**A Short Introduction To Logistic Regression**:

David Goody

1. Linear regression is one of the most common ways to trying to model a relationship between a number of factors. The approach works by constructing an equation (typically of the form y = ax + b) to try an explain the relationships between the predictor variables (x in the equation above) and the dependent variable (y in the equation above).
2. If the dependent (y) variable you are trying to predict only has values of 1 or 0 (or yes/no) you cannot apply normal “least squares” linear regression. This is because it requires that the dependent (y) variable has a normal distribution. Instead you need to use a generalised linear model (GLM) that is specifically set-up to try and predict a binary or binomial outcome (something with only 2 values). This is achieved by a applying a logit function to the linear regression equation to undertake logistic regression. The application of the logit function means that the outcomes are constrained between 0 and 1. The outcomes represent the % chance that the model is predicting that the event you are modelling occurs (the % chance of it being yes or no).



Image taken from <http://www.saedsayad.com/logistic_regression.htm>

1. In some circumstances a percentage may be a suitable outcome. If you are trying to identify a certain number of “at risk” cases you may need to apply a threshold to the model. Where the percentage value returned is above the threshold you would treat the result as “yes” and where it is below the threshold you would count the result as “no”. See the separate document on ROC curves and threshold selection for more information on how to select a threshold.

**Further Reading**

Introduction To Data Mining - <http://www.saedsayad.com/logistic_regression.htm>

Wikipedia (GLM) – <http://en.wikipedia.org/wiki/Generalized_linear_model>

Wikipedia (Logistic Regression) - <http://en.wikipedia.org/wiki/Logistic_regression>

**Annex A – Undertaking logistic regression in R**

#EXAMPLE OF APPLYING LOGISTIC REGRESSION IN R

#This example uses is based on the passenger list from the Titanic. We are trying to

#predict who survived and who died based on their age (Age), the fare they paid (Fare)

#gender (Female), embarkation point (South), how many siblings/spouses in each party (SibSp)

#how many parents/children in each passengers party (Parch)

#

#The dataset is used is publically available. It is the basis of a machine learning

#example on Kaggle (https://www.kaggle.com/c/titanic-gettingStarted).

############################################################

#SET-UP

#Set the working directory where you will be loading and saving data from and to

setwd("/Users/datascientist3/Desktop/R Packages/Kaggle Titanic")

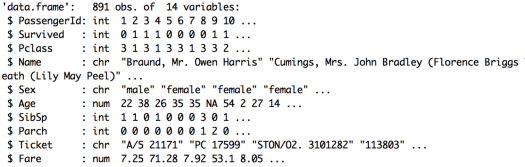
#Load Titanic dataset that we will train the data on

Titanic.Train <- read.csv("titanic\_train\_kaggle.csv", stringsAsFactors=FALSE)

#HAVE A LOOK AT THE DATA YOU ARE GOING TO USE

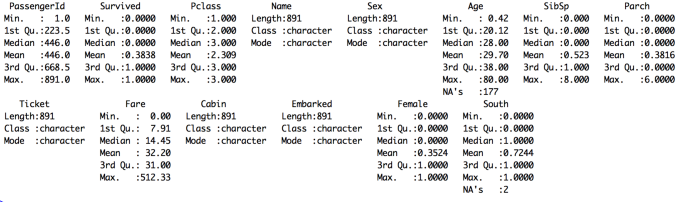
#Review structure of the data (shows field type and outputs form first few cases)

str(Titanic.Train)



#Look at a summary of the data (shows distribution of data for numeric fields)

summary(Titanic.Train)



#Convert text fields to dummy variables with 1/0 flags for use in equation

#Convert gender field into a 1/0 flag for female status

Titanic.Train$Female[Titanic.Train$Sex=='female'] <- 1

Titanic.Train$Female[Titanic.Train$Sex=='male'] <- 0

#Convert embarkation point 1/0 flag for those boarding at Southampton

Titanic.Train$South[Titanic.Train$Embarked=='S'] <- 1

Titanic.Train$South[Titanic.Train$Embarked=='C'] <- 0

Titanic.Train$South[Titanic.Train$Embarked=='Q'] <- 0

#FIT THE MODEL

#Apply GLM model to predict whether passengers survived based on the 5 variables

#This is a basic linear equation (y = a + bx + cz + .....)

Titanic.Train.GLM.model <- glm(Survived ~ Female + Age + SibSp + Parch + Fare + South, #Equation

data= Titanic.Train, #Dataset

family=binomial("logit")) #Type of GLM applied

#Shows a summary of the model including the coefficients applied and whether

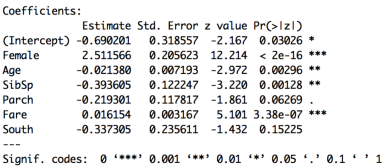
#they are statistically significant or not.

#Gender and fare are important variables. Embarkation point is not significant

summary(Titanic.Train.GLM.model)

#If you just want to pull out the coefficients use this command

summary(Titanic.Train.GLM.model)$coefficients



#APPLY THE MODEL

#We can take the model we've fitted and apply it to our training data

#This will give us a percentage value for each record.

#This is our prediction of how likely they were to survive

#The predict function is of the form predict([model name],[dataset name], [output required])

#In this case response gives us the % chance that they survive according to our model

Titanic.Train.GLM.Preds <-predict(Titanic.Train.GLM.model,Titanic.Train,type="response")

#Look at a summary of our predictions. The % chance of survival varies from 3% to 94%

#There are also 177 cases where the model can predict due to missing data

summary(Titanic.Train.GLM.Preds)



#Match the predictions back onto the main dataset to create a new file

Titanic.Train.With.Predictions <- data.frame(Titanic.Train, Prediction = Titanic.Train.GLM.Preds)

#Show the average prediction chance of survival for those who did live or die (Survived = 1 or 0)

#Model gives an average prediction of 26% for those who died and 61% for those who lived

aggregate(Prediction ~ Survived, data=Titanic.Train.With.Predictions,

FUN=function(x) {sum(x)/length(x)})



#Look at an individual result - the fifth case in the dataset

#Mr William Henry Allen did not survive (Survived = 0)

#Our model gave him a 16.9% chance of survival. Not very good odds! The model is doing OK here.

Titanic.Train.With.Predictions [5,] #The square brackets mean take the 5th row (then comma) and show all columns



#We can save our results to a csv file

write.csv(Titanic.Train.With.Predictions, file="Titanic.Train.With.Predictions.csv")