**Classification model evaluation**

**Confusion Matrix:**

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.

[](http://blog.exsilio.com/wp-content/uploads/2016/09/table-blog.png)

It is extremely useful for measuring:’

* Recall,
* Precision,
* Specificity,
* Accuracy and
* most importantly AUC-ROC Curve, F-Measure, …...

**Let’s understand TP, FP, FN, TN**

**True Positives (TP)** - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

**True Negatives (TN)** - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

*True positives and Ture negatives, these values occur when your actual class matches with the predicted class.*

*False positives and false negatives, these values occur when your actual class contradicts (mismatches) with the predicted class.*

**False Positives (FP) (Type 1 Error)** – When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

**False Negatives (FN) (Type 2 Error)** – When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

***Once you understand these four parameters then we can calculate Accuracy, Precision, Recall and F1 score…..***

**Accuracy** - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best.

**Accuracy = (TP+TN) / (TP+FP+FN+TN)**



**Accuracy:** Overall, how often is the classifier correct?

* + (TP+TN)/total = (100+50)/165 = 0.91

**Misclassification Rate:** Overall, how often is it wrong?

* + (FP+FN)/total = (10+5)/165 = 0.09
  + equivalent to 1 minus Accuracy
  + also known as "Error Rate"
* **True Positive Rate:** When it's actually yes, how often does it predict yes?
  + TP/total-actual yes = 100/105 = 0.95
  + also known as "Sensitivity" or "Recall"
* **True Negative Rate:** When it's actually no, how often does it predict no?
  + TN/actual no = 50/60 = 0.83
  + equivalent to 1 minus False Positive Rate
  + also known as "Specificity"
* **False Positive Rate:** When it's actually no, how often does it predict yes?
  + FP/actual no = 10/60 = 0.17
* **False Negative Rate:** When it's actually yes, how often does it predict no?
  + FN/actual Yes = 5/105 = 0.17

**Precision**  - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate.

Precision = TP/(TP+FP)

**Precision:** When it predicts yes, how often is it correct?

* TP/total\_predicted\_yes = 100/110 = 0.91

**Prevalence:** How often does the yes condition actually occur in our sample?

= (TP+FN)/( TP+FP+FN+TN)

* actual yes/total = 105/165 = 0.64

**Recall**(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

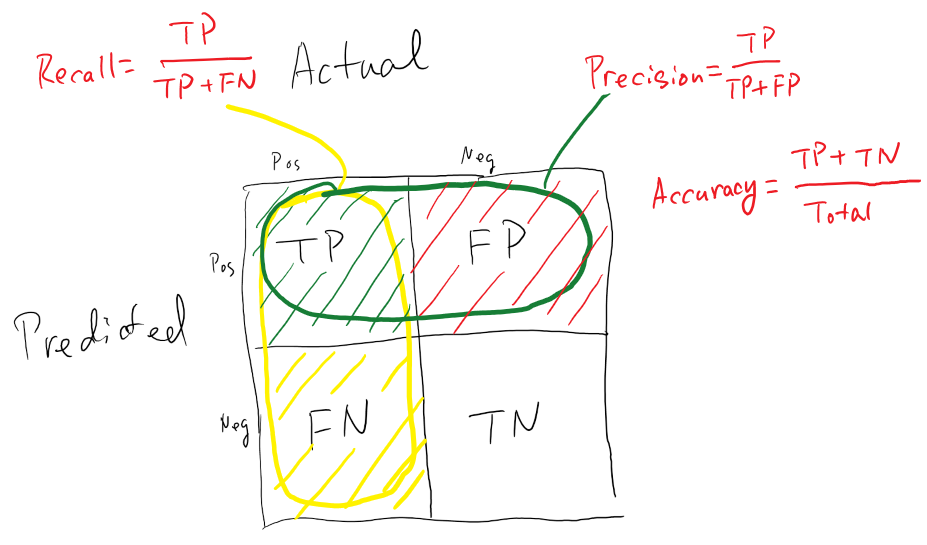
*The question recall answers is: Of all the passengers that truly survived, how many did we label?*

