MSc in Business Analytics AUEB
Statistics for Business Analytics II

Classification & Clustering

Professor: Karlis Dimitrios | Student: Despotis Spyridon - P2822111

Contents

Introduction

In this project we have been asked to experiment with classification and clustering methods. The data cleaning and transformations are the same as the first project. In the first part of this project, we are focusing on creating a predictive model to classify whether a client will buy or not buy a new product, and in the second part, to use specific variables to cluster the clients, and to characterize the clusters. Then we will investigate if the final clustering relates to the subscribe variable.

The first part deals with supervised learning methods with Lasso, Logistic Regression and LDA combined with different variable selection methods. Supervised learning uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time.

The second part deals with unsupervised learning problems and tries to implement cluster analysis. The goal of this unsupervised machine learning technique is to find similarities in the data point and group similar data points together. Hierarchical clustering starts by assigning all data points as their own cluster. As the name suggests it builds the hierarchy and in the next step, it combines the two nearest data points and merges them together into one cluster. On the other hand, K-Means performs the division of objects into clusters that share similarities and are dissimilar to the objects belonging to another cluster.

1. Classification

In this section we create a predictive model to classify whether a client will buy or not buy a new product. We use three different methods and we assess how good the predictions are by comparing the three models. We split the data to 60% in training and 40% in testing.

1.1. Lasso Method

Lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model. We have chosen this method because it can eliminate variables that are not useful for the analysis quickly, and effectively. The crucial decision was to choose the appropriate threshold that would dichotomize the values in order to classify our observations and define if a client will buy or not buy a new product. In the graph below we see a comparison between different dichotomized values with their percentages.

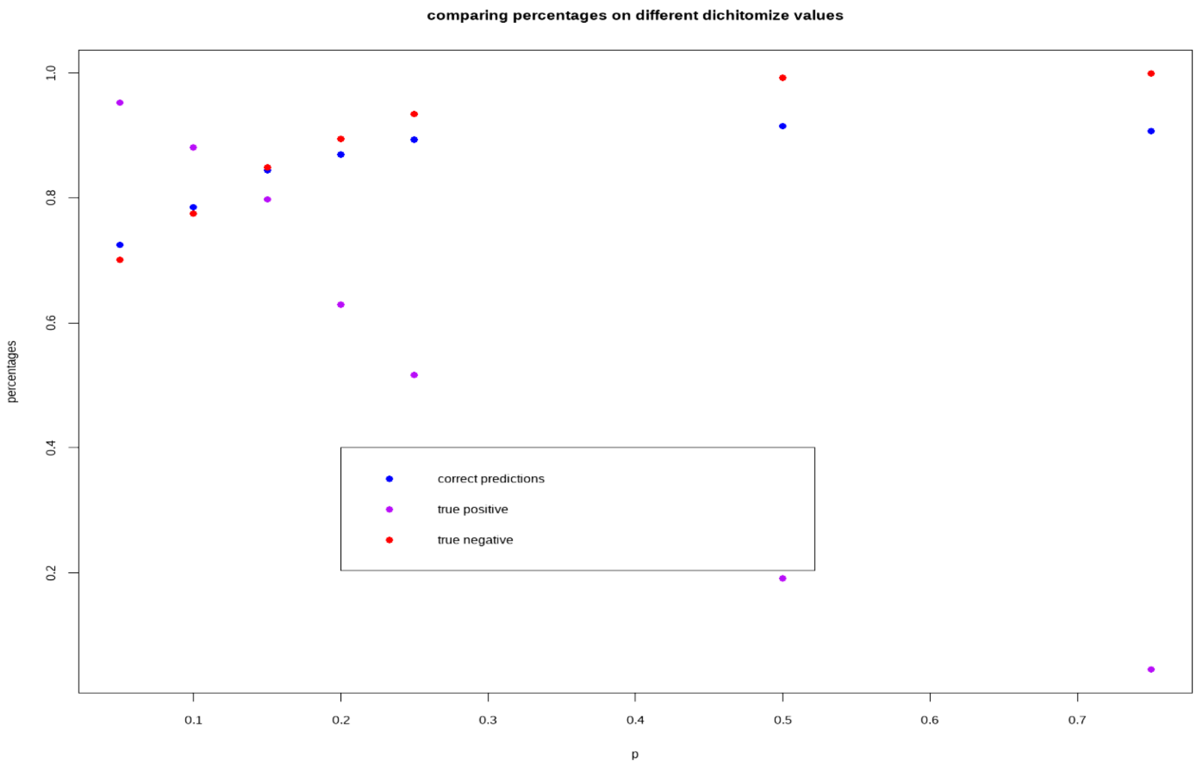


Figure 1-1 Comparing Percentages on Different Dichitomize Values

From the graph above we can see as we increase the threshold, the true positives percentages are decreasing, while at the same time the true negatives and correct predictions percentages are increasing. Therefore a threshold with a probability of 0.15 seems to be a good choice in order to dichotomize the values. Having that in mind, we proceed by using a confusion matrix in test data in order to have a straightforward view of the effectiveness of our decision and evaluate the ability of the model to make accurate predictions.

