"Education is the most powerful weapon which you can use to change the world."

- Nelson Mandela

Logistic Regression

By: Anirudh
Data science Training

Admission to graduate school

A researcher is interested in how variables, such as GRE (Graduate Record Exam scores), GPA (grade point average) and prestige of the undergraduate institution, effect admission into graduate school. The response variable, admit/don't admit, is a binary variable

1: As a student you are busy around preparation for your Graduate Record Exam. After the examination you received your score that typically range from 200 to 800. For sake of simplicity we will first consider only GRE score for criterion for admission and for our model preparation. Lets assume your GRE score is 660

2: Practically institute would look for your GRE score, GPA and undergraduate institution ranking for taking the admission decision… we will consider these three parameters to build our model

Admission to graduate school

Using the data we have, we would like to do the following:

1. Develop a model that will provide the probability and the odds of being getting admission for any given score.
2. Discover approximately what GRE score is associated with a probability is 50% (the odds are even) for the admission
3. Input your score of 660 into the model to determine the probability and odds of you getting admission
4. Determine how improving your credit score from 660 to 700 would effect your probability and odds for getting admission

Model Data

Sl No.	Admit	GRE_Score	GPA	Rank
1	0	380	3.61	3
2	1	660	3.67	3
3	1	800	4	1
4	1	640	3.19	4
5	0	520	2.93	4
6	1	760	3	2
7	1	560	2.98	1
8	0	400	3.08	2
9	1	540	3.39	3
10	0	700	3.92	2
11	0	800	4	4
12	0	440	3.22	1
13	1	760	4	1
14	0	700	3.08	2
15	1	700	4	1

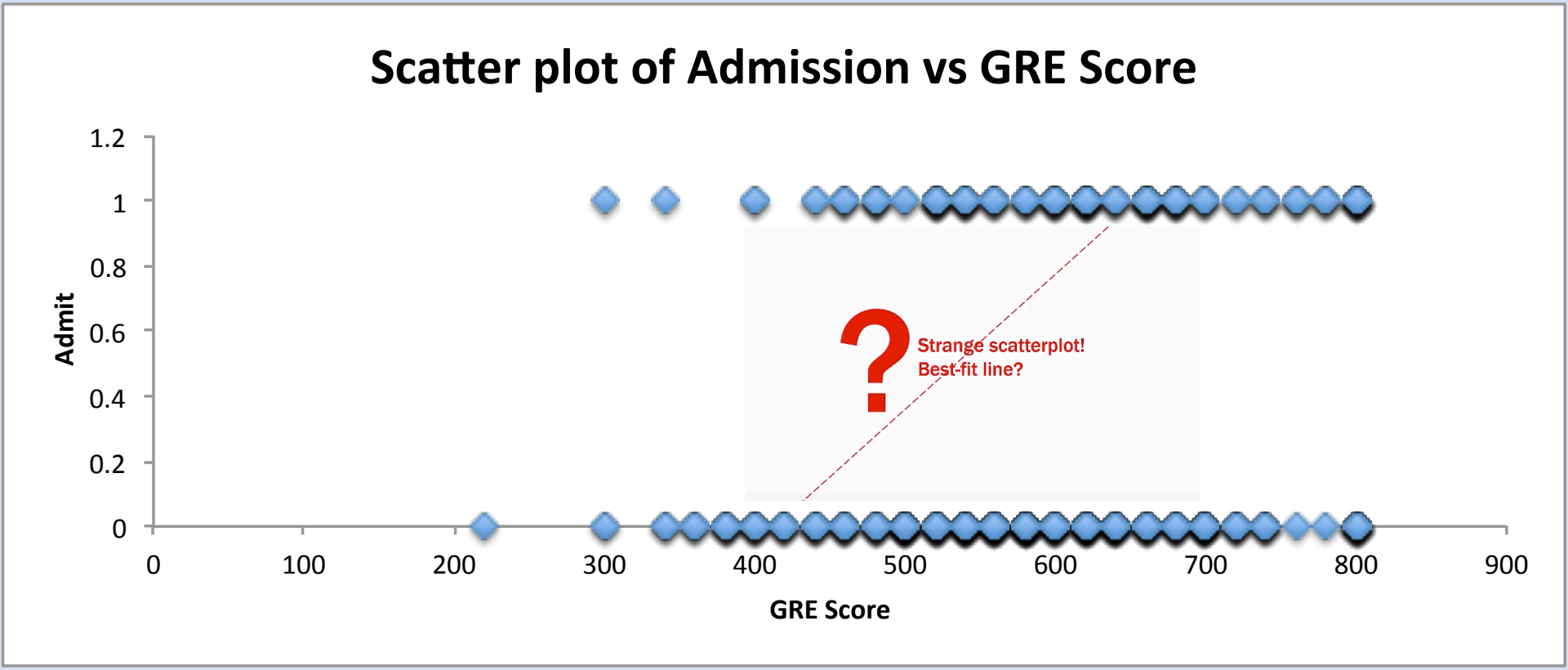
```
mydata = read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
---- or specify the path in your system--------
mydata = read.csv("/Users/asiac/Documents/Work/Study/Data Science Training Material/Analytics Training for ASPL/Data set/GRE admission Binary Data.csv")
head(mydata)
newdata=mydata[c(2,3)]
head(newdata)
str(newdata)
summary(newdata)
```

```
> head(newdata)
  admit gre
1     0 380
2     1 660
3     1 800
4     1 640
5     0 520
6     1 760
```

```
> str(newdata)
'data.frame':    400 obs. of  2 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 ...
```

```
> summary(newdata)
     admit             gre       
 Min.   :0.0000   Min.   :220.0  
 1st Qu.:0.0000   1st Qu.:520.0  
 Median :0.0000   Median :580.0  
 Mean   :0.3175   Mean   :587.7  
 3rd Qu.:1.0000   3rd Qu.:660.0  
 Max.   :1.0000   Max.   :800.0
```

