes <- ifelse(pred_probs > 0.5, 1, 0)
pred_classes <- as.factor(pred_classes)
```

```r
head(pred_probs)
```

```
##         1         2         3         4         5         6
## 0.5025550 0.4183302 0.6533686 0.3211226 0.4294442 0.4497511
```

```r
head(pred_classes)
```

```
## 1 2 3 4 5 6
## 1 0 1 0 0 0
## Levels: 0 1
```

#comments

we have a 50% first applicant admitted, second applicant 41% who was not, third applicant with a 65% who was admitted and others below 50% who were not.

```r
do.call(rbind, Map(data.frame, predicted_classes = pred_classes, admit = test_data$admit))
```

```
##    predicted_classes admit
## 1                  1     0
## 2                  0     0
## 3                  1     1
## 4                  0     0
## 5                  0     1
## 6                  0     0
## 7                  0     0
## 8                  0     0
## 9                  0     0
## 10                 0     0
## 11                 0     1
## 12                 0     0
## 13                 0     1
## 14                 0     0
## 15                 0     0
## 16                 0     0
## 17                 0     0
## 18                 1     0
## 19                 0     0
## 20                 0     0
## 21                 0     0
## 22                 0     1
## 23                 0     0
## 24                 1     1
## 25                 0     0
## 26                 0     0
## 27                 0     0
## 28                 0     0
## 29                 1     1
## 30                 1     1
## 31                 0     0
## 32                 0     0
## 33                 0     0
## 34                 0     0
## 35                 0     0
## 36                 0     0
## 37                  0     0
## 38                  1     1
## 39                  0     1
## 40                  0     0
## 41                  0     0
## 42                  0     0
## 43                  0     0
## 44                  0     0
## 45                  0     1
## 46                  0     0
## 47                  0     0
## 48                  0     0
## 49                  1     0
## 50                  0     0
## 51                  0     0
## 52                  0     1
## 53                  0     1
## 54                  0     1
## 55                  0     0
## 56                  0     1
## 57                  0     0
## 58                  0     1
## 59                  0     0
## 60                  0     1
## 61                  1     1
## 62                  1     1
## 63                  0     0
## 64                  0     0
## 65                  0     1
## 66                  0     0
## 67                  0     1
## 68                  0     0
## 69                  0     0
## 70                  0     0
## 71                  0     1
## 72                  0     0
## 73                  0     0
## 74                  0     0
## 75                  0     0
## 76                  0     0
## 77                  0     1
## 78                  0     0
## 79                  0     0
## 80                  0     1
## 81                  0     0
## 82                  0     1
## 83                  0     0
## 84                  0     0
## 85                  0     1
## 86                  0     0
## 87                  1     1
## 88                  0     0
## 89                  0     1
## 90                  0     0
## 91                  0     0
## 92                  0     1
## 93                  1     1
## 94                  0     0
## 95                  0     0
## 96                  0     1
## 97                  0     0
## 98                  0     0
## 99                  0     0
## 100                 0     0
## 101                 0     1
## 102                 0     0
## 103                 1     0
## 104                 0     0
## 105                 0     1
## 106                 0     0
## 107                 0     0
## 108                 0     0
## 109                 0     1
## 110                 0     0
## 111                 0     0
## 112                 0     1
## 113                 0     0
## 114                 0     1
## 115                 0     1
## 116                 0     0
## 117                 0     1
## 118                 1     0
## 119                 0     0
## 120                 0     0
```

#comments we can see the ones that were correctly predicted and the ones the weren't in which 8 were correctly predicted out of the 10, as we have 120 rows we can see we are predicting correctly more.

```
conf_matrix <- table(Predicted = pred_classes, Actual = test_data$admit)
conf_matrix
```

```
##          Actual
## Predicted  0  1
##         0 77 29
##         1  5  9
```

#comments

we have 77 true negative, 5 false negative, 29 false positive, and 9 true positive. we have a high false positives and for our recall it seems we capture most admitted students but miss by 5.