Confusion Matrix & Statistics

Prediction	Reference: 0	Reference: 1
0	**12235**	**312**
1	**2181**	**1226**

Accuracy :	**0.8437**
95% CI :	(0.838 , 0.8493)
No Information Rate :	0.9036
P-Value [Acc > NIR] :	1
Kappa :	0.4186
Mcnemar's Test P-Value :	<2e-16
Sensitivity :	**0.79714**
Specificity :	0.84871
Pos Pred Value :	**0.35985**
Neg Pred Value :	**0.97513**
Prevalence :	0.09640
Detection Rate :	0.07685
Detection Prevalence :	0.21355
Balanced Accuracy :	0.82292
'Positive' Class :	1

Table 1.1 Results in Confusion Matrix with Threshold 0.15

We can conclude that if the value is 0 in the test data in our model, it correctly predicts 0 in 12235 cases, and it predicts a faulty 1 in 2818 cases. In addition, if the value is 1 our model predicts a faulty 0 in 312 cases and predicts 1 correctly in 1226 cases. The total sensitivity (the metric that evaluates a model's ability to predict true positives of each available category) is 0.79714 and the accuracy (metric that summarizes the performance of a classification model as the number of correct predictions divided by the total number of predictions) is 0.8437. Also, we found another interesting threshold (0.5) in our test data in order to compare it with the 0.15

Confusion Matrix and Statistics		
	Reference	
Prediction	0	1
0	**14293**	**1243**
1	**123**	**295**

Accuracy :	**0.9144**
95% CI :	(0.9099- 0.9187)
No Information Rate :	0.9036
P-Value [Acc > NIR] :	1.455e-06
Kappa :	0.2716
Mcnemar's Test P-Value :	< 2.2e-16
Sensitivity :	**0.19181**
Specificity :	0.99147
Pos Pred Value :	**0.70574**
Neg Pred Value :	**0.91999**
Prevalence :	0.09640
Detection Rate :	0.01849
Detection Prevalence :	0.02620
Balanced Accuracy :	0.59164
'Positive' Class :	1

Table 1.2 Results in Confusion Matrix with Threshold 0.5

In this case as we can see in the table, compared to the confusion matrix of the 0.15 threshold, we have better accuracy of 0.9144, however worse sensitivity of 0.19181. So, in fact, the best threshold is not just one number, it depends on the strategy of each company. For example, if a company wants to predict more subscriptions, the first threshold seems to be better since it can correctly predict 1226 values for subscription, compared to 295 with the second threshold. But at the same time, this company will have to consider the cost of having many wrong predictions for having more subscriptions for customers that did not subscribe. That could possibly lead to more inaccurate planning for the company, since it will be expecting development of a larger customer base, or money lost from campaigns that would be expected to bring more customers. Therefore, the set of thresholds is also a management decision taking into account the amount of risk that the company wants to take at the moment.

1.2. Logistic Regression Method

We used the Logistic Regression method followed by variable selection with AIC and BIC criterions. We continued only with AIC as it is better for estimating predictions. For cross validation we decided to use 10 folds with the procedure of resampling repeated 5 times. As thresholds, we found that the probabilities 0.1 and 0.25 seemed like good choices in order to dichotomize our values. The final decision from the two depends on the strategy of each company and how risk averse, risk neutral or risk seeking the company is at the moment.

Table 1.3 Confusion Matrix Results with AIC Criterion & 0.1 Threshold

Confusion Matrix and Statistics

Prediction	Reference 0	Reference 1
0	11707	196
1	2709	1342

Accuracy :	**0.8179**
95% CI :	(0.8118, 0.8239)
No Information Rate :	0.9036
P-Value [Acc > NIR] :	1
Kappa :	0.3958
Mcnemar's Test P-Value :	<2e-16
Sensitivity :	**0.87256**
Specificity :	0.81208
Pos Pred Value :	**0.33128**
Neg Pred Value :	**0.98353**
Prevalence :	0.09640
Detection Rate :	0.08412
Detection Prevalence :	0.25392
Balanced Accuracy :	0.84232
'Positive' Class :	1