Scatter plot of admission vs. GRE Score



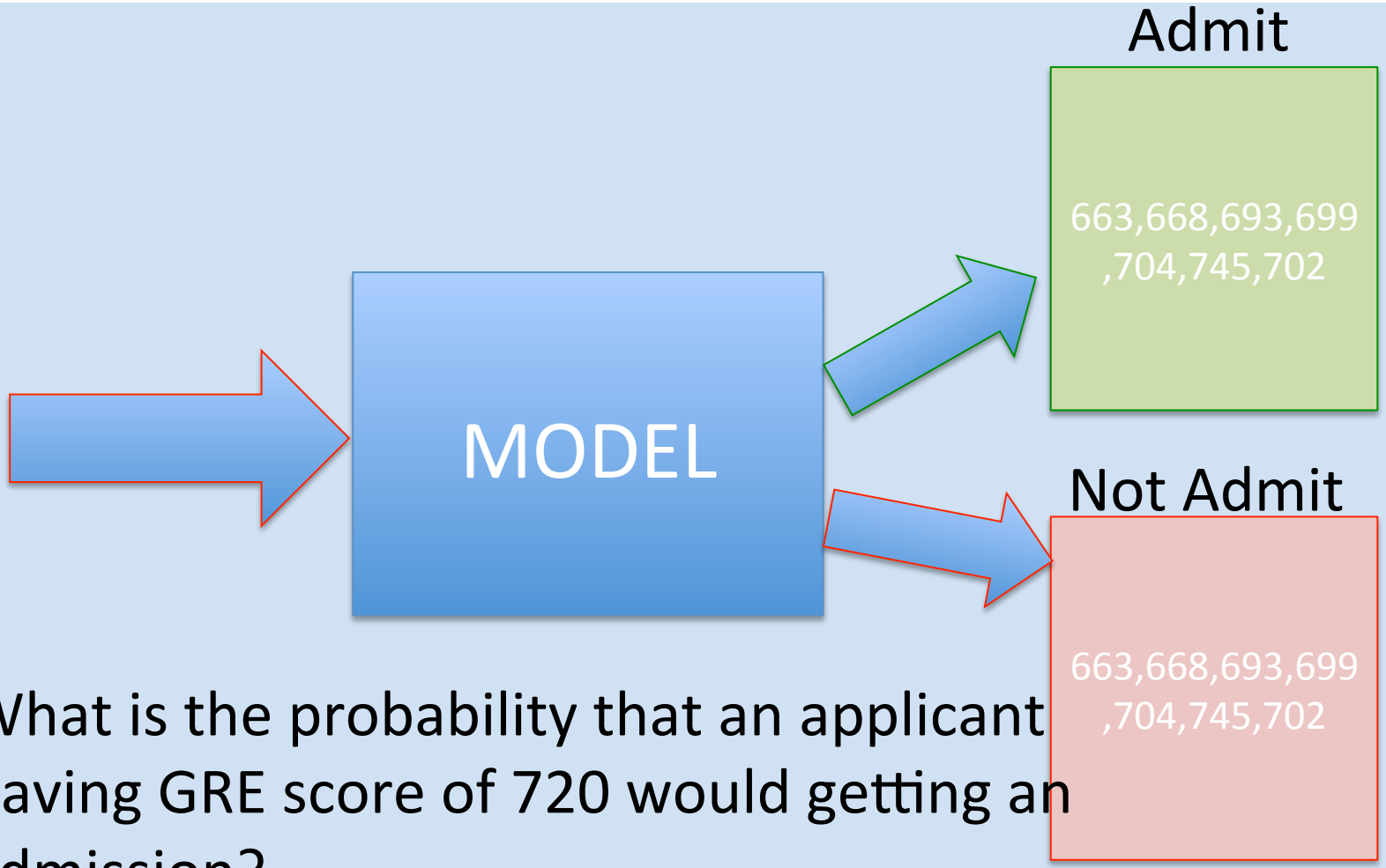
What is Logistic Regression?

Logistic regression seeks to:

- Model the probability of an event occurring depending on the values of the independent variables, which can be categorical or numerical
- Estimate the probability that an event occurs for a randomly selected observation versus the probability that the event does not occur
- Predict the effect of a series of variables on binary response variable
- Classify observation by estimating the probability that an observation is in a particular category (such as admit or not admit in our problem)

Understanding the the process

GRE_Score
380
660
800
640
520
760
560
400
540
700
800
440
760
700
700



What is the probability that an applicant having GRE score of 720 would getting an admission?

Why not other regression methods?

