Lecture 1

IAQS | INSTITUTE OF ACTUARIAL & QUANTITATIVE STUDIES

Class: SY BSc

Subject : Statistical Modelling in R - 1

Subject Code:

Chapter:

Chapter Name: Logistic Regression

What is Logistic Regression

- Logistic Regression is part of a larger class of algorithms known as Generalized Linear Model (glm).
- In 1972, Nelder and Wedderburn proposed this model with an effort to provide a means of using linear regression to the problems which were not directly suited for application of linear regression.

Use case of Logistic Regression

outlook	temperatu	humidity	windy	play
overcast	hot	high	FALSE	yes
overcast	cool	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
rainy	mild	normal	FALSE	yes
rainy	mild	high	TRUE	no
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
sunny	mild	normal	TRUE	yes

A	B	C	D	E	F	G	H	I	J	K
id	diagnosis	radius_me	texture_mean	perimeter_	area_mea	smoothne	compactn	concavity_	concave p	symmetry_f
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812
84300903	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069
84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597
84358402	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809
843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087
844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794
84458202	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196
844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235
84501001	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203
845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528
84610002	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842
846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397
846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847
84667401	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069
84799002	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303
848406	M	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586
84862001	M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164
849014	M	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498	0.1582
8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885
8510653	B	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967
8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815
8511133	M	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521
851509	M	21.16	23.04	137.2	1404	0.09428	0.1022	0.1097	0.08632	0.1769
852552	M	16.65	21.38	110	904.6	0.1121	0.1457	0.1525	0.0917	0.1995
852631	M	17.14	16.4	116	912.7	0.1186	0.2276	0.2229	0.1401	0.304

Features of Logistic Regression

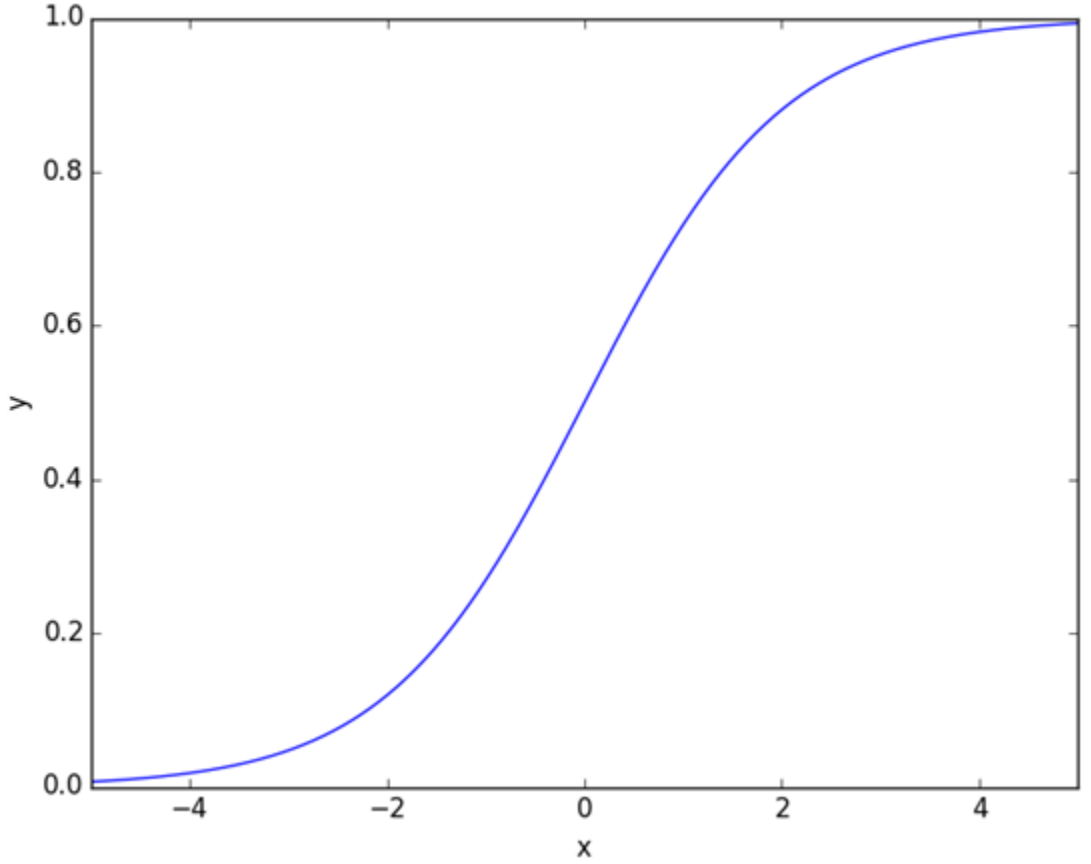
- Important Points
- GLM does not assume a linear relationship between dependent and independent variables. However, it assumes a linear relationship between link function(alpha) and independent variables in logit model.
- The dependent variable need not to be normally distributed.
- Errors need to be independent but not normally distributed.