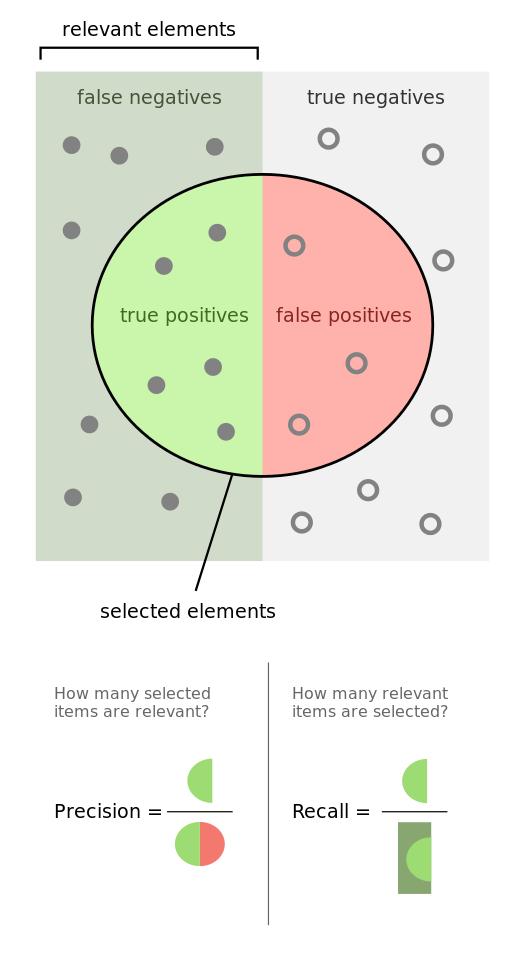
Recall= TP/ total\_actual\_yes

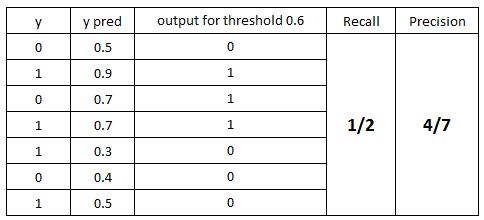
Recall = TP/(TP+FN)

=100/105

**Recall (**also called the **true positive rate)** measures the proportion of actual positives which are correctly identified as such positive.







?

**Specificity** (also called the **true negative rate**) measures the proportion of negatives which are correctly identified as such (e.g., the percentage of healthy people who are correctly identified as not having the condition), and is complementary to the [false positive rate](http://en.wikipedia.org/wiki/False_positive_rate).

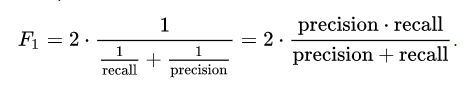
*Specificity=true negatives/(true negative + false positives)*

**F1 score** - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

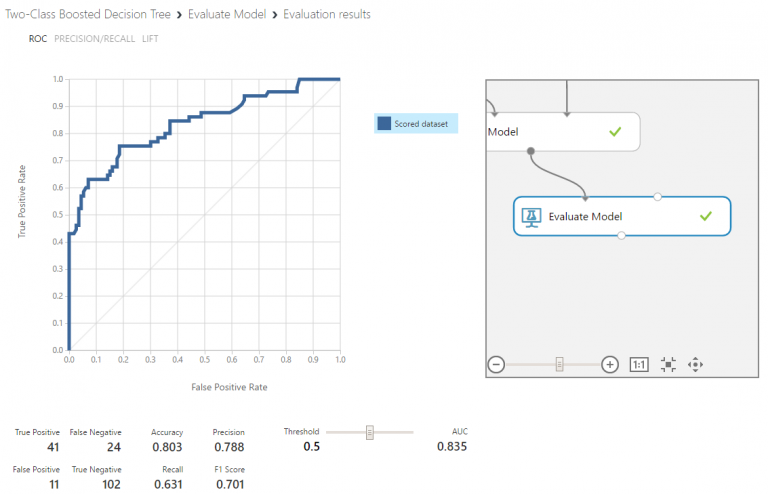
* F1 is usually more useful than accuracy, especially if you have an uneven class distribution.
* Accuracy works best if false positives and false negatives have similar cost.
* If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)

The traditional F-measure or balanced F-score (**F1 score**) is the [harmonic mean](https://en.wikipedia.org/wiki/Harmonic_mean#Harmonic_mean_of_two_numbers) of precision and recall:



Precision/Recall/F1 Measure



The first thing you will see here is ROC curve and we can determine whether our ROC curve is good or not by looking at AUC (Area Under the Curve) and other parameters which are also called as Confusion Metrics. A confusion matrix 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. All the measures except AUC can be calculated by using left most four parameters. So, let’s talk about those four parameters first.

[](http://blog.exsilio.com/wp-content/uploads/2016/09/table-blog.png)

True positive and true negatives are the observations that are correctly predicted and therefore shown in green. We want to minimize false positives and false negatives so they are shown in red color. These terms are a bit confusing. So let’s take each term one by one and understand it fully.

**True Positives (TP)** - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

**True Negatives (TN)** - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

**False Positives (FP)** – When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

**False Negatives (FN)** – When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

**Once you understand these four parameters then we can calculate Accuracy, Precision, Recall and F1 score.**

**Accuracy** - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model. For our model, we have got 0.803 which means our model is approx. 80% accurate.

Accuracy = TP+TN/TP+FP+FN+TN

**Precision**  - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Precision = TP/TP+FP

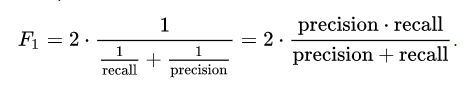
**Recall**(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes. The question recall answers is: Of all the passengers that truly survived, how many did we label? We have got recall of 0.631 which is good for this model as it’s above 0.5.

