CUTe - 2

Application of Logistic Regression and naïve Bayes

Mudit Gupta | PGP in Big Data and Optimization | 14th April, 2018

# INDEX

|  |  |  |
| --- | --- | --- |
| Sr. No. | Topic | Page No. |
| 1. | Problem Statement | 2 |
| 2. | Business Objective | 2 |
| 3. | Logistic Regression | 3 |
| 4. | VIF | 4 |
| 5. | SMOTE | 5 |
| 6. | Lasso/Ridge Regression | 7 |
| 7. | Methodology | 8 |
| 8. | Approach | 10 |
| 9. | Final Results | 16 |
| 10. | Conclusion |  |

# Problem Statement

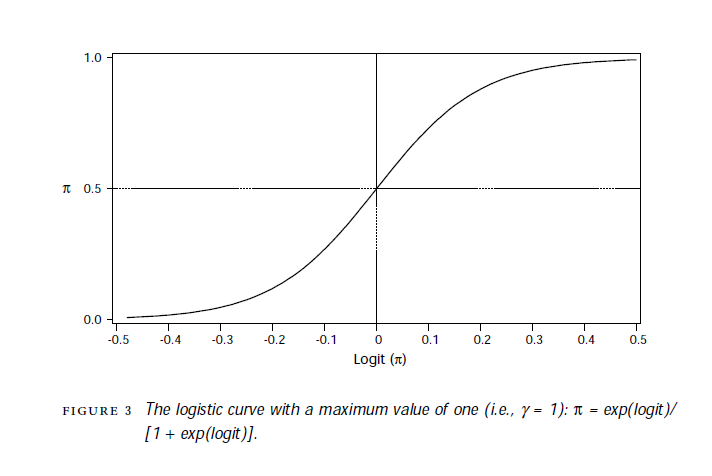
A healthcare organization together with a couple of government hospitals in a city has collected information about the vitals that would reveal if the person might have a coronary heart disease in the next ten years or not. This study is useful in early identification of disease and have medical intervention if necessary. This would help not only in improving the health conditions but also the economy as it has been identified that health performance and economic performance are interlinked. Given the data, we need to develop appropriate models to identify/predict if the person likely to have heart disease or not.

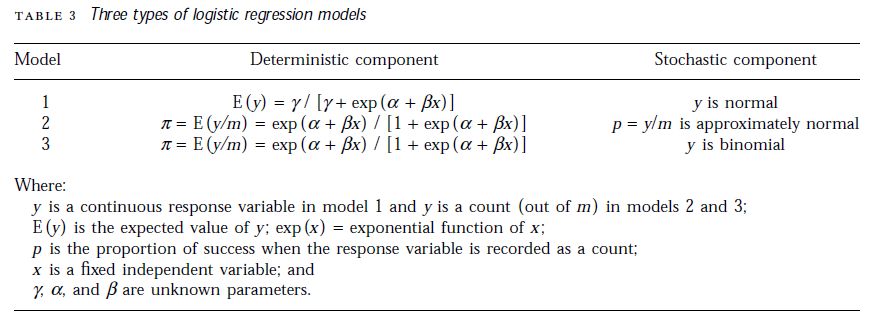
# Business Objective

The objective of the study is to predict whether an individual is likely to get affected with CHD or not in coming 10 years. The insurance company is looking for sufficient evidence to predict high risk CHD individuals for estimating premium of insurance products. Classification of high risk v/s low risk is an educated estimate, using a simple formula that requires very complicated input. We try to use medical records as predictors of the classification problem. The model statistics can help in developing a product for high risk individuals to cover future medical expenses and administrative costs.

# Logistic Regression

Logistic regression models use the logistic function to fit models to data. This is an S-shaped function and an example curve is shown. This function can be used to fit data in three ways. Although each is distinct, these approaches can be called logistic regression and are briefly described. They all fit a response variable, either *y* or *y*/*m*, to the S-shaped logistic function of the independent variable, *x*. The first model could fit growth data (*y* on any scale) versus time (*x* ) with a logistic curve, while the next two fit proportional responses (with values restricted to the range between zero and one) with the logistic curve.