Why other regression procedures will not work:

- Simple linear regression is one quantitative variables predicting another
- Multiple regression is simple linear regression with more independent variables

Running a typical linear regression in the same way has major problems:

- Binary data does not have a normal distribution, which is a condition needed for most other types of regression
- Predicted values of the Dependent Variable can be beyond 0 and 1 which violates the definition of probability
- Probabilities are often not linear such as “U” shape “S” shape where probability is very low and very high at the extremes of x values

Probability Review

$$P = \frac{\textit{Outcomes of interest}}{\textit{All possible outcomes}}$$

Fair Coin flip

P (heads) = 1/2 = 0.5

Fair die roll

P (1 or 2) = 2/6 = 0.333

Deck of Playing cards

P (diamond card) = 13/52 = 0.25

What are the Odds?

$$odds = \frac{P(occurring)}{P(not\ occurring)}$$

$$odds = \frac{P}{1 - P}$$

Fair Coin flip

odds (heads) = 0.5/0.5 = 1 or 1:1

Fair die roll

odds (1 or 2) = 0.333/0.666 = 0.5 or 1:2

Deck of Playing cards

odds (diamond card) = 0.25/0.75 = 0.333 or 1:3

Odds Ratio

The odds ratio exactly it says it is, a ratio of two odds

Fair Coin flip

P (heads) = 1/2 = 0.5

odds (heads) = 0.5/0.5 = 1 or 1:1

$$\text{Odds ratio} = \frac{odds_1}{odds_0}$$

$$\text{Odds ratio} = \frac{p_1/(1-p_1)}{P_0/(1-p_0)}$$

Loaded Coin flip

P (heads) = 7/10 = 0.7

odds (heads) = 0.7/0.3 = 2.33

$$\text{Odds ratio} = \frac{0.7/0.3}{0.5/0.5} = 2.333$$

The odds of getting "heads" on the loaded coin are 2.333x greater than the fair coin.

Odds Ratio in Logistic Regression

The odds ratio for a variable in logistic regression represents how the odds change with a 1 unit increase in that variable holding all other variable constant

Lets take some example:

- Body weight and sleep apnea (two categories: apnea/no apnea)
- Weight variable had an odds ratio of 1.07
- This means a one pound increase in weight increase the odds of having sleep apnea by 1.07 (not very high b/c we are looking at 1lb increments)
- A ten pound increase in weight increase the odds to 1.98, or almost doubles a person's odds of having sleep apnea and a 20 pound increases raises the odds to 3.87 or almost 4X greater (we will calculate that later)
- This holds true at any point in the weight spectrum

A Warning

- It is very important to separate probability and odds
- In the pervious example a person gaining pounds increases their odds of sleep apnea by almost a factor of regardless of their starting weight
- However the probability of having apnea is lower in people with lower body weight to begin with
- So while odds are 4X greater, the probability may still be low
- Basically what this means is that the odds can have a large magnitude even if the underlying probabilities are low

Bringing back Bernoulli

- The dependent variable in logistic regression follows the Bernoulli distribution having an unknown probability , *p*
- Remember that the Bernoulli distribution is just a special case of the Binomial distribution where $n = 1$ (just one trial)
- Success in “1” and failure “0”
- So the probability of success is p and failure is $q = 1 - p$
- In logistic regression we are estimating an unknown p for any given linear combination of the independent variables
- Therefore we need to link together our independent variables to essentially the Bernoulli distribution; that link is called the logit

What is the Logit?

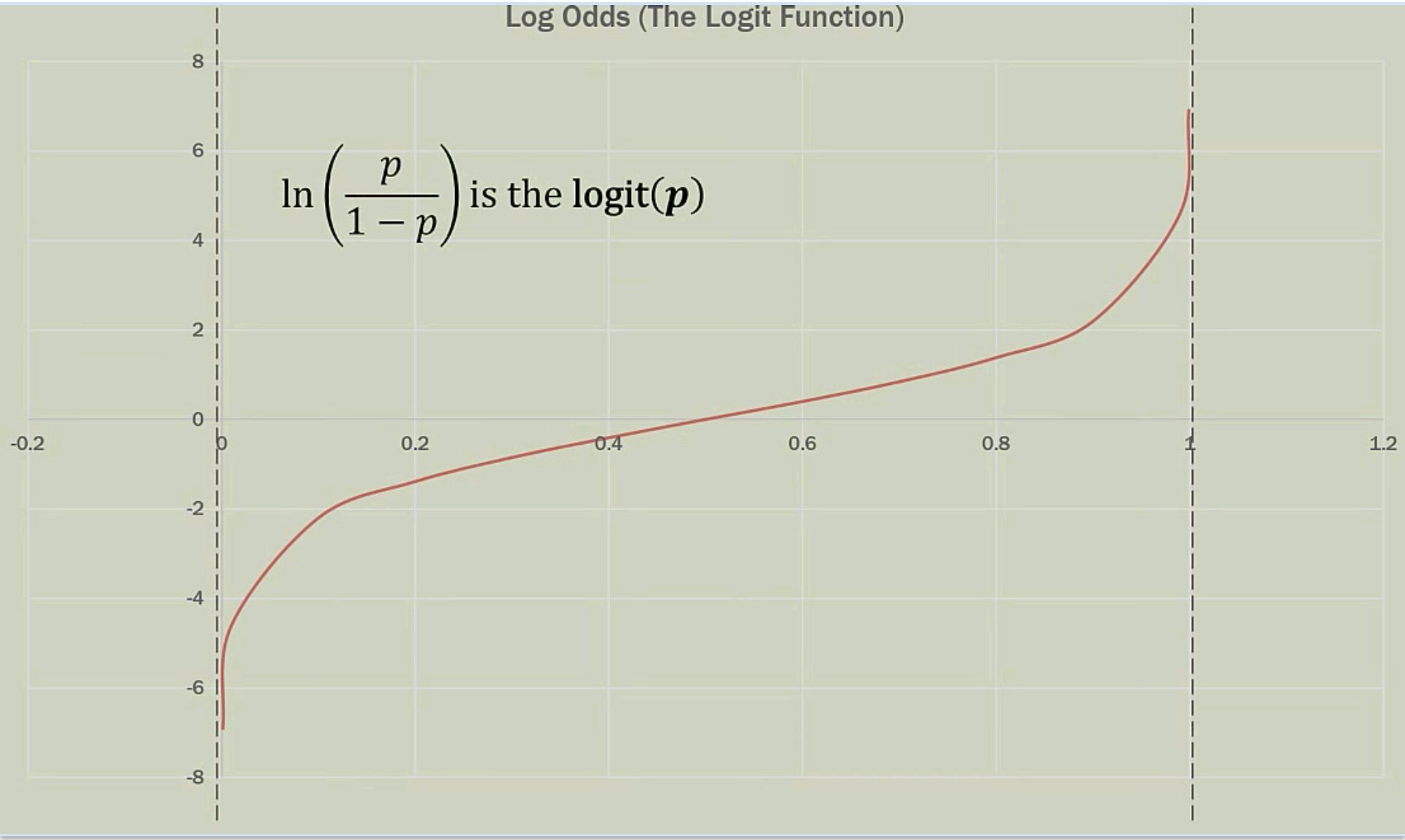
In logistic regression we do not know p like we do in binomial (Bernoulli) distribution problems. The goal of logistic regression is to estimate p for a linear combination of the independent variables. Estimate of p is p-hat.