Recall = TP/TP+FN

**F1 score** - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall. In our case, F1 score is 0.701.

F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)

The traditional F-measure or balanced F-score (**F1 score**) is the [harmonic mean](https://en.wikipedia.org/wiki/Harmonic_mean#Harmonic_mean_of_two_numbers) of precision and recall:



So, whenever you build a model, this article should help you to figure out what these parameters mean and how good your model has performed.

I hope you found this blog useful. Please leave comments or send me an email if you think I missed any important details or if you have any other questions or feedback about this topic.

**Sensitivity** and **specificity** are statistical measures of the performance of a [binary classification](http://en.wikipedia.org/wiki/Binary_classification) [test](http://en.wikipedia.org/wiki/Classification_rule), also known in statistics as [classification function](http://en.wikipedia.org/wiki/Statistical_classification):

**Sensitivity** (also called the **true positive rate**, or the [**recall**](http://en.wikipedia.org/wiki/Precision_and_recall#Definition_.28classification_context.29) in some fields) measures the proportion of actual positives which are correctly identified as such (e.g., the percentage of sick people who are correctly identified as having the condition), and is [complementary](http://en.wikipedia.org/wiki/Complementary_event) to the [false negative rate](http://en.wikipedia.org/wiki/False_negative_rate).

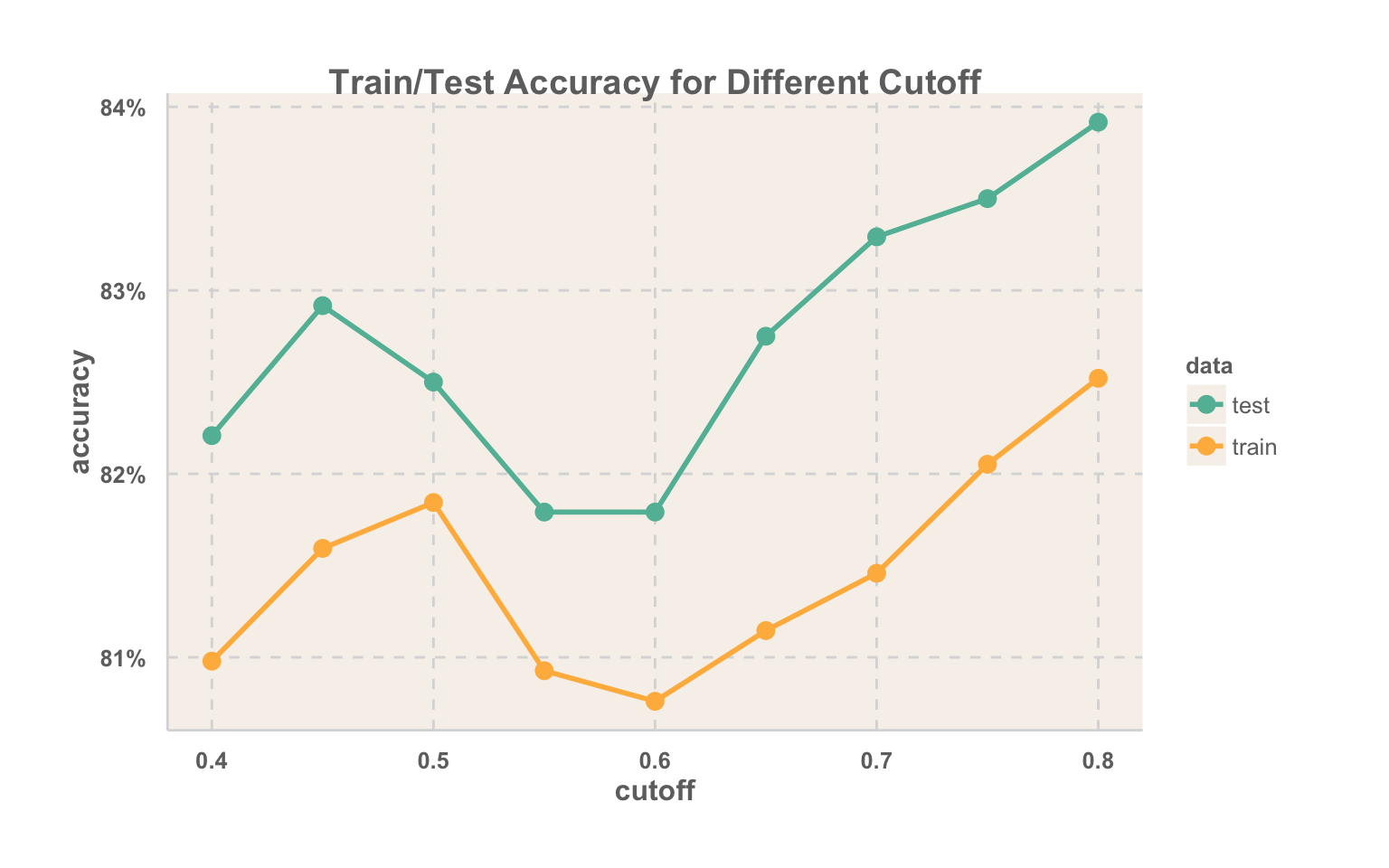
*Sensitivity= true positives/(true positive + false negative)*