# VIF

Collinearity, or excessive correlation among explanatory variables, can complicate or prevent the identification of an optimal set of explanatory variables for a statistical model. For example, forward or backward selection of variables could produce inconsistent results, variance partitioning analyses may be unable to identify unique sources of variation, or parameter estimates may include substantial amounts of uncertainty. The temptation to build an ecological model using all available information (i.e., all variables) is hard to resist. Lots of time and money are exhausted gathering data and supporting information. We also hope to identify every significant variable to more accurately characterize relationships with biological relevance. Analytical limitations related to collinearity require us to think carefully about the variables we choose to model, rather than adopting a naive approach where we blindly use all information to understand complexity. The purpose of this blog is to illustrate use of some techniques to reduce collinearity among explanatory variables using a simulated dataset with a known correlation structure.

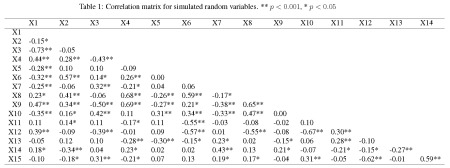
A simple approach to identify collinearity among explanatory variables is the use of variance inflation factors (VIF). VIF calculations are straightforward and easily comprehensible; the higher the value, the higher the collinearity. A VIF for a single explanatory variable is obtained using the r-squared value of the regression of that variable against all other explanatory variables:

\displaystyle VIF_j=\frac{1}{1-R_j^2} 

where the VIF for variable j is the reciprocal of the inverse of R^2 from the regression. A VIF is calculated for each explanatory variable and those with high values are removed. The definition of ‘high’ is somewhat arbitrary but values in the range of 5-10 are commonly used.

Several packages in R provide functions to calculate VIF: [vif](http://rss.acs.unt.edu/Rdoc/library/HH/html/vif.html) in package HH, [vif](http://hosho.ees.hokudai.ac.jp/~kubo/Rdoc/library/car/html/vif.html)in package car, [VIF](http://www.inside-r.org/packages/cran/fmsb/docs/VIF) in package fmsb, [vif](http://rss.acs.unt.edu/Rdoc/library/faraway/html/vif.html) in package faraway, and [vif](http://cran.open-source-solution.org/web/packages/VIF/index.html) in package VIF. The number of packages that provide VIF functions is surprising given that they all seem to accomplish the same thing. One exception is the function in the VIF package, which can be used to create linear models using VIF-regression. The nuts and bolts of this function are a little unclear since [the documentation](http://cran.open-source-solution.org/web/packages/VIF/VIF.pdf) for the package is sparse. However, what this function does accomplish is something that the others do not: stepwise selection of variables using VIF. Removing individual variables with high VIF values is insufficient in the initial comparison using the full set of explanatory variables. The VIF values will change after each variable is removed. Accordingly, a more thorough implementation of the VIF function is to use a stepwise approach until all VIF values are below a desired threshold. For example, using the full set of explanatory variables, calculate a VIF for each variable, remove the variable with the single highest value, recalculate all VIF values with the new set of variables, remove the variable with the next highest value, and so on, until all values are below the threshold.

The covariance matrix is chosen from a uniform distribution such that some variables are correlated while some are not. The correlation matrix for the random variables should look very similar to the correlation matrix from the actual values (as sample size increases, the correlation matrix approaches cov.mat).

[](https://beckmw.files.wordpress.com/2013/02/cor_tab1.pdf)

Response variable as a linear combination of the explanatory variables is shown in the table. First, we create a vector for the parameters describing the relationship of the response variable with the explanatory variables. Then, we use some matrix algebra and a randomly distributed error term to create the response variable. This is the standard form for a linear regression model.

# SMOTE

Unbalanced data refers to classification problems where we have unequal instances for different classes. Having unbalanced data is actually very common in general, but it is especially prevalent when working with disease data where we usually have more healthy control samples than disease cases. Even more extreme unbalance is seen with fraud detection, where e.g. most credit card uses are okay and only very few will be fraudulent. In the example from a breast cancer dataset, we have about twice as many benign than malignant samples.