Sigmoid function

$y = 1/(1 + \exp(-x))$

Where y is the target variable and x is the independent variable

In multiple linear regression x is replace by the linear regression equation also called link function.



➢ *The Difference Between "Probability" and "Odds"*

- Probabilities always range between 0 and 1.
- The odds are defined as the probability that the event will occur (probability of success)
- divided by the probability(probability of failure) that the event will not occur.

- If the probability of an event occurring is Y,
- then the probability of the event not occurring is 1-Y. (Example: If the probability of an event is 0.80 (80%), then the probability that the event will not occur is 1-0.80 = 0.20, or 20%.

- The odds of an event represent the ratio of the (probability that the event will occur) / (probability that the event will not occur).
- This could be expressed as follows:

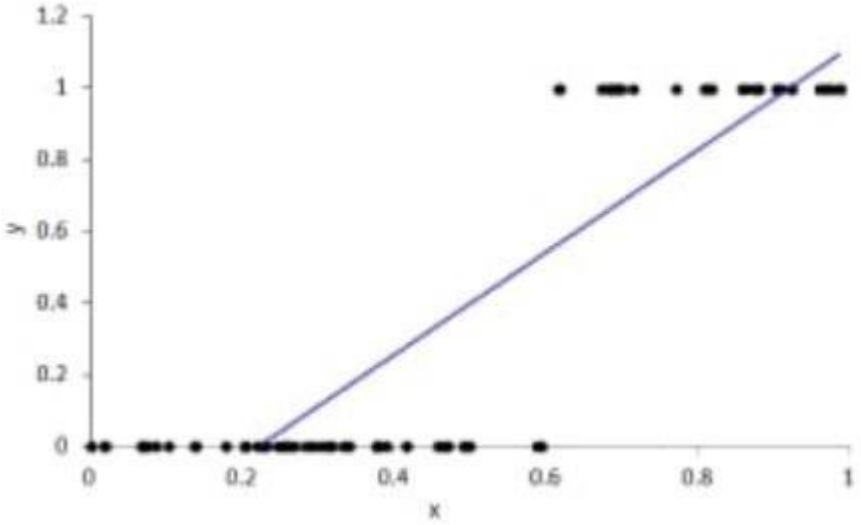
- Odds of event = Y / (1-Y)

P=ODDS/(1+ODDS)

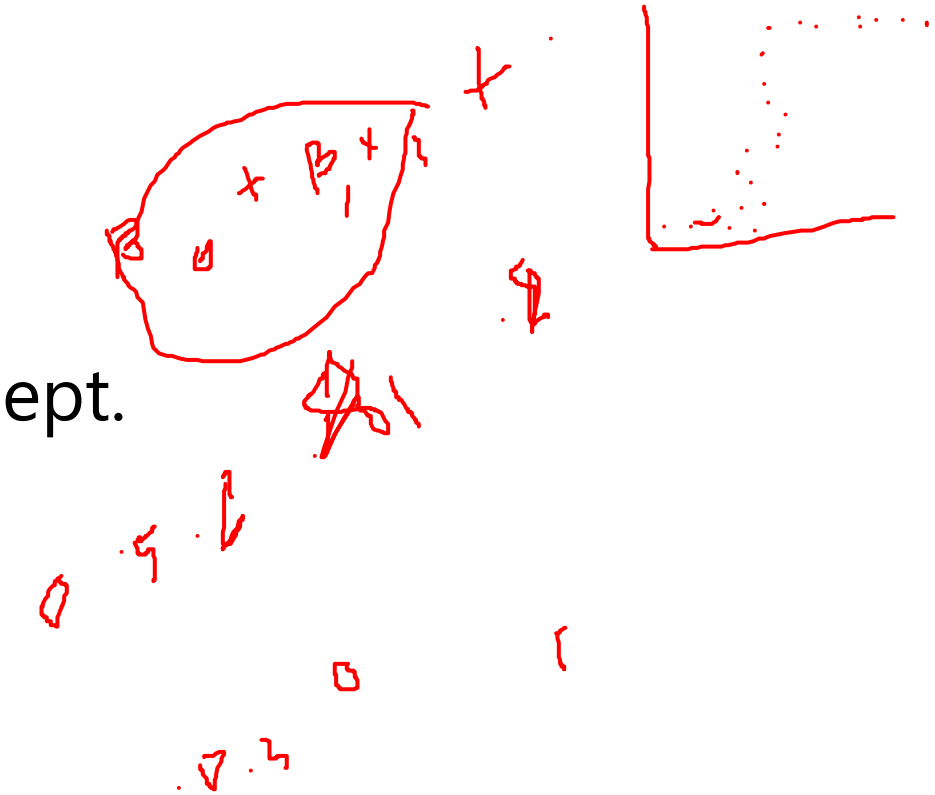
ODDS=Probability of success/ 1-Probability of success

