Logistic regression:

* the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary).
* Like all regression analyses, the logistic regression is a predictive analysis.
* Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more independent variables.
* Its output has only two possible outcomes (0,1)
  + P(y==0) = 1- P(y==1)1
  + P(y=1) == 1/(1+ e^-(b0 +b1 x1 + …..+ bk xk))
  + Odds = P(y==1)/P(y==0)
  + log(Odds) = b0 +b1 x1 + …..+ bk xk
  + This is also called as logit. The bigger logit is, the bigger P(y=1)
* In linear regression model, we find baseline model to predict average probability.
* Similarly, in logistic classification model, we have a standard baseline method that just predicts the most frequent outcome.
* TO split the data set into training and testing data set, we need to install packages
  + install.packages(“caTools”)
  + library(“caTools”) ⇒ to load the package
  + We use sample.split which is part of caTools that randomly splits the data. So , we all might have differnet training and testing datsets for same dataset
  + So, to maintain that we all get the same split, we will set seed that initialises the random number generator.
    - set.seed(88)
  + split = sample.split(quality$Poorcare, splitRatio = 0.75)
    - The split variable contains the values true and false
    - True is for training set. I.e., all the records with value True will come in training set
    - False is for testing set
  + Split function always balances the splitting of data. I.e., if the dataset has 75% outcome 1 then the split function takes care that its training and testing datasets also have 75% outcomes 1 respectiviely.
  + Trainset = subset(dataset\_name, split == TRUE)
  + Testset = subset(dataset\_name, split == FALSE)
* To generate a logistic regression model:
  + Logreg = glm(dependent\_variable ~ independent\_variables, data = dataset\_name, family = binomial)
    - Glm ⇒ generalized linear regression model
    - Binomial to tell that it is an logistic regression.
  + Similar to R-squared value in linear regression model, we have AIC(Akaike information criterion) value in logistic regression that measures the quality of the model.
  + It can be only compared on the models of same dataset.
* To predict:
  + predictTrain = predict(model\_name, type = “response”)
  + Response type tells the model to give the probabilities for the results.
* The outcome of a logistic regression model is a probability.
* Often, we need binary outcomes.
* For this we need a threshold value t
  + If P(y=1)>=t, outcome = 1
  + If P(y=1)< t, outcome = 0
* T value should be decided carefully
  + If t is large, the outcomes of 1 would be low and we might predict wrong for those and vice versa
  + If there is no preference for wrrors then its better that t= 0.5
  + We can select a t value using classification matrix

|  |  |  |
| --- | --- | --- |
|  | Predicted = 0 | Predicted = 1 |
| Actual = 0 | True negatives(TN) | False Positives(FP) |
| Actual = 1 | False Negatives(FN) | True positives(TP) |

* + We can get to the types of errors we are making by
    - Sensitivity = TP/(TP + FN)
    - Specificity = TN/ (TN +FP)
    - Overal accuracy = (TN+TP)/N
    - Overall error rate = (FP+FN)/N
    - False negative error rtae = FN/(TP+FN)
    - Flase positive error rate = FP/(TN + FP)
  + A model with a higher threshold have lower sensitivity and higher specificity and vice versa
  + Classification table with t= 0.5
    - table(qualityTrain&poorcare, PredictTrain>0.5)
* To predict on a new data set
  + predict(model\_naem, type = “response” , newdata = testset\_name)
* To calculate AUC
  + library(ROCR)
  + ROCRpred = prediction(predictTest, test$TenYearCHD)
    - predictTest ⇒ test\_dataset\_prediction
    - Test ⇒ test\_dataset
  + as.numeric(performance(ROCRpred, “auc”) @y.values)