Example:

summary(bc\_data$classes)

## benign malignant

## 458 241

Most machine learning classification algorithms are sensitive to unbalance in the predictor classes. Consider an even more extreme example than our breast cancer dataset: assuming we have 10 malignant vs 90 benign samples. A machine learning model that has been trained and tested on such a dataset could now predict “benign” for all samples and still gain a very high accuracy. An unbalanced dataset will bias the prediction model towards the more common class.

The basic theoretical concepts behind over- and under-sampling are very simple:

* With under-sampling, we randomly select a subset of samples from the class with more instances to match the number of samples coming from each class. In our example, we would randomly pick 241 out of the 458 benign cases. The main disadvantage of under-sampling is that we lose potentially relevant information from the left-out samples.
* With oversampling, we randomly duplicate samples from the class with fewer instances or we generate additional instances based on the data that we have, so as to match the number of samples in each class. While we avoid losing information with this approach, we also run the risk of overfitting our model as we are more likely to get the same samples in the training and in the test data, i.e. the test data is no longer independent from training data. This would lead to an overestimation of our model’s performance and generalizability.

In reality though, we should not simply perform over- or under-sampling on our training data and then run the model. We need to account for cross-validation and perform over- or under-sampling on each fold independently to get an honest estimate of model performance.

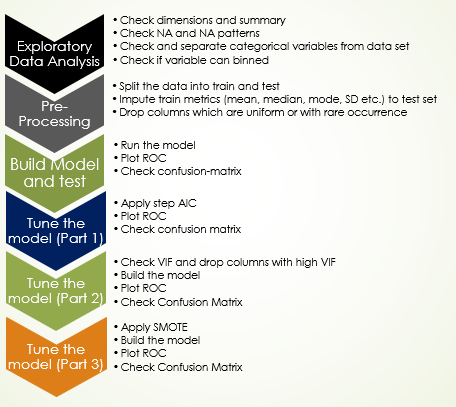
Therefore, we use SMOTE(Synthetic Minority Over-sampling Technique).

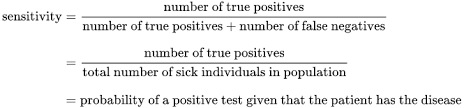
It is an approach to the construction of classifiers from imbalanced datasets. A dataset is imbalanced if the classification categories are not approximately equally represented. Often real-world data sets are predominately composed of “normal” examples with only a small percentage of “abnormal” or “interesting” examples. It is also the case that the cost of misclassifying an abnormal (interesting)example as a normal example is often much higher than the cost of the reverse error. Under-sampling of the majority (normal)class has been proposed as a good means of increasing the sensitivity of a classifier to the minority class. This technique shows that a combination of SMOTE method of over-sampling the minority (abnormal)class and under-sampling the majority (normal)class can achieve better classifier performance (in ROC space) than only under-sampling the majority class. This technique also shows that a combination of our method of over-sampling the minority class and under-sampling the majority class can achieve better classifier performance (in ROC space) than varying the class priors in Naive Bayes. The method of over-sampling the minority class involves creating synthetic minority class examples. It is evaluated using the area under the Receiver Operating Characteristic curve (AUC)and the ROC convex hull strategy.

# Lasso/Ridge Regression

Ridge regression and the lasso are closely related, but only the Lasso has the ability to select predictors. Like OLS, ridge attempts to minimize residual sum of squares of predictors in a given model. However, ridge regression includes an additional ‘shrinkage’ term – the square of the coefficient estimate – which shrinks the estimate of the coefficients towards zero. The impact of this term is controlled by another term, lambda (determined seperately). Two interesting implications of this design are the facts that when λ = 0 the OLS coefficients are returned and when λ = ∞, coefficients will approach zero.

