Data: <https://stats.idre.ucla.edu/stat/data/binary.csv>

**Importance of Area under Curve and Concordant**  
Area under Curve (AUC) or Receiver operating characteristic (ROC) curve is used to evaluate and compare the performance of binary classification model. It measures **discrimination power of your predictive classification model**. In simple words, it checks how well model is able to distinguish (separates) events and non-events. Suppose you are building a predictive model for bank to identify customers who are likely to buy credit card. In this case case, purchase of credit card is event (or desired outcome) and non-purchase of credit card is non-event. 

*AUC or ROC curve is a plot of the proportion of true positives (events predicted to be events) versus the proportion of false positives (nonevents predicted to be events). True Positive Rate is also called Sensitivity. False Positive Rate is also called (1-Specificity). Sensitivity is on Y-axis and (1-Specificity) is on X-axis. Higher the AUC score, better the model.  
  
Diagonal line represents random classification model. It is equivalent to prediction by tossing a coin. All points along the diagonal line say same true positive and false positive rate.*

|  |
| --- |
| <https://2.bp.blogspot.com/-jxahDxn7uqk/XOLdlR1EOlI/AAAAAAAAHjQ/opxB1AyxhXkbP6XoXailoC89SnAp8z1XwCLcBGAs/s1600/roc_curve.png> |
| ROC Curve |

**Manual Calculation to estimate ROC, Concordant, Discordant, Gini**

1. Calculate the predicted probability in logistic regression model. It can be any binary classification model, not restricted to logistic regression.
2. Divide the data into two datasets. One dataset contains observations having actual value of dependent variable with value 1 (i.e. event) and corresponding predicted probability values. And the other dataset contains observations having actual value of dependent variable 0 (non-event) against their predicted probability scores.
3. Compare each predicted value in first dataset with each predicted value in second dataset.

*Total Number of pairs to compare = x \* y  
x : Number of observations in first dataset (actual values of 1 in dependent variable)  
y : Number of observations in second dataset (actual values of 0 in dependent variable).   
  
In this step, we are performing****cartesian product (cross join) of events and non-events****. For example, you have 100 events and 1000 non-events. It would create 100k (100\*1000) pairs for comparison.*

1. A pair is concordant if 1 (observation with the desired outcome i.e. event) has a higher predicted probability than 0 (observation without the outcome i.e. non-event).
2. A pair is discordant if 0 (observation without the desired outcome i.e. non-event) has a higher predicted probability than 1 (observation with the outcome i.e. event).
3. A pair is tied if 1 (observation with the desired outcome i.e. event) has same predicted probability than 0 (observation without the outcome i.e. non-event).
4. The final percent values are calculated using the formula below -

Percent Concordant = 100\*[(Number of concordant pairs)/Total number of pairs]  
Percent Discordant = 100\*[(Number of discordant pairs)/Total number of pairs]  
Percent Tied = 100\*[(Number of tied pairs)/Total number of pairs]  
Area under curve (c statistics) = (Percent Concordant + 0.5 \* Percent Tied)/100

**Interpretation of Concordant, Discordant and Tied Percent**  
  
**Percent Concordant :** Percentage of pairs where the observation with the desired outcome (event) has a higher predicted probability than the observation without the outcome (non-event).  
 **Percent Discordant :** Percentage of pairs where the observation with the desired outcome (event) has a lower predicted probability than the observation without the outcome (non-event).  
  
**Percent Tied :** Percentage of pairs where the observation with the desired outcome (event) has same predicted probability than the observation without the outcome (non-event).  
  
**c statistics (AUC) :**c-statistics is also called area under curve (AUC). Some statisticians also call it AUROC which stands for area under the receiver operating characteristics. It is calculated by adding Concordance Percent and 0.5 times of Tied Percent.  
  
**Gini coefficient or Somers' D statistic** is closely related to AUC. It is calculated by (2\*AUC - 1).  
  
In general, higher percentages of concordant pairs and lower percentages of discordant and tied pairs indicate a more desirable model.