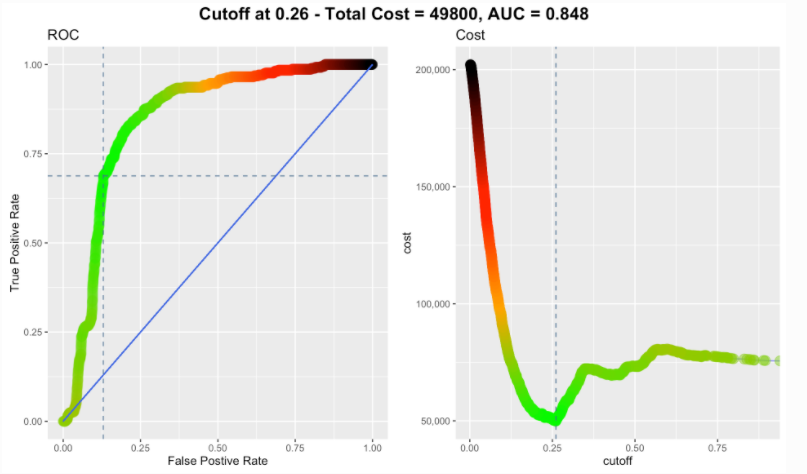
**Specificity** (also called the **true negative rate**) measures the proportion of negatives which are correctly identified as such (e.g., the percentage of healthy people who are correctly identified as not having the condition), and is complementary to the [false positive rate](http://en.wikipedia.org/wiki/False_positive_rate).

*Specificity=true negatives/(true negative + false positives)*

For any test, there is usually a trade-off between the measures.

Now if your model is a good model, one way to check is to have a high sensitivity and specificity. There is a trade off in accuracy levels and it depends on the business situation.





False positive rate (FPR) and false negative rate (FNR), you can think of this as the objective function for our model, where we’re trying to minimize the number of mistakes we’re making or so called to cost.

**Result interpretation and performance measures of *model***

(Logistic Regression)

* Null Deviance
* Residual Deviance
* LR-Test
* AIC
* Fisher Score
* Pseudo R2
* Adj-Pseudo R2

glm result interpretation

# Load package "datasets":

library(datasets)

#Load data "mtcars":

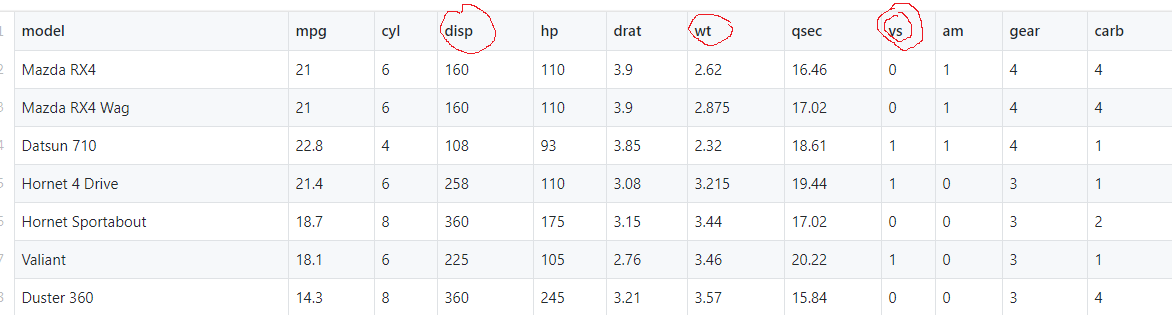
data(mtcars)

#Display info about dataset

dim(mtcars)

nrow(mtcars)

ncol(mtcars)



In the mtcars data set, the variable vs indicates if a car has a [V engine](http://en.wikipedia.org/wiki/V_engine) or a [straight engine](http://en.wikipedia.org/wiki/Straight_engine).