To tie together our linear combination of variables and in essence the Bernoulli distribution we need a function that links them together, or maps the linear combination of variables that could result in any value onto the Bernoulli probability distribution with a domain from 0 to 1. the natural log of the oods ratio, the logit, is that link function

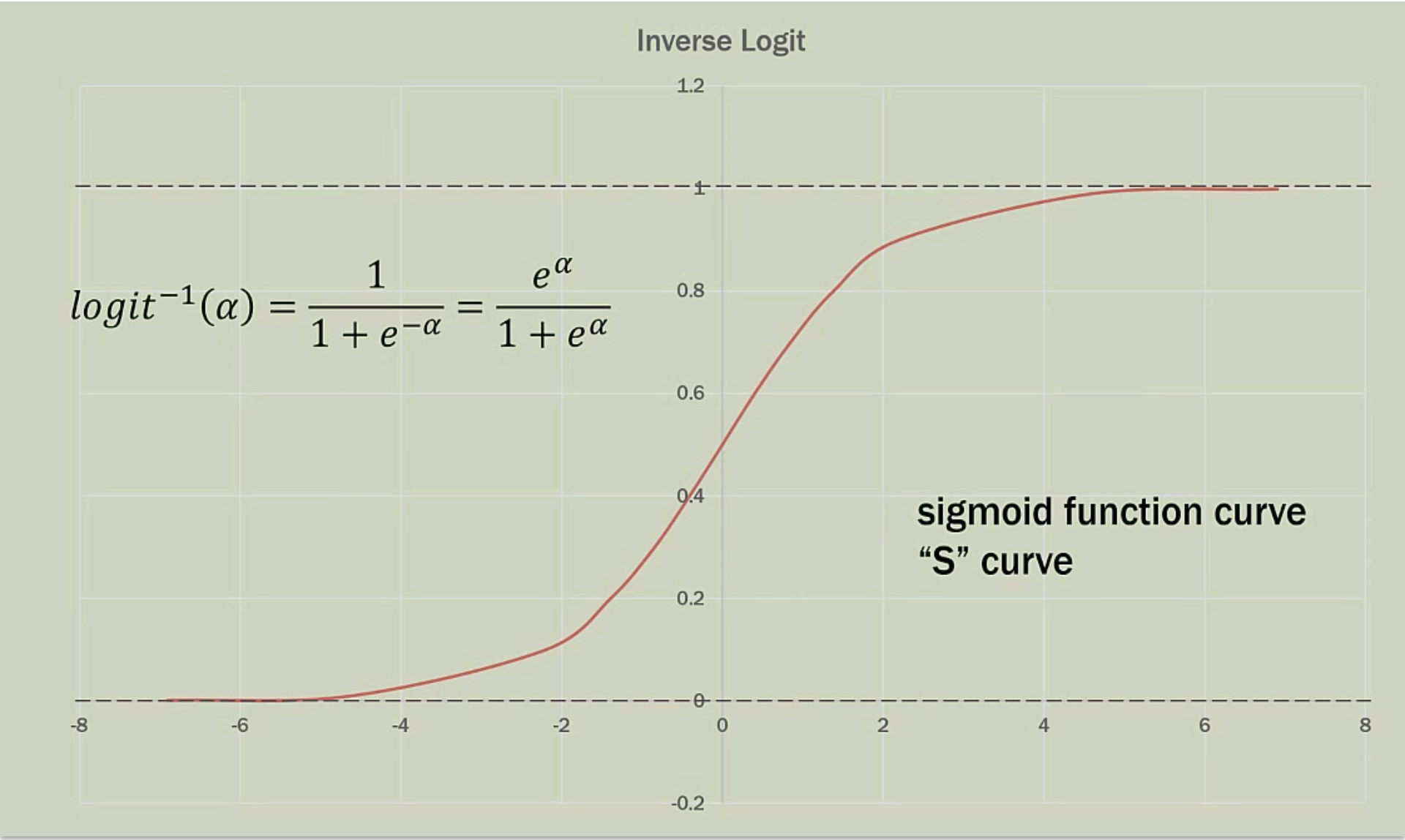
Ln(odds) → *ln(p/(1-p)) is the logit(p) OR ln(p) − ln(1-p) = logit (p)*