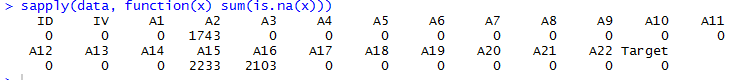
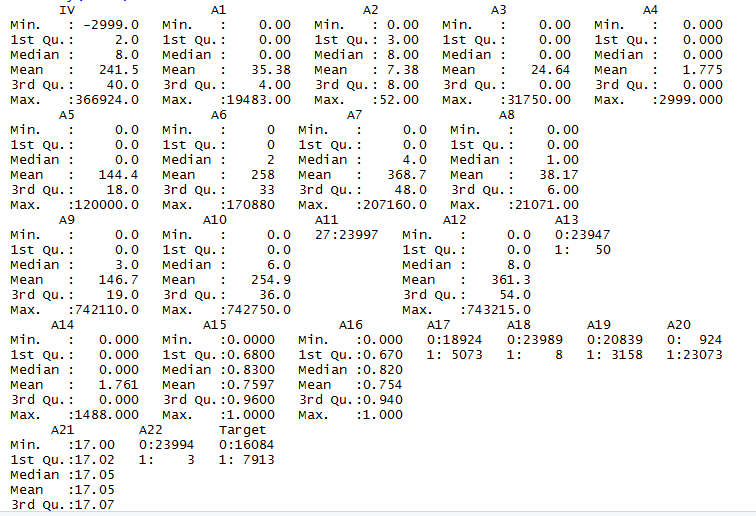
# Methodology/Approach



1. Check dimensions of the dataset.
2. Check NA and NA patterns.
3. Check and separate categorical variables from numeric variables.
4. Check if records can be binned(group).
5. Impute test with train (mean, median, mode).
6. Drop the columns which have uniform values or which have rare occurrence of events.
7. Split data into train set and test set.
8. Train the model.
9. Predict for test set using command predict() in R console.
10. Plot ROC curve.
11. Check confusion matrix(TP,FP,TN,FN).
12. Check for sensitivity.  
    
13. Check VIF for multi-collinearity.
14. Drop the columns with High VIF(follow thumb rule of (VIF > 10) -> !variables).
15. Repeat steps 7 to 12.
16. Apply stepAIC() method on the model.
17. Repeat steps 9 to 12.
18. Store the whole dataset in a new variable.
19. Split the dataset into train and test.
20. Apply SMOTE on train set.
21. Drop columns which are uniform or which have rare occurrence of events.
22. Train the model with train set.
23. Predict for test set using command predict() in R console.
24. Plot ROC curve.
25. Check confusion matrix(TP,FP,TN,FN).
26. Check for sensitivity.
27. Drop variables with high VIF.
28. Rebuilt the model.
29. Predict for test set using command predict() in R console.
30. Plot ROC.
31. Check confusion matrix(TP,FP,TN,FN).
32. Check sensitivity.
33. Apply Lasso/Ridge Regression  
    ([How to deal with perfect separation in logistic regression?](https://stats.stackexchange.com/questions/11109/how-to-deal-with-perfect-separation-in-logistic-regression))

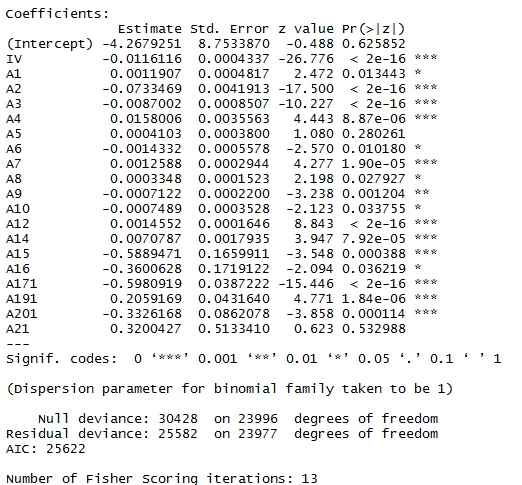
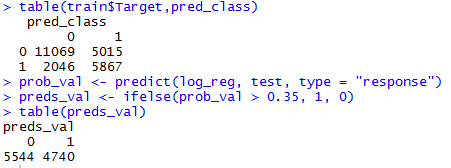
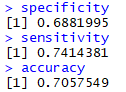
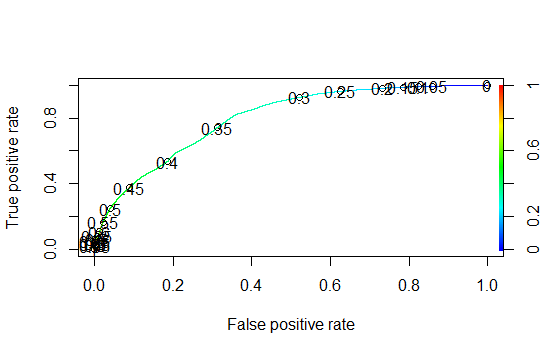
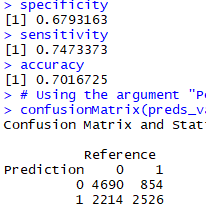
# Approach