Logistic Regression

- In logistic regression, we are only concerned about the probability of outcome dependent variable (success or failure).
- So we are using two functions ,a **link function** based on linear regression model and a logit function.
- alpha as $\beta 0+\beta 1X_1+\beta 1X_2$
- alpha is the link function.
- Log(p/1-p) is the log of odd ratio and is called the logit function.



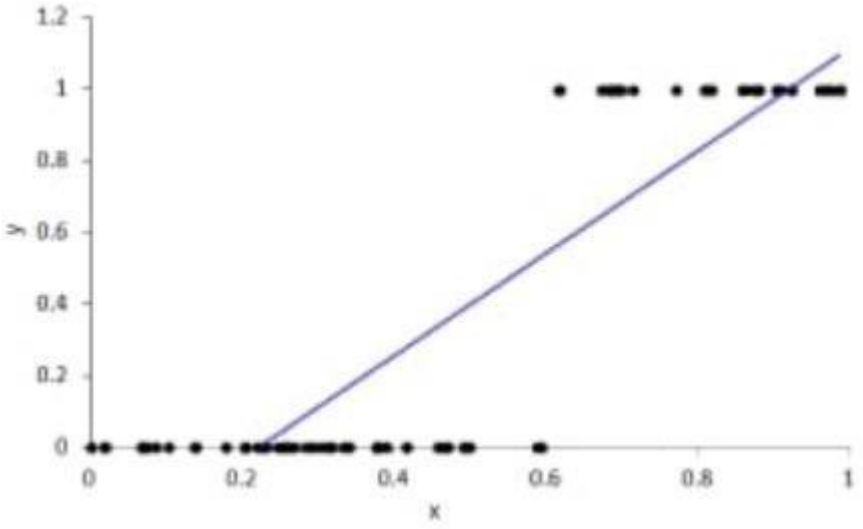
Logistic Regression



- Probability should always be positive
- Probability must be less than 1
- Where β1 Coefficient and β0 is the intercept.
- The expression can be rewritten as:
- Therefore : $p = \frac{e^{alpha}}{(1+e^{alpha})}$
- $odd = \frac{e^{alpha}}{(1/1+e^{alpha})}$
- $\log(P/1-P) = alpha$
- $P(Y|X) = 1/(1+e^{-\beta 0+\beta 1X})$

Logistic Regression

- Logistic Regression predicts binary output based on predictors or attributes. The probability that the output is 1 given a set of p features is represented as; $P(y=1|x)= \beta_o + \beta_1 x_1 + \ldots + \beta_p x_p$
- GLM does not assume a linear relationship between dependent and independent variables. Hence the output of a linear regression model, fit to binary data yields vague results.
- The generalized linear model (GLM) is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution.
- The regression algorithm could fit an equation to the data however the output would be vague, since the value of the linear predictor would range from $-\infty$ to ∞ and the probability value is in the range of 0 to 1.



Odds and log-odds

Odds is the ratio of probability of success to the probability of failure.