```
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
print(conf_matrix)
```

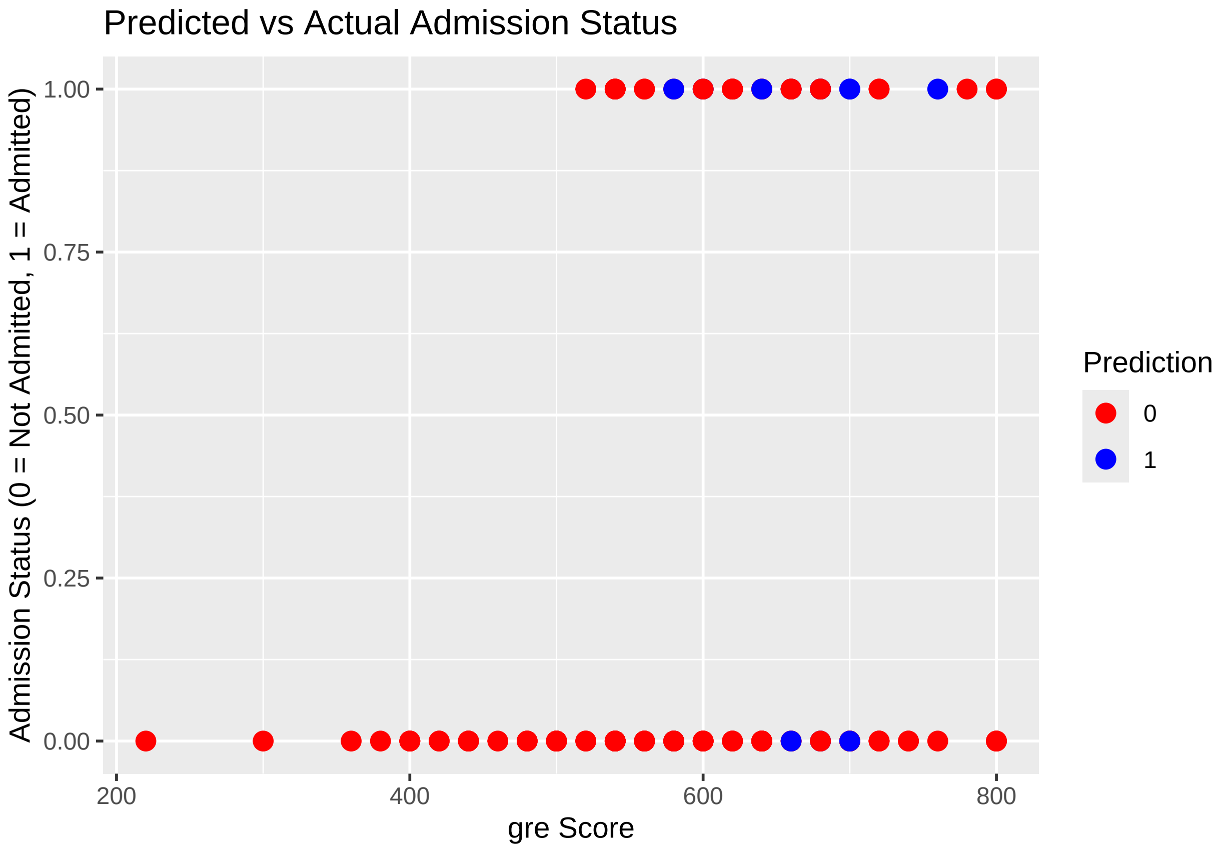
```
##          Actual
## Predicted  0  1
##         0 77 29
##         1  5  9
```

```r
print(paste("Accuracy:", round(accuracy, 4)))
```

```
## [1] "Accuracy: 0.7167"
```

our model is 71% accurate

```r
ggplot(test_data, aes(x = gre, y = as.numeric(as.character(admit)), color = as.factor(pred_classes))) +
  geom_point(size = 3) +
  labs(title = "Predicted vs Actual Admission Status",
       x = "gre Score",
       y = "Admission Status (0 = Not Admitted, 1 = Admitted)") +
  scale_color_manual(values = c("red", "blue"), name = "Prediction")
```



most of the lower GRE scores are correctly predicted as the not admitted red dots. and the higher scores seem to have mix predictions. a few false negatives and positives seem to be in our data, knowing our data is only 71% percent accurate a higher accuracy would make fewer mistakes but accuracy itself can’t be the only reason for imbalance.