We want to create a model that helps us to predict the probability of a vehicle having a [V engine](http://en.wikipedia.org/wiki/V_engine) or a [straight engine](http://en.wikipedia.org/wiki/Straight_engine) given a weight of 2100 lbs and engine displacement of 180 cubic inches.

model <- glm(formula= vs ~ wt + disp,

data=mtcars, family=binomial)

We call the model **“binomial logistic regression” i.e.,** family=binomial, since the variable to predict is binary, however, logistic regression can also be used to predict a dependent variable which can assume more than 2 values.

summary(model)

Call:

glm(formula = vs ~ wt + disp, family = binomial, data = mtcars)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.67506 -0.28444 -0.08401 0.57281 2.08234

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 1.60859 2.43903 0.660 0.510

wt 1.62635 1.49068 1.091 0.275

disp -0.03443 0.01536 -2.241 0.025\*

---

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 43.86 on 31 degrees of freedom

Residual deviance: 21.40 on 29 degrees of freedom

AIC: 27.4

Number of Fisher Scoring iterations: 6

Result Interpretation

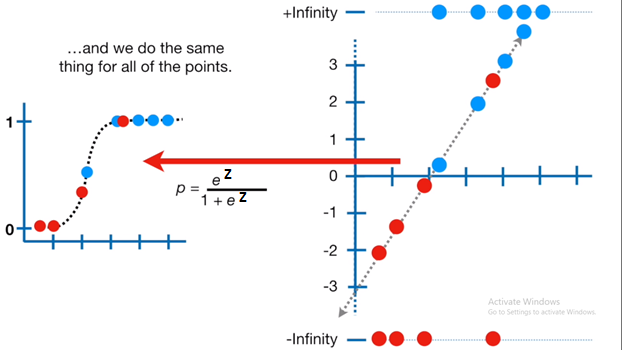
We also see that the coefficient of weight is non-significant (p > 0.05), while the coefficient of displacement is significant.

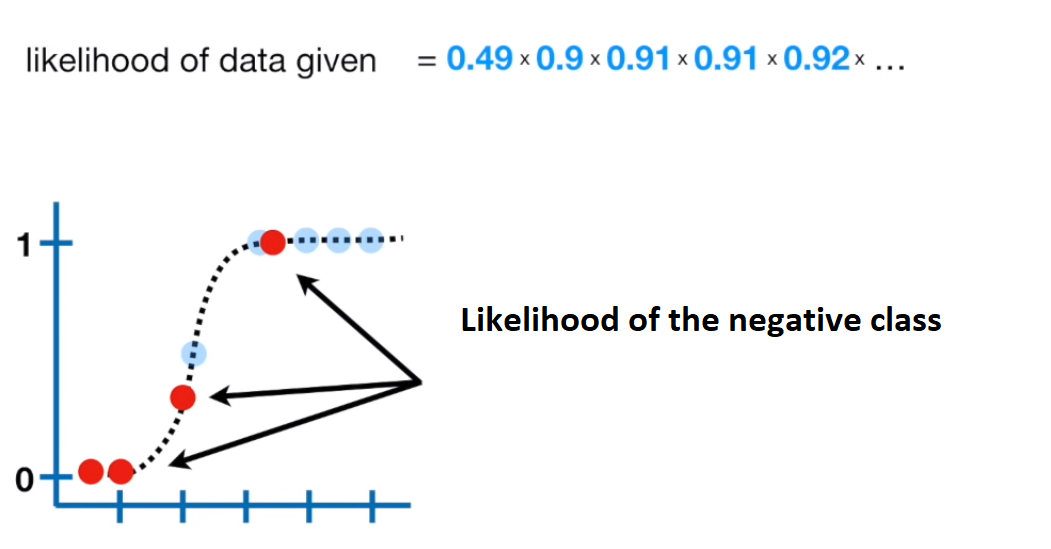
Deviance

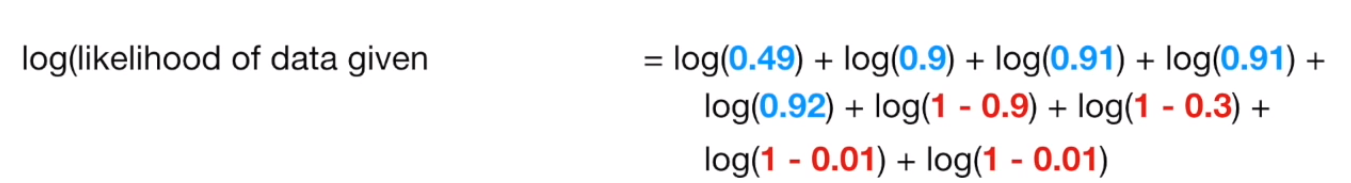
We see the word Deviance twice over in the model output. [Deviance](http://www.theanalysisfactor.com/concepts-you-need-to-understand-to-run-a-mixed-or-multilevel-model/) is a measure of goodness of fit of a generalized linear model. Lets understand deviance:

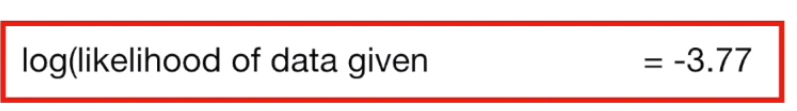
**Prerequisite- Likelihood Ratio (LR), i.e., LR-Test (goodness-of-fit)**