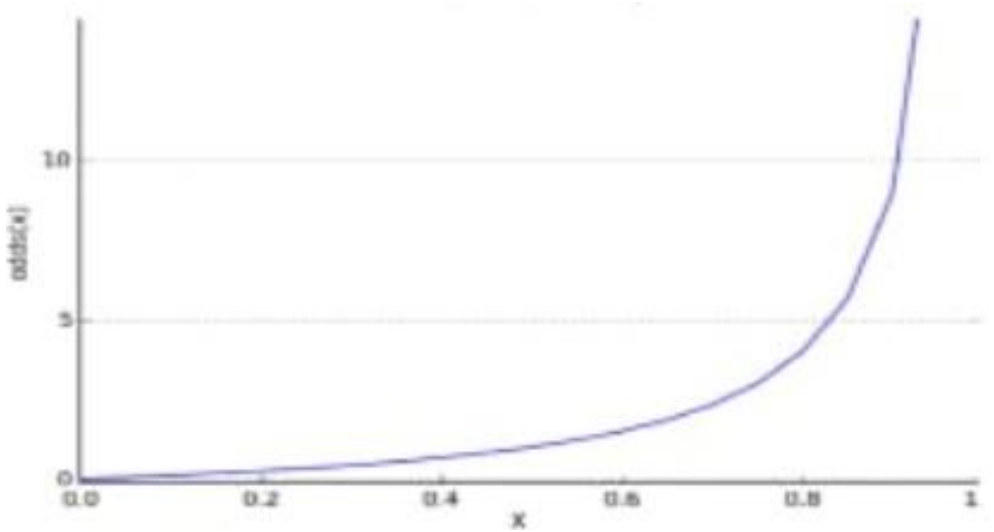
$$odds(p) = p / 1-p$$

Range of odds can be any number between 0 and ∞.
This is better than probability and one step closer to match the range of RHS of GLM.

Taking the log of odds gives the range from $-\infty$ to ∞ on LHS of the equation

$$\ln(o) = \beta_o + \beta_1 x_1 + \ldots + \beta_p x_p$$

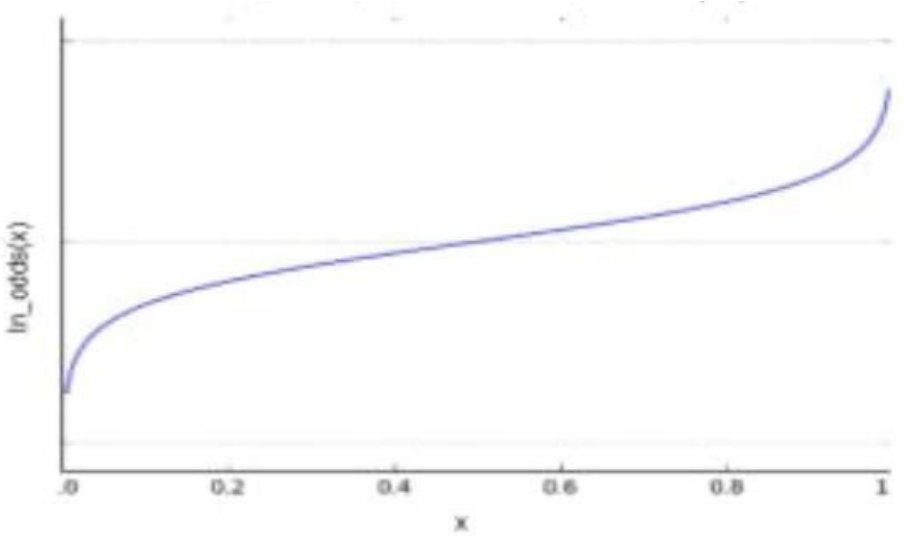
Where o represents odds.



Logistic function

Hence we have a regression model, where the output is log of the odds, also known as logit or logistic function. This is the link function of logistic regression.