## General Observations

1. No. of NA’s  
   
2. NA patterns  
   
3. Summary of the data  
   
4. Bin(range) -> A21  
   
5. Frequency of Bin -> A21  
   

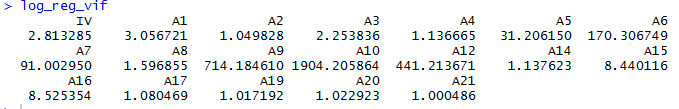
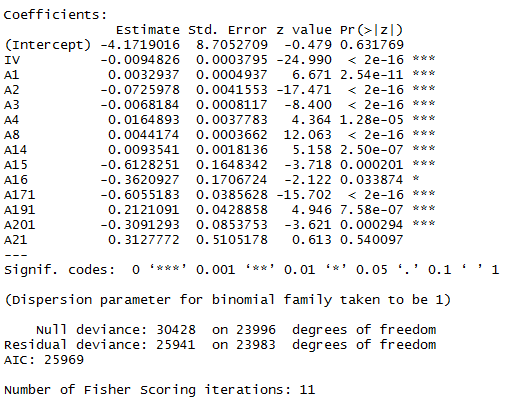
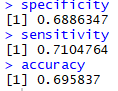
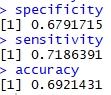
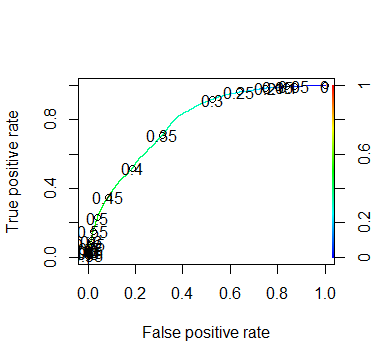
# Approach(contd.)

## Model 1(RAW GLM)

1. Summary of the model  
   
2. Confusion Matrix, Sensitivity, Specificity, Accuracy(Train)  
    
3. ROC curve  
   
4. Area under curve  
   
5. Confusion Matrix, Sensitivity, Specificity, Accuracy(Test)  
   
6. Reason of Rejection – The sensitivity of the model is quite low which is not acceptable.

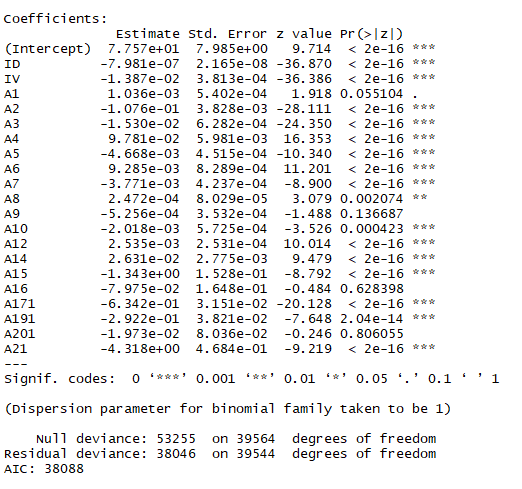
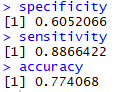
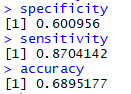
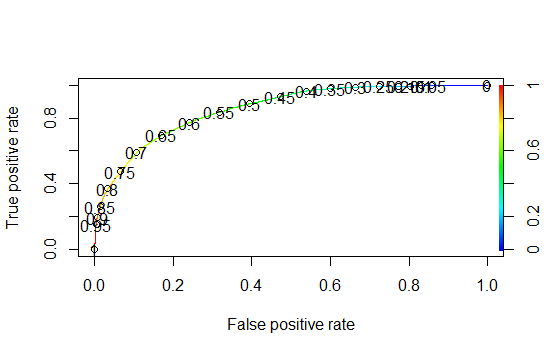
# Approach(contd.)