Let’s quick review of likelihood function-

****







**Continue LR ratio-**

* The lr test is performed by estimating two models and comparing the fit of one model to the fit of the other.
* Removing predictor variables from a model will almost always make the model fit less well (i.e., a model will have a lower log likelihood).
* Adding predictor variables from a model will almost always make the model fit more well (i.e., a model will have a higher log likelihood).
* But, it is necessary to test whether the observed difference in model fit is statistically significant.
* The lr test does this by comparing the log likelihoods of the two models, if this difference is statistically significant, then the less restrictive model (the one with more variables) is said to fit the data significantly better than the more restrictive model.
* If one has the log likelihoods from the models, the lr test is fairly easy to calculate. The formula for the lr test statistic is:

lr = -2 ln(L(m1)/L(m2)) = 2(ll(m2)-ll(m1))

lr = 2(ll(complex\_model)-ll(simple\_model))

* Where L(m\*) denotes the likelihood of the respective model (either model 1 or model 2), and
* ll(m\*) the natural log of the model’s final likelihood (i.e., the log likelihood).
* Where m1 is the more restrictive model, and m2 is the less restrictive model.

**Note:** The resulting test statistic is distributed chi-squared, with degrees of freedom equal to the number of parameters that are constrained (in the current example, the number of variables removed from the model, i.e., 2).

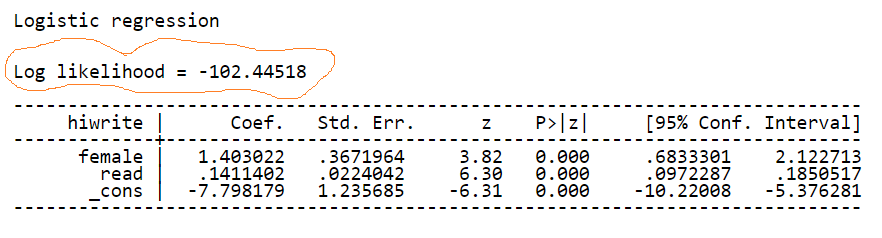
DF= DF (m1)- DF (m2), i.e., no. of extra features

Understand with example:

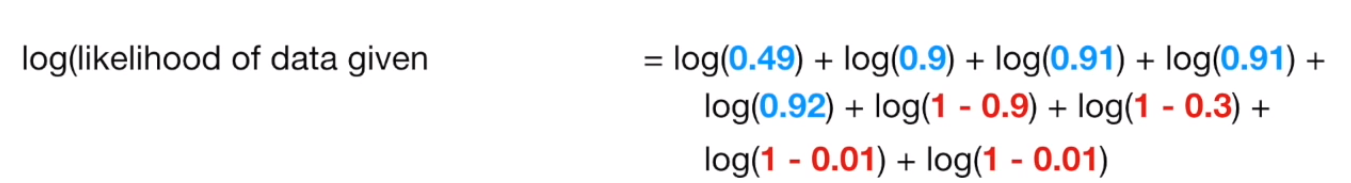
Suppose model (A) is= B0 + B1 female+ B2 read

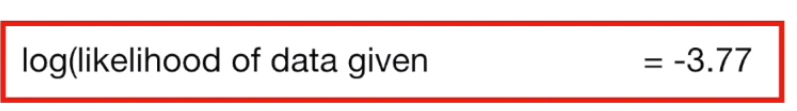
Output: handwriting is (good/bad)

Following is the logistic regression Model-A

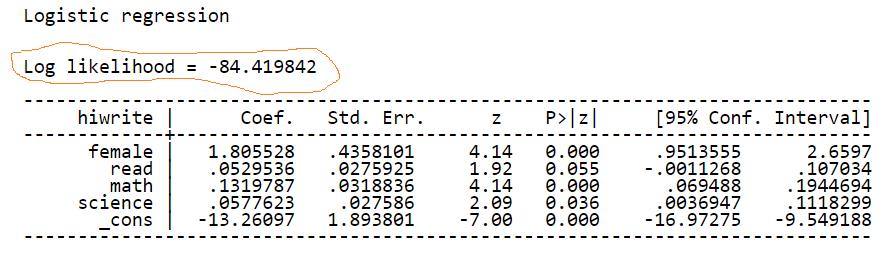


Log likelihood: as discussed in last class, i.e. summation of log of each observation’s probability. The probability comes from sigmoidal function against each observation.





Now we can run model B, in which coefficients for **science** and **math** are included, that is, a model with the full set of predictor variables. Below is output for model B.



Now that we have both log likelihoods, calculating the test statistic is simple:

LR = 2 \* (-84.419842 – (-102.44518)  )

= 2 \* (-84.419842 + 102.44518 ) =  36.050676

*So our likelihood ratio test statistic is 36.05 (distributed chi-squared), with two degrees of freedom.*

***Note:***

* *Larger value of LR test is desirable.*
* *Larger value means less restrictive model (complex) is significantly fit on data than more restrictive model (simpler).*
* *LR Test score used to calculate p-value.* Which indicating that the model with all four predictors fits significantly better than the model with only two predictors.