Sigmoid curve
Take the inverse of log odds function,

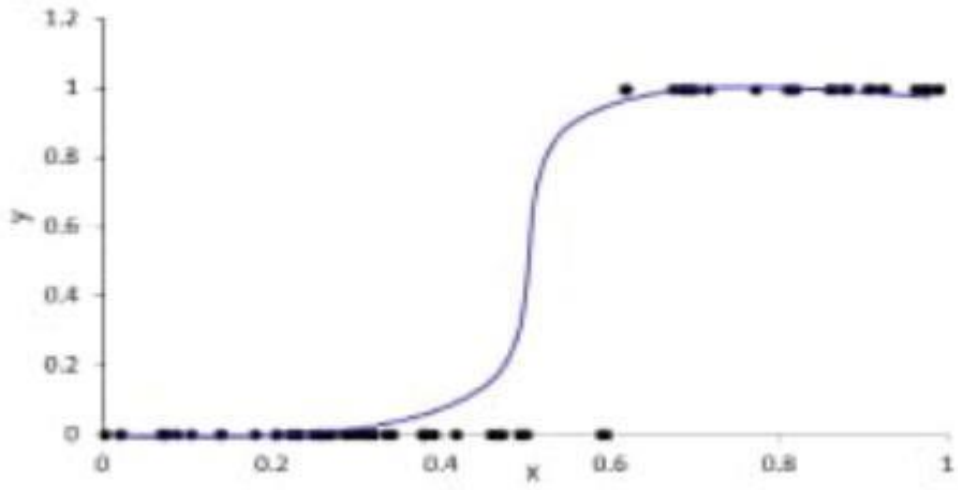


Solving for p

$$ln(o) = \beta_o + \beta_i x_i$$
$$p/(1-p) = exp(\beta_o + \beta_i x_i)$$

The inverse of the logistic function is called sigmoid function.

$$p = \frac{1}{1 + exp-(\beta_o + \beta_i x_i)}$$



Types of Logistic regression

- ***Binomial:***

In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

- ***Multinomial:***

In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep".

- ***Ordinal:***

In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

Case study carseats

- install.packages("ISLR")
- library(ISLR)
- df=Carseats> head(Carseats)

	Sales	CompPrice	Income	Advertising	Population	Price	ShelveLoc	Age	Education	Urban	US
1	9.50	138	73	11	276	120	Bad	42	17	Yes	Yes
2	11.22	111	48	16	260	83	Good	65	10	Yes	Yes
3	10.06	113	35	10	269	80	Medium	59	12	Yes	Yes
4	7.40	117	100	4	466	97	Medium	55	14	Yes	Yes
5	4.15	141	64	3	340	128	Bad	38	13	Yes	No
6	10.81	124	113	13	501	72	Bad	78	16	No	Yes

Case study carseats

```
> df=Carseats
> str(df)
'data.frame':   400 obs. of  11 variables:
 $ Sales       : num  9.5 11.22 10.06 7.4 4.15 ...
 $ CompPrice   : num  138 111 113 117 141 124 115 136 132 132 ...
 $ Income      : num  73 48 35 100 64 113 105 81 110 113 ...
 $ Advertising: num  11 16 10 4 3 13 0 15 0 0 ...
 $ Population : num  276 260 269 466 340 501 45 425 108 131 ...
 $ Price       : num  120 83 80 97 128 72 108 120 124 124 ...
 $ ShelveLoc   : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3 ...
 $ Age         : num  42 65 59 55 38 78 71 67 76 76 ...
 $ Education   : num  17 10 12 14 13 16 15 10 10 17 ...
 $ Urban       : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...
 $ US          : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...
```