Table 1.4 Variables Selected with AIC criterion

	Dependent variable: SUBSCRIBED
jobblue-collar	-0.391***
	(0.082)
jobservices	-0.248**
	(0.102)
jobtechnician	-0.091
	(0.082)
jobOther	0.111
	(0.069)
maritalmarried	-0.096
	(0.085)
maritalsingle	0.078
	(0.091)
maritalunknown	0.287
	(0.469)
contacttelephone	-0.261***
	(0.087)
poutcomenonexistent	0.471***
	(0.084)
poutcomesuccess	2.112***
	(0.130)
cons.price.idx	0.840***
	(0.096)
cons.conf.idx	0.151***
	(0.008)
dur_bin[200,400)	1.356***
	(0.075)
dur_bin[400,5.1e+03)	3.410***
	(0.073)
campaign_bin(1,2]	-0.193***
	(0.063)
campaign_bin(2,60]	-0.156**
	(0.064)
nr.employed_bin[5.2e+03,5.3e+03)	0.951***
	(0.173)
emp.var.rate_bin[1,1.5)	-3.936***
	(0.171)
seasonspring	1.236***
	(0.099)
seasonsummer	1.505***
	(0.105)
seasonwinter	1.234***
	(0.251)
Constant	-75.985***
	(8.772)
Observations	23,929
Note:	*p<0.1; **p<0.05; ***p<0.01

Confusion Matrix and Statistics

Prediction	Reference 0	Reference 1
0	13236	598
1	1180	940

Accuracy :	**0.8886**
95% CI :	(0.8836, 0.8934)
No Information Rate :	0.9036
P-Value [Acc > NIR] :	1
Kappa :	0.4528
Mcnemar's Test P-Value :	<2e-16
Sensitivity :	**0.61118**
Specificity :	0.91815
Pos Pred Value :	**0.44340**
Neg Pred Value :	**0.95677**
Prevalence :	0.09640
Detection Rate :	0.05892
Detection Prevalence :	0.13288
Balanced Accuracy :	0.76466
'Positive' Class :	1

Table 1.5 Confusion Matrix Results with AIC Criterion & 0.25 Threshold

1.3. Linear Discriminant Analysis Method (LDA)

Linear Discriminant Analysis is focused on maximizing the separability among known categories. We experimented with different formulas of variable selection (AIC, BIC, Lasso) but only the AIC criterion seemed more efficient. From the following histograms were created using the LDA method, following from the AIC criterion. We can see that there is a certain amount of overlapping between group zero and group one but not a significant amount.

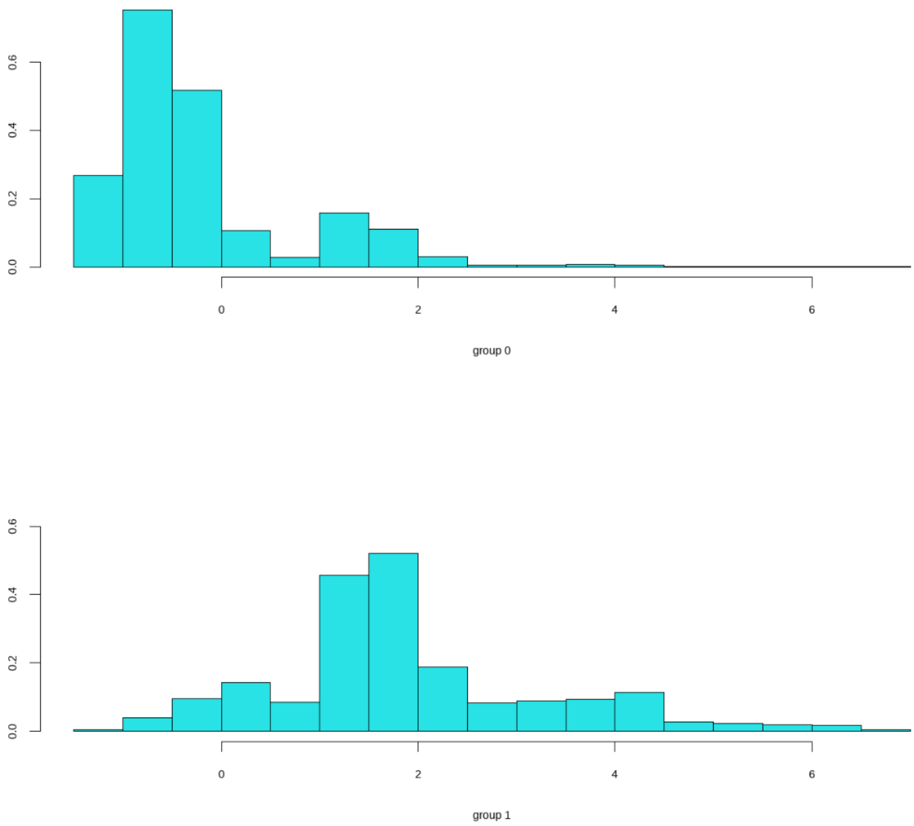


Figure 1-2 Histogram with distribution of observations between two groups

Table 1.7 Confusion Matrix with AIC criterion

Confusion Matrix & Statistics

Prediction	Reference 0	Reference 1
0	**13868**	**936**
1	**548**	**602**

Accuracy :	**0.907**
95% CI :	(0.9024, 0.9114)
No Information Rate :	0.9036
P-Value [Acc > NIR] :	0.07507
Kappa :	0.3983
Mcnemar's Test P-Value :	< 2e-16
Sensitivity :	**0.39142**
Specificity :	0.96199
Pos Pred Value :	**0.52348**
Neg Pred Value :	**0.93677**
Prevalence :	0.09640
Detection Rate :	0.03773
Detection Prevalence :	0.07208
Balanced Accuracy :	0.67670
'Positive' Class :	1