Reminder: $\log_e x = \ln x$

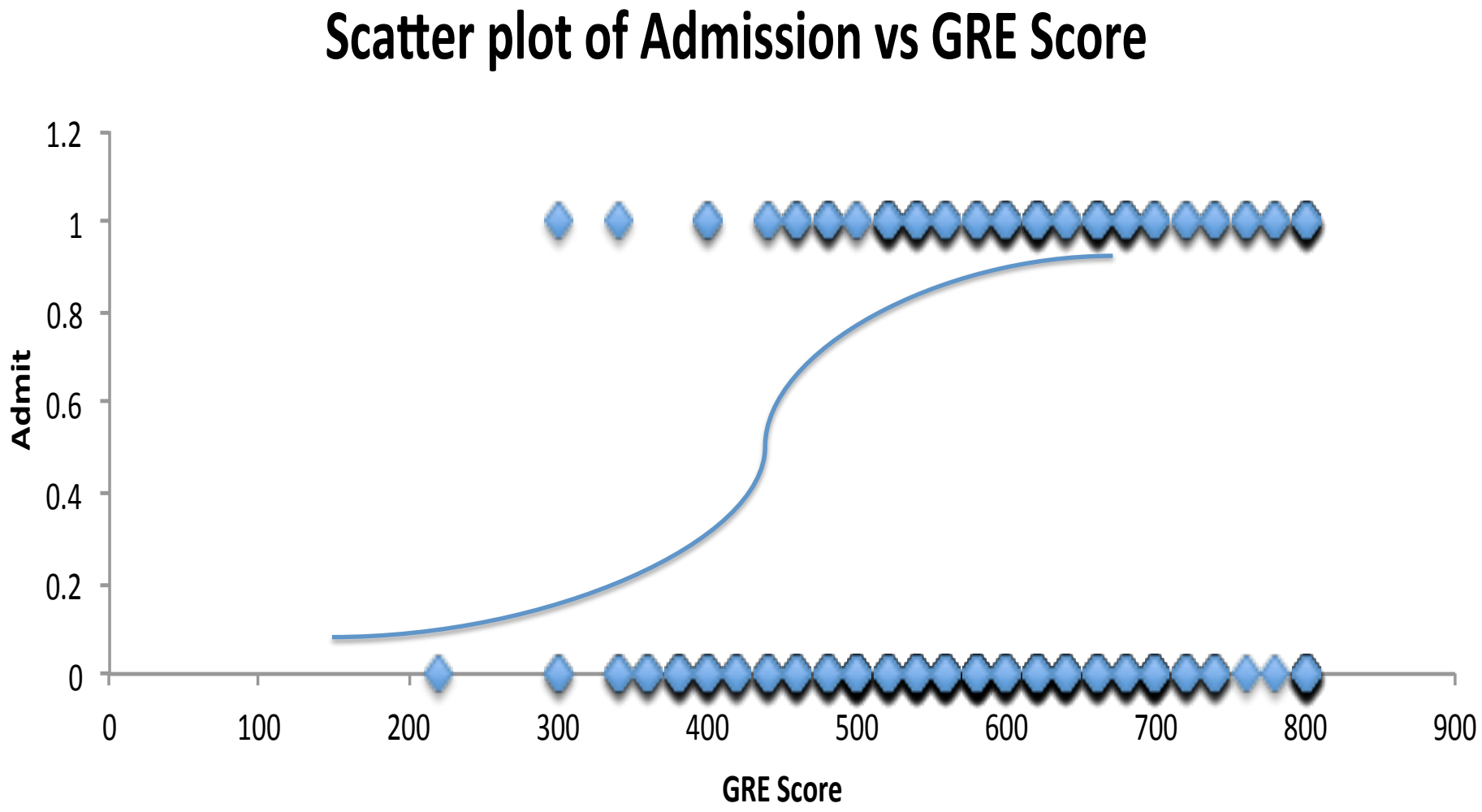
What is the Logit?



What is the inverse Logit?