Converting yes no values to 0 and 1

- library(plyr)
- df$Urban
- df$Urban <- revalue(df$Urban, c("Yes"=1))
- df$Urban <- revalue(df$Urban, c("No"=0))

```
> df$Urban
  [1] Yes Yes Yes Yes Yes No  Yes Yes No  No  No  Yes Yes Yes Yes No  Yes Yes No  Yes Yes No
 [23] Yes Yes Yes No  No  Yes Yes Yes Yes Yes No  Yes Yes No  No  Yes Yes No  No  Yes Yes Yes
 [45] Yes Yes No  Yes Yes Yes Yes Yes Yes Yes No  Yes Yes Yes Yes Yes Yes No  Yes Yes No  No
 [67] Yes Yes Yes Yes Yes No  Yes No  No  No  Yes No  Yes Yes Yes Yes Yes Yes No  No  Yes No
 [89] Yes No  No  Yes Yes Yes Yes Yes No  Yes No  No  No  Yes No  Yes Yes Yes No  Yes Yes No
[111] Yes Yes Yes Yes Yes Yes No  Yes Yes Yes Yes Yes Yes No  Yes No  Yes Yes Yes No  Yes Yes
[133] Yes Yes Yes No  No  Yes Yes No  Yes Yes Yes Yes No  Yes Yes No  No  Yes No  No  No  No
[155] No  Yes Yes No  No  No  No  No  Yes No  No  Yes Yes Yes Yes Yes Yes Yes Yes Yes No  Yes
[177] No  Yes No  Yes Yes Yes Yes Yes No  Yes No  Yes Yes No  No  Yes No  Yes Yes Yes Yes Yes
[199] Yes Yes No  Yes No  Yes Yes Yes Yes No  Yes No  No  Yes Yes Yes Yes Yes Yes No  Yes Yes
[221] Yes Yes Yes Yes No  Yes Yes Yes No  No  No  No  Yes No  No  Yes Yes Yes Yes Yes Yes Yes
[243] No  Yes Yes No  Yes Yes Yes Yes Yes Yes Yes No  Yes Yes Yes Yes No  No  Yes Yes Yes Yes
[265] Yes Yes No  No  Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes No  Yes Yes No  Yes No  No  Yes
[287] No  Yes No  Yes No  Yes Yes Yes Yes No  Yes Yes Yes No  Yes Yes Yes Yes Yes Yes Yes Yes
[309] Yes Yes Yes Yes Yes Yes Yes Yes Yes No  No  No  Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes
[331] No  Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes No  No  Yes No  Yes No  No  Yes No  No  No
[353] Yes No  Yes Yes Yes Yes Yes Yes No  No  Yes Yes Yes No  No  Yes No  Yes Yes Yes No  Yes
[375] Yes Yes Yes No  Yes Yes Yes Yes Yes Yes Yes Yes Yes No  Yes Yes Yes Yes Yes No  Yes Yes
[397] No  Yes Yes Yes
Levels: No Yes
```

```
> df$Urban
  [1] 1 1 1 1 1 0 1 1 0 0 0 1 1 1 1 0 1 1 0 1 1 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 0 1 1 0 0 1 1 1 1
 [46] 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 1 1 1 1 1 0 0 1 0 1 0
 [91] 0 1 1 1 1 1 1 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1
[136] 0 0 1 1 0 1 1 1 1 1 0 1 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1
[181] 1 1 1 1 1 0 1 0 1 1 0 0 1 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0
[226] 1 1 1 1 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1
[271] 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[316] 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 1 1 1
[361] 0 0 1 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1
Levels: 0 1
```

Converting yes no values to 0 and 1 in US

- df$US=ifelse(df$US=="Yes",1,0)
- df$US=as.factor(df$US)
- s=sample(nrow(df),.8*nrow(df))
- df_tr=df[s,]
- df_test=df[-s,]

Converting yes no values to 0 and 1 in US

- logitmod=glm(US~Advertising+Population,data=df_tr,family=binomial,control=list(maxit=100))
- summary(logitmod)

- logitmod1=glm(US~.,data=df_tr,family=binomial,
- control=list(maxit=100))
- summary(logitmod1)

Summary of logistic regression

```
Call:
glm(formula = US ~ ., family = binomial, data = df_tr, control = list(maxit = 100))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.18449  -0.37619   0.00164   0.05037   2.41402

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)     -3.637462   3.038864  -1.197  0.23131
Sales            0.064185   0.214262   0.300  0.76451
CompPrice        0.015363   0.027582   0.557  0.57754
Income           0.014303   0.008835   1.619  0.10547
Advertising      1.070820   0.159651   6.707 1.98e-11 ***
Population      -0.005714   0.001887  -3.028  0.00246 **
Price           -0.001168   0.023540  -0.050  0.96044
ShelveLocGood   -0.345150   1.170703  -0.295  0.76813
ShelveLocMedium -1.251007   0.685190  -1.826  0.06788 .
Age              0.010638   0.017462   0.609  0.54237
Education        0.030983   0.088339   0.351  0.72579
UrbanYes        -0.814396   0.492598  -1.653  0.09828 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 417.96  on 319  degrees of freedom
Residual deviance: 134.10  on 308  degrees of freedom
AIC: 158.1
```

prediction

```
#prediction
p1=predict(logitmod,df_test,type="response")
p1
pred1=ifelse(p1>.8,1,0)
```