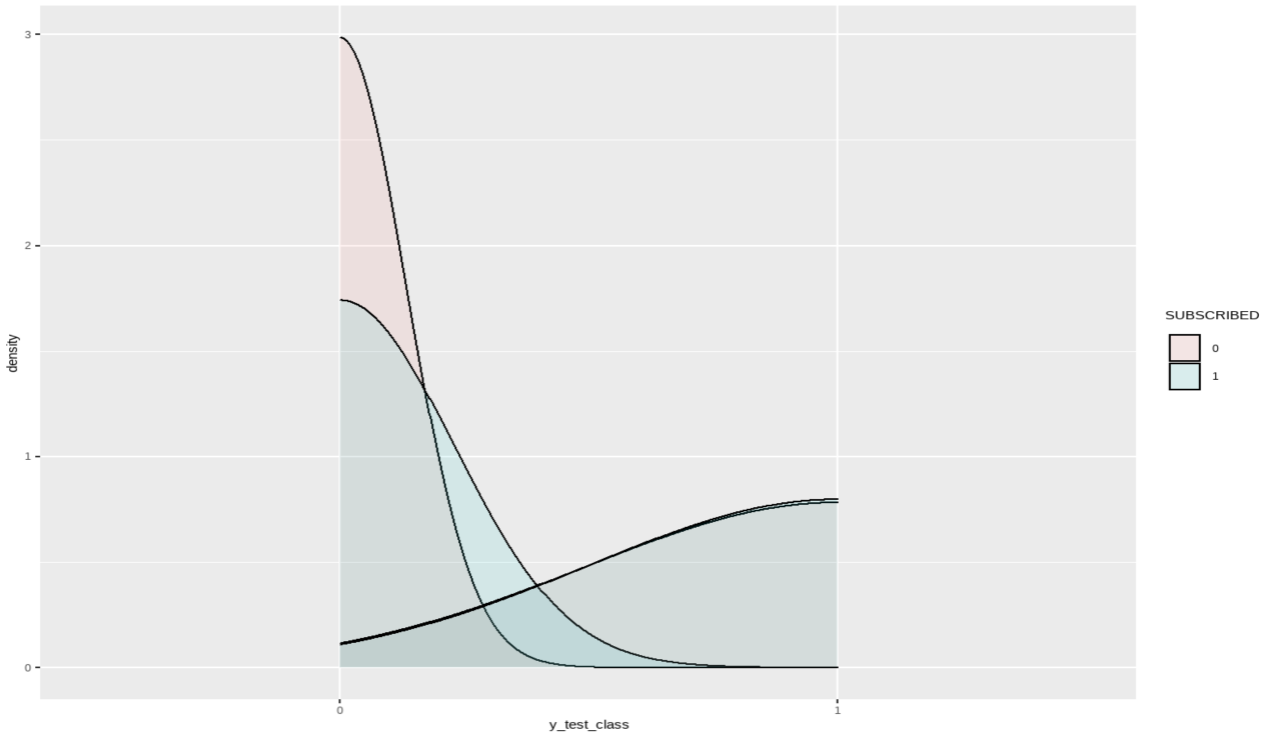


Figure 1-3 LDA compared with true values in test set

1.4. Models Comparison

From the graph below, we plotted ROC curves for all methods in order to have a straight forward view for which is the best. We can conclude that there are small differences between them with the AIC method being slightly better.