Scatter plot of admission vs. GRE Score



Estimated Regression Equation

The natural logarithm of the odds ratio is equivalent to a linear function of the independent variables. The antilog of the logit function allows us to find the estimated regression equation

$$logit\ (p) = ln(p/(1-p)) = \beta_0 + \beta_1 X_1$$

$$Antilog \Rightarrow p/(1-p) = e^{(\beta 0 + \beta 1 X1)}$$

$$P\text{-}hat = (e^{(\beta 0 + \beta 1 X1)}) / (1 + e^{(\beta 0 + \beta 1 X1)})$$

Estimated Regression Equation

Model Data

Sl No.	Admit	GRE_Score	GPA	Rank
1	0	380	3.61	3
2	1	660	3.67	3
3	1	800	4	1
4	1	640	3.19	4
5	0	520	2.93	4
6	1	760	3	2
7	1	560	2.98	1
8	0	400	3.08	2
9	1	540	3.39	3
10	0	700	3.92	2
11	0	800	4	4
12	0	440	3.22	1
13	1	760	4	1
14	0	700	3.08	2
15	1	700	4	1

```
mydata = read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
---- or specify the path in your system--------
mydata = read.csv("/Users/asiac/Documents/Work/Study/Data Science Training Material/Analytics Training for ASPL/Data set/GRE admission Binary Data.csv")
head(mydata)
newdata=mydata[c(2,3)]
head(newdata)
str(newdata)
summary(newdata)
```

```
> head(newdata)
  admit gre
1     0 380
2     1 660
3     1 800
4     1 640
5     0 520
6     1 760
```

```
> str(newdata)
'data.frame':    400 obs. of  2 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 ...
```

```
> summary(newdata)
     admit             gre       
 Min.   :0.0000   Min.   :220.0  
 1st Qu.:0.0000   1st Qu.:520.0  
 Median :0.0000   Median :580.0  
 Mean   :0.3175   Mean   :587.7  
 3rd Qu.:1.0000   3rd Qu.:660.0  
 Max.   :1.0000   Max.   :800.0
```

Binary Logistic Regression: Admit versus GRE Score

```
mylogit <- glm(admit ~ gre, data = newdata, family = "binomial")
summary(mylogit)
```

```
Call:
glm(formula = admit ~ gre, family = "binomial", data = mydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1623  -0.9052  -0.7547   1.3486   1.9879

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.901344   0.606038  -4.787 1.69e-06 ***
gre          0.003582   0.000986   3.633  0.00028 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 486.06  on 398  degrees of freedom
AIC: 490.06

Number of Fisher Scoring iterations: 4
```

Binary Logistic Regression: Admit versus GRE Score

- In the 'R' output , the first thing we see is the call, this is R reminding us what the model we ran was, what options we specified, etc.
- Next we see the deviance residuals, which are a measure of model fit. This part of output shows the distribution of the deviance residuals for individual cases used in the model.
- The next part of the output shows the coefficients, their standard errors, the z-statistic (sometimes called a Wald z-statistic), and the associated p-values. GRE is statistically significant. The logistic regression **coefficients give the change in the log odds of the outcome for a one unit change in the predictor variable.**
- *For every one unit change in gre, the log odds of admission (versus non-admission) increases by 0.0035.*
- Below the table of coefficients are fit indices, including the null and deviance residuals and the AIC. Later we show an example of how you can use these values to help assess model fit

Binary Logistic Regression: Admit versus GRE Score

$\beta_0 = -2.901344$
$\beta_1 = 0.003582$

$logit\ (p) = ln(p/(1 - p)1/2 = -2.901344 + 0.003582X_1$

$Antilog \qquad p/(1-p) = e^{(-2.901344 + 0.003582X1)}$

$P\text{-}hat = (e^{(-2.901344 + 0.003582X1)}) / (1 + e^{(-2.901344 + 0.003582X1)})$
Estimated Regression Equation

Now we substitute your score 720 in the above equation
$P\text{-}hat = (e^{(-2.901344 + 0.003582*720)}) / (1 + e^{(-2.901344 + 0.003582*720)})$

Model Data let's consider other variables in the model

Sl No.	Admit	GRE_Score	GPA	Rank
1	0	380	3.61	3
2	1	660	3.67	3
3	1	800	4	1
4	1	640	3.19	4
5	0	520	2.93	4
6	1	760	3	2
7	1	560	2.98	1
8	0	400	3.08	2
9	1	540	3.39	3
10	0	700	3.92	2
11	0	800	4	4
12	0	440	3.22	1
13	1	760	4	1
14	0	700	3.08	2
15	1	700	4	1

mydata = read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")

---- or specify the path in your system--------

mydata = read.csv("/Users/asiac/Documents/Work/Study/Data Science Training Material/Analytics Training for ASPL/Data set/GRE admission Binary Data.csv")

head(mydata)

Head(mydata)

Str(mydata)

Summary(mydata)

xtabs(~admit + rank, data = mydata)

```
> str(mydata)
'data.frame':    400 obs. of  5 variables:
 $ X    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ 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 ...
```

```
> xtabs(~admit + rank, data = mydata)
     rank
admit  1  2  3  4
    0 28 97 93 55
    1 33 54 28 12
```

Binary Logistic Regression: Admit versus GRE Score, GPA & Rank

```
mydata$rank <- factor(mydata$rank)
mylogit <- glm(admit ~ gre + gpa + rank, data = mydata, family = "binomial")
summary(mylogit)
```

```
Call:
glm(formula = admit ~ gre + gpa + rank, family = "binomial",
    data = mydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6268  -0.8662  -0.6388   1.1490   2.0790

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.989979   1.139951  -3.500 0.000465 ***
gre          0.002264   0.001094   2.070 0.038465 *
gpa          0.804038   0.331819   2.423 0.015388 *
rank2       -0.675443   0.316490  -2.134 0.032829 *
rank3       -1.340204   0.345306  -3.881 0.000104 ***
rank4       -1.551464   0.417832  -3.713 0.000205 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 458.52  on 394  degrees of freedom
AIC: 470.52

Number of Fisher Scoring iterations: 4
```

Using the logit model
The code above estimates a logistic regression model using the glm (generalized linear model) function.