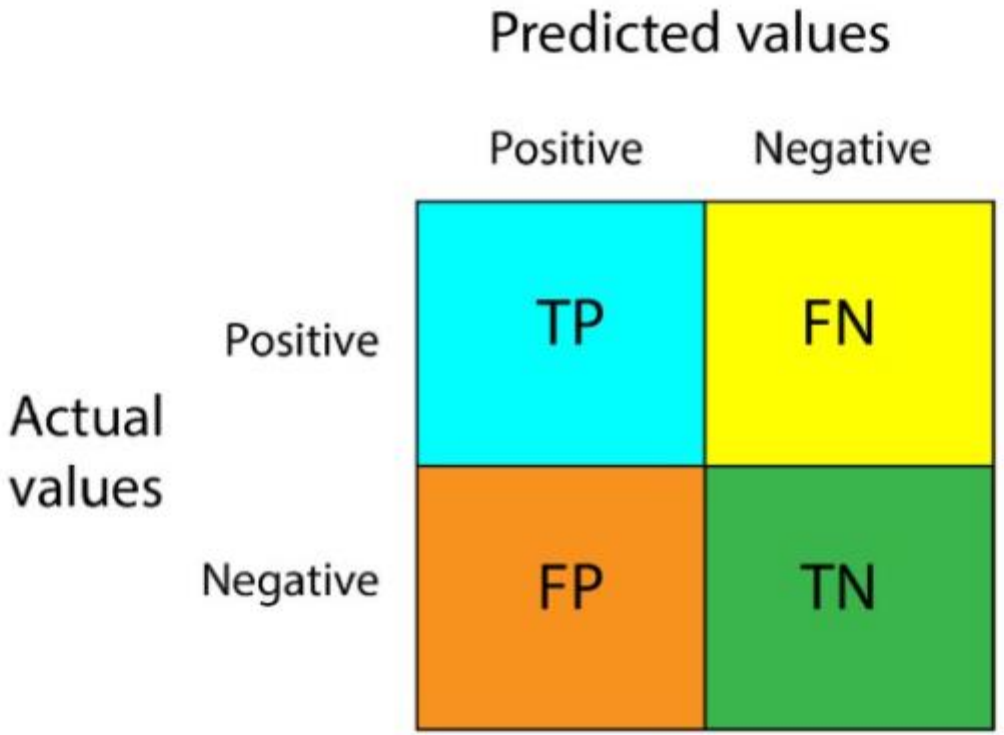
Confusion matrix

```
> table(df_test$US,pred1)
   pred1
     0  1
  0 20  5
  1 12 43

> (20+43)/nrow(df_test)*100
[1] 78.75
```

Confusion matrix

What is Confusion Matrix and why you need it?
Well, it is a performance measurement for machine learning classification problem where output can be two or more classes.
It is a table with 4 different combinations of predicted and actual values.


Predicted values
Positive
Negative
Actual values
Positive
Negative
TP
FN
FP
TN

Understanding True Positive, True Negative, False Positive and False Negative in a Confusion Matrix

True Positive (TP)
The predicted value matches the actual value
The actual value was positive and the model predicted a positive value
True Negative (TN)
The predicted value matches the actual value
The actual value was negative and the model predicted a negative value

False Positive (FP) – Type 1 error

The predicted value was falsely predicted
The actual value was negative but the model predicted a positive value
Also known as the Type 1 error

False Negative (FN) – Type 2 error
The predicted value was falsely predicted
The actual value was positive but the model predicted a negative value
Also known as the Type 2 error

Precision vs. Recall

> *Precision tells us how many of the correctly predicted cases actually turned out to be positive.*

Here's how to calculate Precision:

$$Precision = \frac{TP}{TP + FP}$$

This would determine whether our model is reliable or not.

> *Recall tells us how many of the actual positive cases we were able to predict correctly with our model.*

And here's how we can calculate Recall:

$$\text{Recall} = \frac{tp}{tp + fn}$$

Harmonic Mean Definition

- The Harmonic Mean (HM) is defined as the reciprocal of the arithmetic mean of the reciprocals of the observations.
- Harmonic mean gives less weightage to the larger values and more weightage to the smaller values to balance the values properly.
- The harmonic mean is generally used when there is a necessity to give greater weight to the smaller items.
- The harmonic mean is often used to calculate the average of the ratios or rates of the given values.
- It is the most appropriate measure for ratios and rates because it equalizes the weights of each data poi

$$\text{Harmonic Mean}(H) = n / [(1/x_1)+(1/x_2)+(1/x_3)+\ldots+(1/x_n)]$$

F1-Score

In practice, when we try to increase the precision of our model, the recall goes down, and vice-versa. The F1-score captures both the trends in a single value:

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

F1-score is a harmonic mean of Precision and Recall, and so it gives a combined idea about these two metrics. It is maximum when Precision is equal to Recall.