## MODEL 2(VIF)

1. VIF  
   
2. Summary  
   
3. Sensitivity, Specificity, Accuracy(Train)  
   
4. Sensitivity, Specificity, Accuracy(Test)  
   
5. ROC curve  
   
6. Area under curve  
   
7. Reason of Rejection – The sensitivity of the model on test set is quite low even after removing collinearity.

# Approach(contd.)

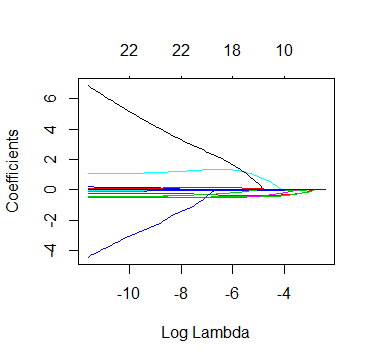
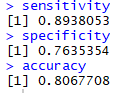
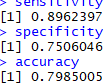
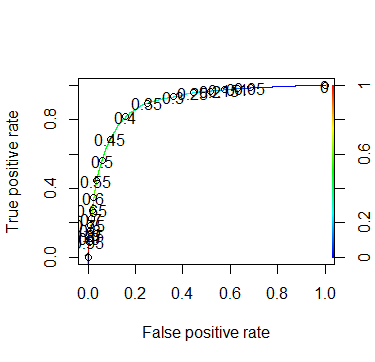
## MODEL 3(SMOTE)

1. Summary  
   
2. Sensitivity, Specificity, Accuracy(Train)  
   
3. Sensitivity, Specificity, Accuracy(Test)  
   
4. ROC curve  
   
5. Area under curve  
   
6. Reason of Rejection – The warning was still occurring, which means there may be presence of perfect linear separation.

Warning message:   
glm.fit: fitted probabilities numerically 0 or 1 occurred

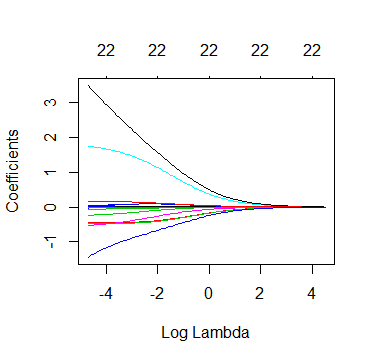
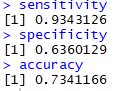
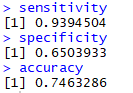
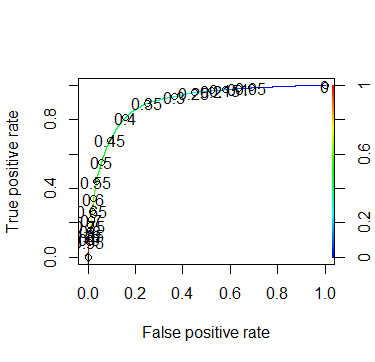
# Approach(contd.)

## MODEL 4(LASSO)

1. Lasso Lambda Graph  
   
2. Train v/s Test metric  
    
3. ROC Curve  
   
4. Area under curve  
   
5. Reason for acceptance of the model – Lasso has the ability to select predictors so that we may overcome perfect linear separation problem and penalizing the coefficients.

# Approach(contd.)

## MODEL 5(RIDGE)

1. Lasso Lambda Graph  
   
2. Train v/s Test metric  
    
3. ROC Curve  
   
4. Area under curve  
   
5. Reason for acceptance of the model –  ridge attempts to minimize residual sum of squares of predictors in a given model. However, ridge regression includes an additional ‘shrinkage’ term – the square of the coefficient estimate – which shrinks the estimate of the coefficients towards zero.