First, we convert rank to a factor to indicate that rank should be treated as a categorical variable.

Binary Logistic Regression: Admit versus GRE Score, GPA & Rank

Model Diagnostics
The summary(mylogit) gives the

- beta coefficients,
- Standard error,
- z Value and p Value.

If your model has categorical variables with multiple levels, you will find a row-entry for each category of that variable. That is because, each individual category is considered as an independent binary variable by the glm().
For Eg. Rank1, Rank2, Rank3

Note: *In this case it is ok if few of the categories in a multi-category variable don't turn out to be significant in the model (i.e. p Value turns out greater than significance level of 0.5).*

Binary Logistic Regression: Admit versus GRE Score, GPA & Rank

- In the 'R' output , the first thing we see is the call, this is R reminding us what the model we ran was, what options we specified, etc.
- Next we see the deviance residuals, which are a measure of model fit. This part of output shows the distribution of the deviance residuals for individual cases used in the model.
- The next part of the output shows the coefficients, their standard errors, the z-statistic (sometimes called a Wald z-statistic), and the associated p-values. Both gre and gpa are statistically significant, as are the three terms for rank. The logistic regression **coefficients give the change in the log odds of the outcome for a one unit change in the predictor variable.**
- *For every one unit change in gre, the log odds of admission (versus non-admission) increases by 0.002.*
- *For a one unit increase in gpa, the log odds of being admitted to graduate school increases by 0.804.*
- The indicator variables for rank have a slightly different interpretation. For example, having attended an undergraduate institution with rank of 2, versus an institution with a rank of 1, changes the log odds of admission by -0.675.
- Below the table of coefficients are fit indices, including the null and deviance residuals and the AIC. Later we show an example of how you can use these values to help assess model fit.

Binary Logistic Regression: Admit versus GRE Score, GPA & Rank

$\beta_0 = -3.989979 \quad \beta_1 = 0.002264 \quad \beta_2 = 0.804038$
$\beta_{R2} = -0.675443 \quad \beta_{R3} = -1.340204 \quad \beta_{R4} = -1.551464$

logit (p) = ln(p/(1 − p)1/2 = -3.989979 + 0.002264(gre) + 0.804038*(gpa) - 0.675443*(rank2) - 1.340204*(rank3) -1.551464 *(rank4) = f(x)*

Antilog $p/(1-p) = e^{f(x)}$

P-hat = $(e^{f(x)}) / (1 + e^{(f(x)})$ *Estimated Regression Equation*

Now we substitute your GRE score, GPA and Rank in the above equation to get the probability of the admission

Binary Logistic Regression: Admit versus GRE Score, GPA & Rank ~ Odds Ratio

odds ratio=exp{β}

odds ratios only
exp(**coef**(mylogit))

odds ratios and 95% CI
exp(**cbind**(OR = **coef**(mylogit), **confint**(mylogit)))

Now we can say that for a **one unit increase in gpa**, **the odds of being admitted to graduate school (versus not being admitted) increase by a factor of 2.23**.

```
> exp(coef(mylogit))
(Intercept)         gre         gpa       rank2       rank3       rank4
  0.0185001   1.0022670   2.2345448   0.5089310   0.2617923   0.2119375
```

Binary Logistic Regression: Admit versus GRE Score, GPA & Rank ~ Other Parameters

VIF

Like in case of linear regression, we should check for multicollinearity in the model. As seen below, all X variables in the model have VIF well below 4.

```
> vif(mylogit)
          GVIF Df GVIF^(1/(2*Df))
gre   1.134377  1         1.065071
gpa   1.155902  1         1.075129
rank  1.025759  3         1.004248
```

library(InformationValue)
library(car)
vif(logitMod)

Misclassification Error

Misclassification error is the percentage mismatch of predcited vs actuals, irrespective of 1's or 0's. The lower the misclassification error, the better is your model.

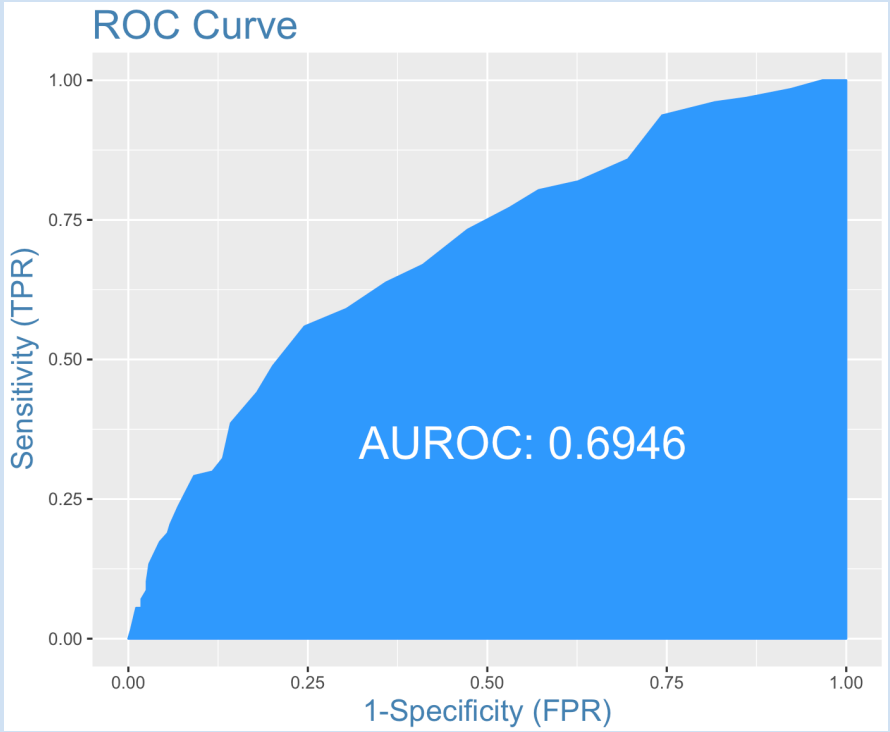
misClassError(testData$admit, predicted, threshold = optCutOff)

```
> misClassError(mydata$admit,predicted)
[1] 0.29
```