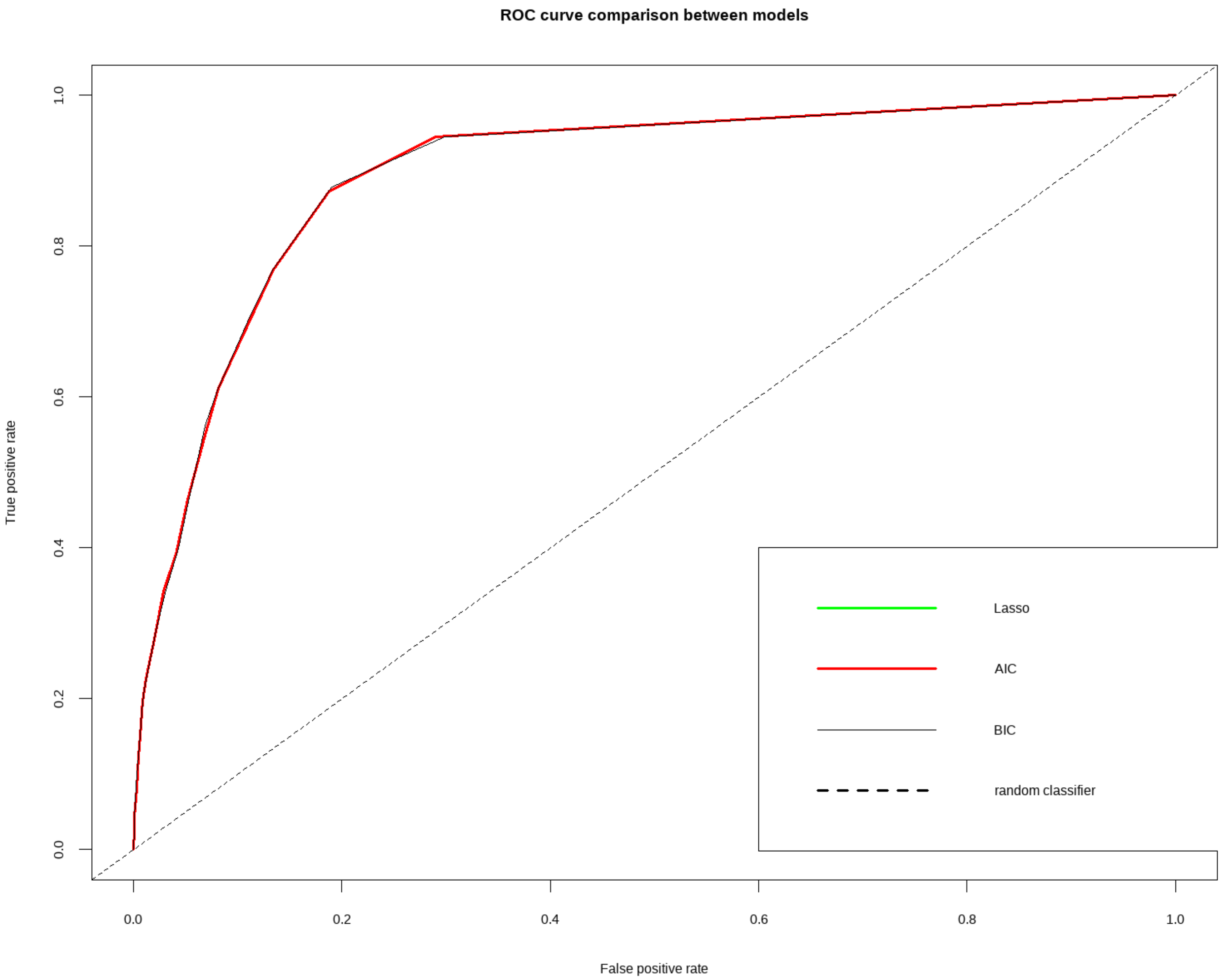
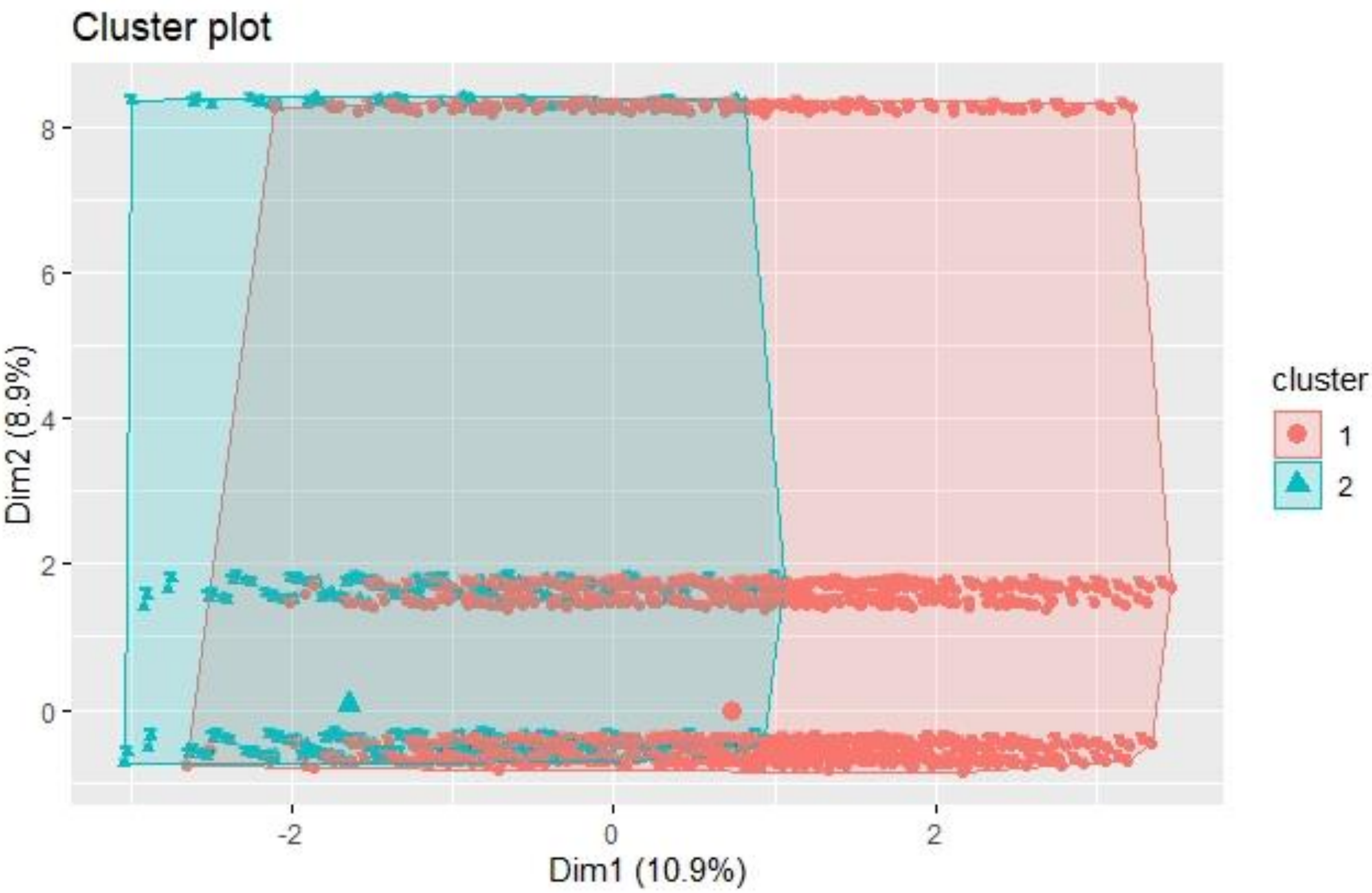