The p-value is P(χ2≥ΔG2).

χ2=Chi-square

ΔG2 =LR test ratio

* In LR test measure, one model must be ‘nested’ to other models, where one model is obtained from the other one by putting some of the parameters to be zero (i.e., drop those features).

Summary-

H0: reduced model is true vs. HA: current model is true

* the reduced model which omits the *k* predictors from current model, and
* the current model which includes them.

The likelihood-ratio statistic is

* ΔG2 = −2 log L from reduced model (simple) −(−2 log L from current model (complex))

and the degrees of freedom is k (the number of coefficients were dropped from complex model).

The p-value is P(χ2≥ΔG2).

------continue deviance

**Understanding-**

* Deviance measures the discrepancy between the current model and the full model.
* The full model (saturated model) is the model that has *n* parameters, one parameter per observation, i.e., number of coefficients in equation is 1+ number of data points.
* The full model maximizes the log-likelihood function.
* The full model provides a point of comparison for models with fewer than *n* parameters.

https://support.minitab.com/en-us/minitab-express/1/png/Analysis_of_deviance_binary_log_mf.dita_dctm_Chron0900045780316fa6_0.png

Two forms of deviance – the null deviance and the residual deviance.

* The null deviance shows how well the response variable is predicted by a model that includes only the intercept.

D= 2{ (LL\_Full\_Model) – (LL\_Null\_Model)}

**on df = df\_Full - df\_Null**

**D= 43.9, df=31**

For our example, we have a value of 43.9 on 31 degrees of freedom.

* The Residual Deviance (how well the response variable is predicted by a trained/proposed model), has reduced by 22.46 with a loss of two degrees of freedom.

Including the independent variables (weight and displacement) decreased the deviance to 21.4 points on 29 degrees of freedom, a significant reduction in deviance.

D= 2{ (LL\_Full\_Model) – (LL\_Current\_Model)

**D= 21.4, df=29**

Interpretation, current model is improved model against null model by 22.46 (deviance reduced, i.e. likelihood increased) with loss of 2df.

### Fisher Scoring

Fisher’s scoring algorithm is a derivative of Newton’s method for solving maximum likelihood problems numerically.

For model, we see that Fisher’s Scoring Algorithm needed six iterations to perform the fit.

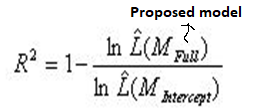
This doesn’t really tell you a lot that you need to know, other than the fact that the model did indeed converge, and had no trouble doing it.

### Information Criteria

* The Akaike Information Criterion (AIC) provides a method for assessing the quality of your model through comparison of related models.
* It’s based on the Deviance, but penalizes you for making the model more complicated.
* Much like adjusted R-squared, it’s intent is to prevent you from including irrelevant predictors.
* If you have more than one similar candidate models (where all of the variables of the simpler model occur in the more complex models), then you should select the model that has the smallest AIC.
* So it’s useful for comparing models, but isn’t interpretable on its own.

**Pseudo R^2**

* Unlike linear regression with ordinary least squares estimation, there is no R2 statistic which explains the proportion of variance in the dependent variable that is explained by the predictors.
* However, there are a number of pseudo R2 metrics that could be of value. Most notable is McFadden’s R2, which is defined as:

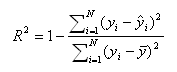


*Mfull*= Model with predictors

*Mintercept*= Model without predictors

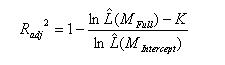
[Image lhat-1](https://stats.idre.ucla.edu/wp-content/uploads/2016/02/lhat-1.jpg)

McFadden’s approaches,  the log likelihood of the intercept model is treated as a total sum of squares (base model), and the log likelihood of the your model is treated as the sum of squared errors (fitted model). R2 with ordinary least squares (OLS) as:



* The measure (Pseudo R^2) ranges from 0 to just under 1,
* with values closer to zero indicating that the model has no predictive power.
* If comparing two models on the same data, McFadden’s would be higher for the model with the greater likelihood.

**McFadden’s (adjusted)- adjusted pseudo R^2**



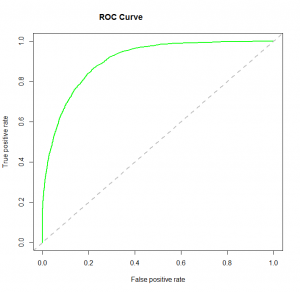
* McFadden’s adjusted mirrors the adjusted R-squared in OLS by penalizing a model for including too many predictors.
* If the predictors in the model are effective, then the penalty will be small relative to the added information of the predictors.
* However, if a model contains predictors that do not add sufficiently to the model, then the penalty becomes noticeable and the adjusted R-squared can *decrease* with the addition of a predictor, even if the R-squared increases slightly.
* Note that negative McFadden’s adjusted R-squared are possible.