**R Code for ROC, Concordant / Discordant :**

#Read Data

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

# Factor Variables

df$admit = as.factor(df$admit)

df$rank = as.factor(df$rank)

# Logistic Model

df$rank <- relevel(df$rank, ref='4')

mylogistic <- glm(admit ~ ., data = df, family = "binomial")

summary(mylogistic)$coefficient

# Predict

pred = predict(mylogistic, type = "response")

finaldata = cbind(df, pred)

AUC <- function (actuals, predictedScores){

fitted <- data.frame (Actuals=actuals, PredictedScores=predictedScores)

colnames(fitted) <- c('Actuals','PredictedScores')

ones <- fitted[fitted$Actuals==1, ] # Subset ones

zeros <- fitted[fitted$Actuals==0, ] # Subsetzeros

totalPairs <- nrow (ones) \* nrow (zeros) # calculate total number of pairs to check

conc <- sum (c(vapply(ones$PredictedScores, function(x) {((x > zeros$PredictedScores))}, FUN.VALUE=logical(nrow(zeros)))), na.rm=T)

disc <- sum(c(vapply(ones$PredictedScores, function(x) {((x < zeros$PredictedScores))}, FUN.VALUE = logical(nrow(zeros)))), na.rm = T)

concordance <- conc/totalPairs

discordance <- disc/totalPairs

tiesPercent <- (1-concordance-discordance)

AUC = concordance + 0.5\*tiesPercent

Gini = 2\*AUC - 1

return(list("Concordance"=concordance, "Discordance"=discordance,

"Tied"=tiesPercent, "AUC"=AUC, "Gini or Somers D"=Gini))

}

AUC(finaldata$admit, finaldata$pred)

## Using Integration to calculate ROC, Gini

Trapezoidal Rule Numerical Integration method is used to find area under curve. The area of a trapezoid is 

( xi+1 – xi ) \* ( yi + yi+1 ) / 2

In our case, **x** refers to values of false positive rate (1-Specificity) at different probability cut-offs, **y**refers to true positive rate (Sensitivity) at different cut-offs. **Vector x needs to be sorted**. Any observation with predicted probability that exceeds or equals probability cut-off is predicted to be an event; otherwise, it is predicted to be a nonevent. 

( fpri+1 – fpri ) \* ( tpri + tpri+1 ) / 2

fpr represents false positive rate (1- specificity). tpr represents true positive rate (sensitivity). See the image below showing step by step calculation. It includes a very few cut-offs for demonstration purpose.

|  |
| --- |
| <https://1.bp.blogspot.com/-J6J7vNf_UKk/XOMqe-1EFPI/AAAAAAAAHj0/xyXyMnmBFT4UwHDmFlLag8qgJKdxWH44ACLcBGAs/s1600/integral_steps.png> |
| Integration Calculation |

**# Rcode**

# Read Data

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

# Factor Variables

df$admit = as.factor(df$admit)

df$rank = as.factor(df$rank)

# Logistic Model

df$rank <- relevel(df$rank, ref='4')

mylogistic <- glm(admit ~ ., data = df, family = "binomial")

summary(mylogistic)$coefficient

# Predict

pred = predict(mylogistic, type = "response")

finaldata = cbind(df, pred)

library(ROCR)

predobj <- prediction(finaldata$pred, finaldata$admit)

perf <- performance(predobj,"tpr","fpr")

plot(perf)

# Trapezoidal rule of integration

# Computes the integral of Sensitivity (Y) with respect to FalsePosRate (x)

x = perf@x.values[[1]]

y = perf@y.values[[1]]

idx = 2:length(x)

testdf=data.frame(FalsePosRate = (x[idx] - x[idx-1]), Sensitivity = (y[idx] + y[idx-1]))

(AUROC = sum(testdf$FalsePosRate \* testdf$Sensitivity)/2)