Figure 1-4 ROC Curves Comparison

2. Model Based Clustering

In this section we tried a variety of different datasets from different variable selection methods (with random subsetting, with and without scaling) with model based clustering, k-means and Clara. The first result from model based clustering with Lasso variable selection (in the data given in part 2) was the bellow cluster plot, with the cluster proved to be not been statistically significant.



Then, due to the fact that we tested many different classes and methods with the given variables and not having good results, we proceed by investigating which other variables would improve our clustering. Then we add them in our dataset and we checked again if there would be any positive difference comparing to our first clustering that was not so good.

The below final results are from model based clustering with Lasso variable selection and dummy variables. Therefore, we found 2 classes were given the best possible outcome. As we can see in the graph below there seems to be a small overlap between the two groups and also we can see the size of each cluster.

Figure 2-1 Model Based Cluster with 2 Classes

Next we run the function "adjustedRandIndex" that measures the similarity between two classifications of the same objects by the proportions of agreements between the two partitions. Specifically, we compare the cluster with the subscribe variable. The result was -0.0170, which indicates that the two partitions have no similarity between them. We proceeded by adding the cluster in our data frame in order to run a Logistic Regression Model with the cluster and find if it is statistically significant.

	Estimate	Std. Error	z value	Pr(>|z|)	
job_Other	3.69159	0.20495	18.012	< 2e-16	***
job_services	3.18001	0.21523	14.775	< 2e-16	***
job_admin.	3.60203	0.20529	17.546	< 2e-16	***
job_blue-collar	3.07033	0.20897	14.693	< 2e-16	***
job_technician	3.50773	0.20824	16.845	< 2e-16	***
poutcome_nonexistent	-2.20125	0.08327	-26.434	< 2e-16	***
poutcome_failure	-2.48112	0.0957	-25.926	< 2e-16	***
poutcome_success	NA	NA	NA	NA	
age_bin_(40,100]	-0.036	0.04112	-0.875	0.381	
`age_bin_(15,40]`	NA	NA	NA	NA	
dur_bin_[200,400)	-1.95244	0.04793	-40.732	< 2e-16	***
dur_bin_[0,200)	-3.2675	0.05442	-60.042	< 2e-16	***
dur_bin_[400,5.1e+03)	NA	NA	NA	NA	
emp.var.rate_bin_[1,1.5)	-1.91705	0.078	-24.577	< 2e-16	***
emp.var.rate_bin_[-4,1)	NA	NA	NA	NA	
season_spring	-1.22655	0.19036	-6.443	1.17e-10	***
season_summer	0.02967	0.19632	0.151	0.88	
season_autumn	-1.34213	0.19305	-6.952	3.59e-12	***
season_winter	NA	NA	NA	NA	
cluster	**-0.59787**	**0.10469**	**-5.711**	**1.12e-08**	***

Table 2.1 Summary of model with cluster variable statistical significant

From the above model summary table we can conclude that the N/A's can be eliminated since they can probably be calculated from other variables. For the effects of each predictor to the intercept we can conclude that:

- The effect of job Other is statistically significant and positive (beta = 3.69)
- The effect of job services is statistically significant and positive (beta = 3.18)
- The effect of job admin is statistically significant and positive (beta = 3.60)
- The effect of job blue-collar is statistically significant and positive (beta = 3.07)
- The effect of job technician is statistically significant and positive (beta = 3.50)
- The effect of poutcome_nonexistent is statistically significant and negative(beta=-2.20)
- The effect of poutcome_failure is statistically significant and negative (beta = -2.48)
- The effect of dur_bin_[200,400) is statistically significant and negative(beta=-1.95)
- The effect of dur_bin_[0,200) is statistically significant and negative (beta = -3.27)
- The effect of emp.var.rate_bin_[1,1.5) is statistically significant and negative (beta =-1.91)
- The effect of season_spring is statistically significant and negative (beta = -1.22)
- The effect of season_autumn is statistically significant and negative (beta =-1.34)
- The effect of cluster is statistically significant and negative (beta =-0.60)

Below, there is a coefficient table in order to draw useful insights. Specifically, it shows the predictors in odds metrics. The odds values that are close to zero could be eliminated to improve the model.

	OR	**2.50%**	**97.50%**
job_Other	40.10874	26.85407	59.98913
job_services	24.04688	15.7768	36.69165
job_admin.	36.67274	24.53833	54.88958
`job_blue-collar`	21.54908	14.31322	32.4816
job_technician	33.37241	22.2007	50.23651
poutcome_nonexistent	0.110665	0.093958	0.130235
poutcome_failure	0.08365	0.069295	0.100843
poutcome_success	NA	NA	NA
`age_bin_(40,100]`	0.964643	0.889856	1.04552
`age_bin_(15,40]`	NA	NA	NA
`dur_bin_[200,400)`	0.141928	0.129142	0.15584
`dur_bin_[0,200)`	0.038101	0.034218	0.042356
`dur_bin_[400,5.1e+03)`	NA	NA	NA
`emp.var.rate_bin_[1,1.5)`	0.14704	0.125969	0.171042
`emp.var.rate_bin_[-4,1)`	NA	NA	NA
season_spring	0.293302	0.202106	0.426417
season_summer	1.030116	0.701609	1.515252
season_autumn	0.26129	0.179091	0.381852
season_winter	NA	NA	NA
cluster	**0.549983**	**0.448365**	**0.675902**

Table 2.2 Table with odds ratio metrics from final model

Confusion Matrix & Statistics

	Reference	
Prediction	0	1
0	**13018**	**581**
1	**1398**	**957**

Accuracy :	**0.876**
95% CI :	(0.8707, 0.881)
No Information Rate :	0.9036
P-Value [Acc > NIR] :	1
Kappa :	0.4245
Mcnemar's Test P-Value :	<2e-16
Sensitivity :	**0.62224**
Specificity :	0.90302
Pos Pred Value :	**0.40637**
Neg Pred Value :	**0.95728**
Prevalence :	0.09640
Detection Rate :	0.05998
Detection Prevalence :	0.14761
Balanced Accuracy :	0.76263
'Positive' Class :	**1**