**ROC Curve**

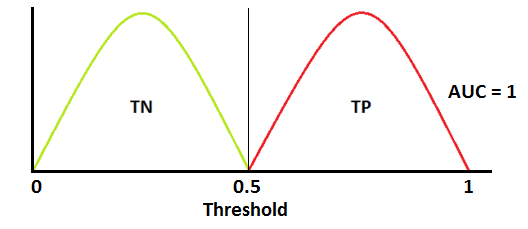
AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between patients with disease and no disease.

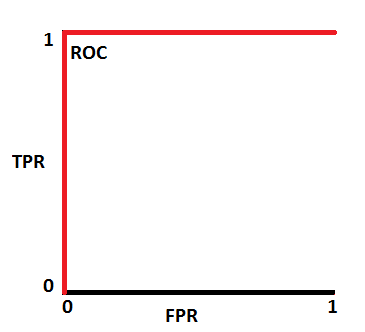
* Receiver Operating Characteristic(ROC) summarizes the model’s performance by evaluating the tradeoffs between true positive rate (sensitivity) and false positive rate(1- specificity).
* ROC summarizes the predictive power for all possible values of p(probability).
* The area under curve (AUC). Higher the area under curve, better the prediction power of the model. The ROC of a perfect predictive model has TP equals 1 and FP equals 0. This curve will touch the top left corner of the graph.



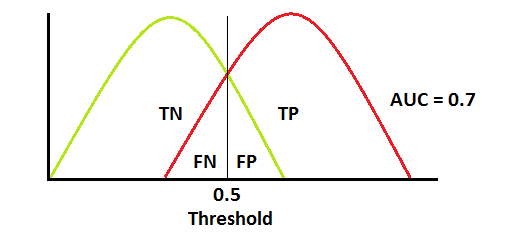
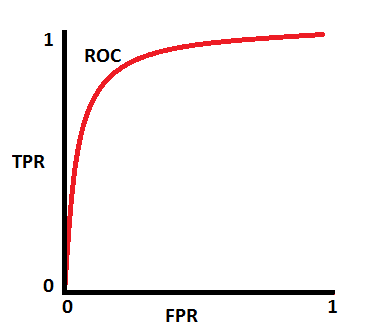
[](https://www.analyticsvidhya.com/wp-content/uploads/2015/11/logit_roc.png)

An excellent model has AUC near to the 1 which means it has good measure of separability. A poor model has AUC near to the 0 which means it has worst measure of separability. In fact it means it is reciprocating the result. It is predicting 0s as 1s and 1s as 0s. And when AUC is 0.5, it means model has no class separation capacity whatsoever.

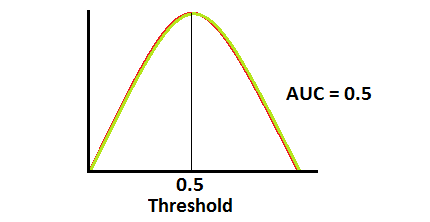
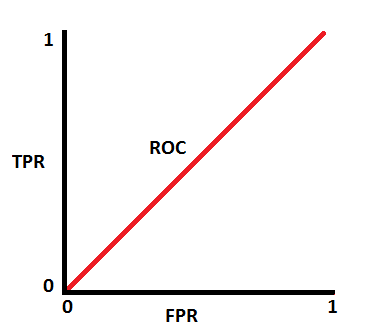




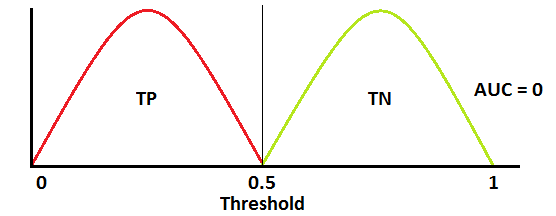
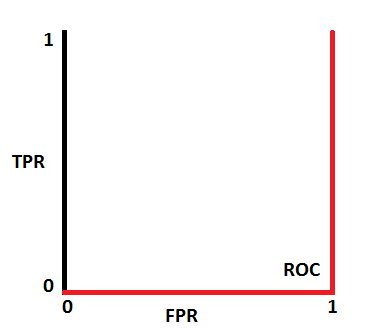
This is an ideal situation. When two curves don’t overlap at all means model has an ideal measure of separability. It is perfectly able to distinguish between positive class and negative class.

When two distributions overlap, we introduce type 1 and type 2 error. Depending upon the threshold, we can minimize or maximize them. When AUC is 0.7, it means there is 70% chance that model will be able to distinguish between positive class and negative class.

This is the worst situation. When AUC is approximately 0.5, model has no discrimination capacity to distinguish between positive class and negative class.

When AUC is approximately 0, model is actually reciprocating the classes. It means, model is predicting negative class as a positive class and vice versa.