Specificity

True negative rate is also called specificity.

$$\text{True negative rate} = \frac{tn}{tn + fp}$$

Sensitivity

Recall in this context is also referred to as the true positive rate or sensitivity

$$\text{Recall} = \frac{tp}{tp + fn}$$

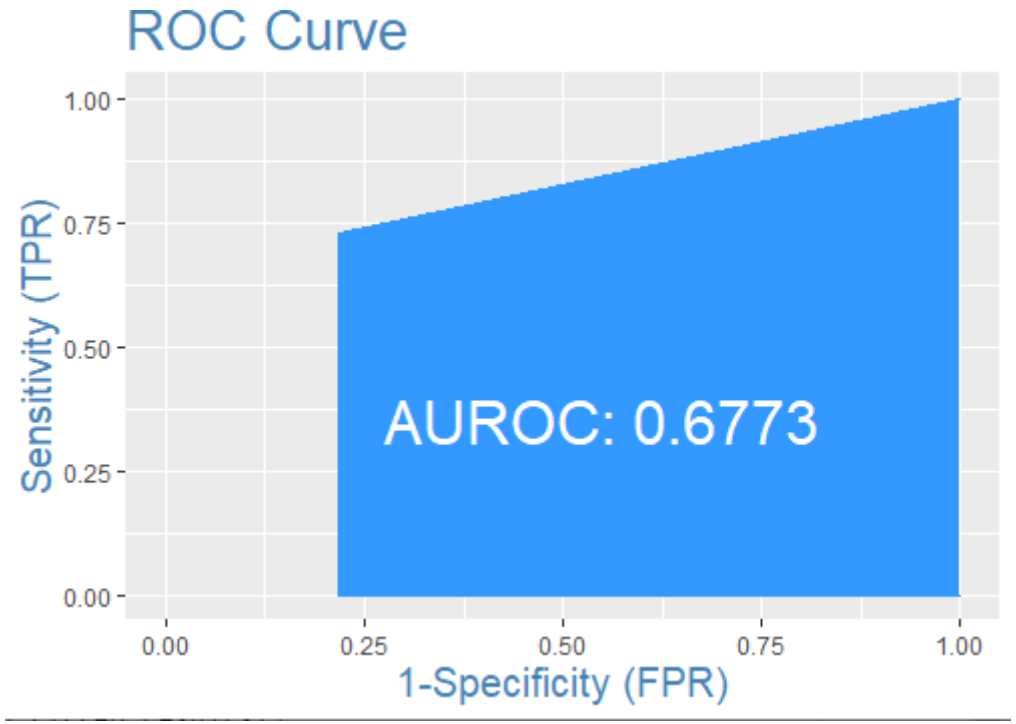
ROC Curve

- The receiving operating characteristic is a measure of classifier performance.
- Using the proportion of positive data points that are correctly considered as positive(TP) and the proportion of negative data points that are mistakenly considered as positive(FP),
- we generate a graphic that shows the trade off between the rate at which you can correctly predict something with the rate of incorrectly predicting something.
- Ultimately, we're concserned about the area under the ROC curve, or AUROC.
- That metric ranges from 0.50 to 1.00, and values above 0.80 indicate that the model does a good job in discriminating between the two categories which comprise our target variable

ROC

```
> plotROC(df.test$fraud_reported,fitted.results1)
```

- AUROC with 1 indicates a perfect model.
- AUROC has reduced from 0.6797 to 0.6773 in the third model.
- Misclassification rate is lowest in first model at 21.33%.
- AUROC is highest in second model at 0.6797
- With the models, False Positives are reducing and True positives are increasing but False negatives are increasing.
- Depending on the purpose of the institution using the model, trade-off between the values in confusion matrix can be examined and appropriate model be chosen.
- The first model has very slightly lower AUROC score which can be let go, first model can be considered the better model.



With AUROC value 0.6773, it is considered to be an acceptable discrimination, not very different from a coin toss.

Thank You