Binary Logistic Regression: Admit versus GRE Score, GPA & Rank ~ Other Parameters

ROC

Receiver Operating Characteristics Curve traces the percentage of true positives accurately predicted by a given logit model as the prediction probability cutoff is lowered from 1 to 0. For a good model, as the cutoff is lowered, it should mark more of actual 1's as positives and lesser of actual 0's as 1's. So for a good model, the curve should rise steeply, indicating that the TPR (Y-Axis) increases faster than the FPR (X-Axis) as the cutoff score decreases. Greater the area under the ROC curve, better the predictive ability of the model.



plotROC(testData$admit, predicted)

We measure area under ROC curve xx%

Binary Logistic Regression: Admit versus GRE Score, GPA & Rank ~ Other Parameters

Concordance

Ideally, the model-calculated-probability-scores of all actual Positive's, (aka Ones) should be greater than the model-calculated-probability-scores of ALL the Negatives (aka Zeroes). Such a model is said to be perfectly concordant and a highly reliable one. This phenomenon can be measured by Concordance and Discordance. In simpler words, of all combinations of 1-0 pairs (actuals), *Concordance* is the percentage of pairs, whose scores of actual positive's are greater than the scores of actual negative's. For a perfect model, this will be 100%. So, the higher the concordance, the better is the quality of model.

```
$Concordance
[1] 0.6927692

$Discordance
[1] 0.3072308

$Tied
[1] 5.551115e-17

$Pairs
[1] 34671
```

Concordance(testData$admit, predicted)

Binary Logistic Regression: Admit versus GRE Score, GPA & Rank ~ Other Parameters

Sensitivity (or True Positive Rate) is the percentage of 1's (actuals) correctly predicted by the model, while, specificity is the percentage of 0's (actuals) correctly predicted. Specificity can also be calculated as 1 − *False Positive Rate*.

$$Sensitivity = \frac{\#\ Actual\ 1's\ and\ Predicted\ as\ 1's}{\#\ of\ Actual\ 1's}$$

$$Specificity = \frac{\#\ Actual\ 0's\ and\ Predicted\ as\ 0's}{\#\ of\ Actual\ 0's}$$

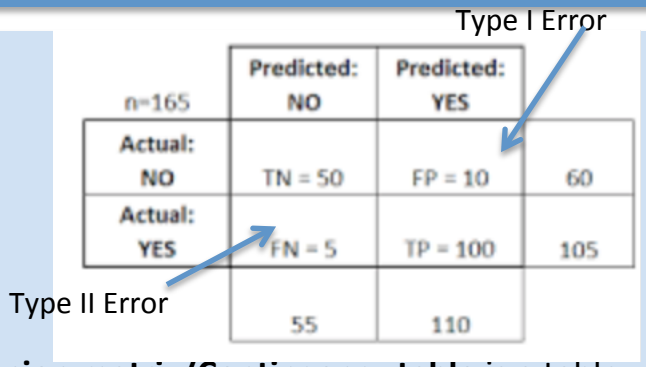
sensitivity(testData$admit, predicted, threshold = optCutOff)
specificity(testData$admit, predicted, threshold = optCutOff)

```
> sensitivity(mydata$admit,predicted)
[1] 0.2362205
> specificity(mydata$admit,predicted)
[1] 0.9304029
```

Binary Logistic Regression: Admit versus GRE Score, GPA & Rank ~ Other Parameters

Confusion Matrix
confusionMatrix(testData$ABOVE50K, predicted, threshold = optCutOff)
The columns are actuals, while rows are predicteds.

```
> confusionMatrix(mydata$admit,predicted)
    0  1
0 254 97
1  19 30
```



Confusion matrix/Contingency table is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known
True Negative (TN)
False Positive (FP) – [FPR = FP/Actual No]
False Negative (FN)
True Positive (TP) – [TPR = TP/Actual Yes]

ROC Curve plotting the true positive rate (TPR) against the false positive rate (FPR**)**

Accuracy

Sensitivity - true positive rate (TPR)
Specificity - true negative rate (TNR) [TN/Actual No]/[1-FPR]

Accuracy – (TP + TN)/Total Population
Precision – TP/Predicted Yes

All the best, for all things you do

Aniruddha,