Table 2.3 Confusion Matrix from train model on train set

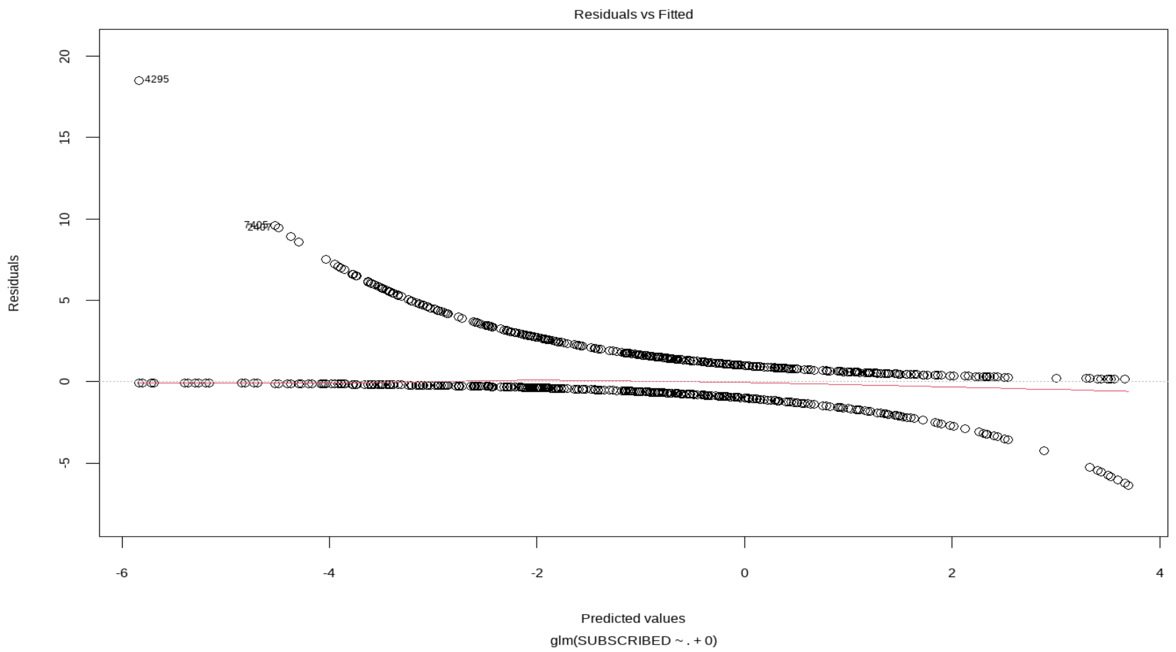


Figure 2-2 Residuals versus fits plot

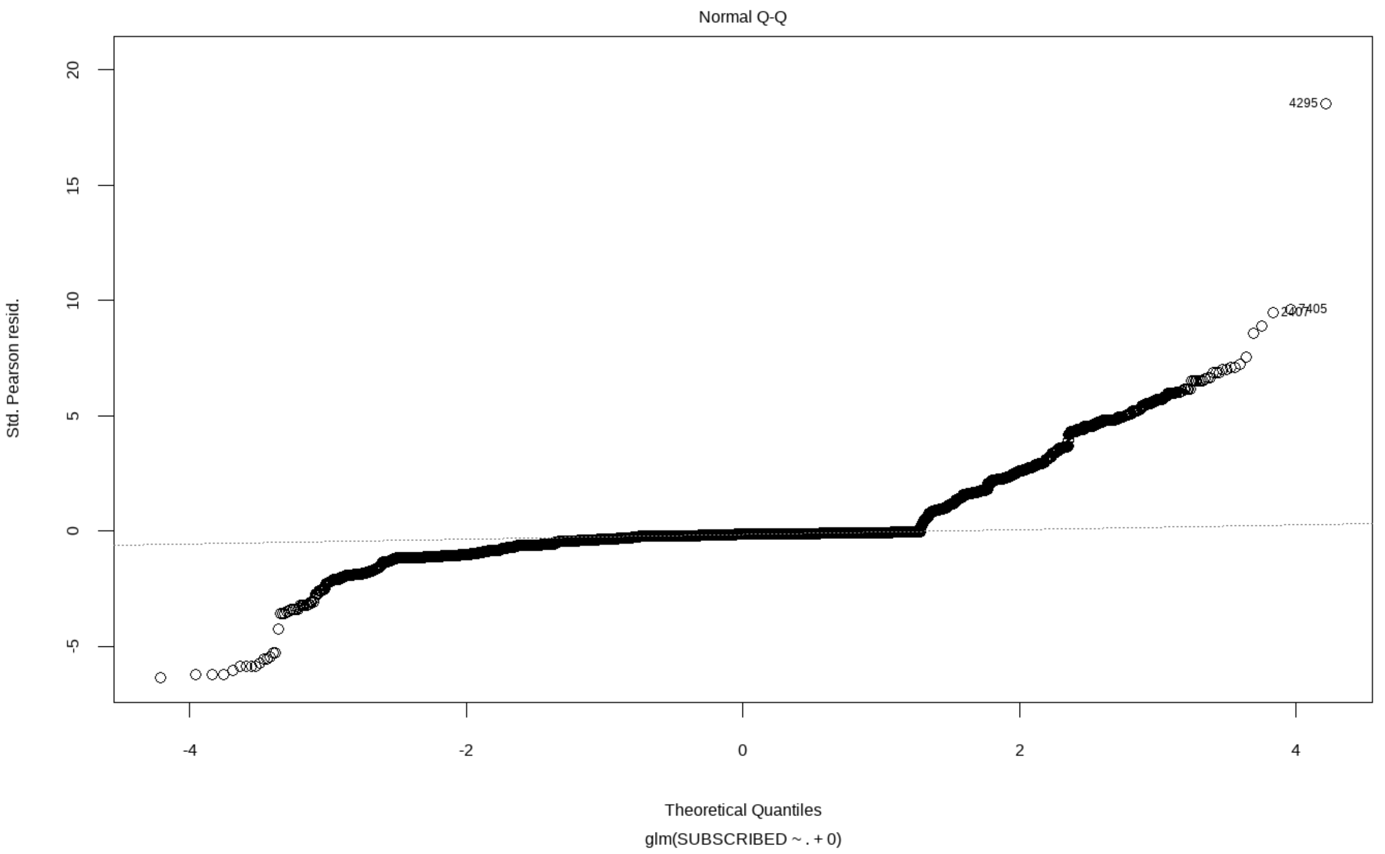


Figure 2-3 Normal QQ plot

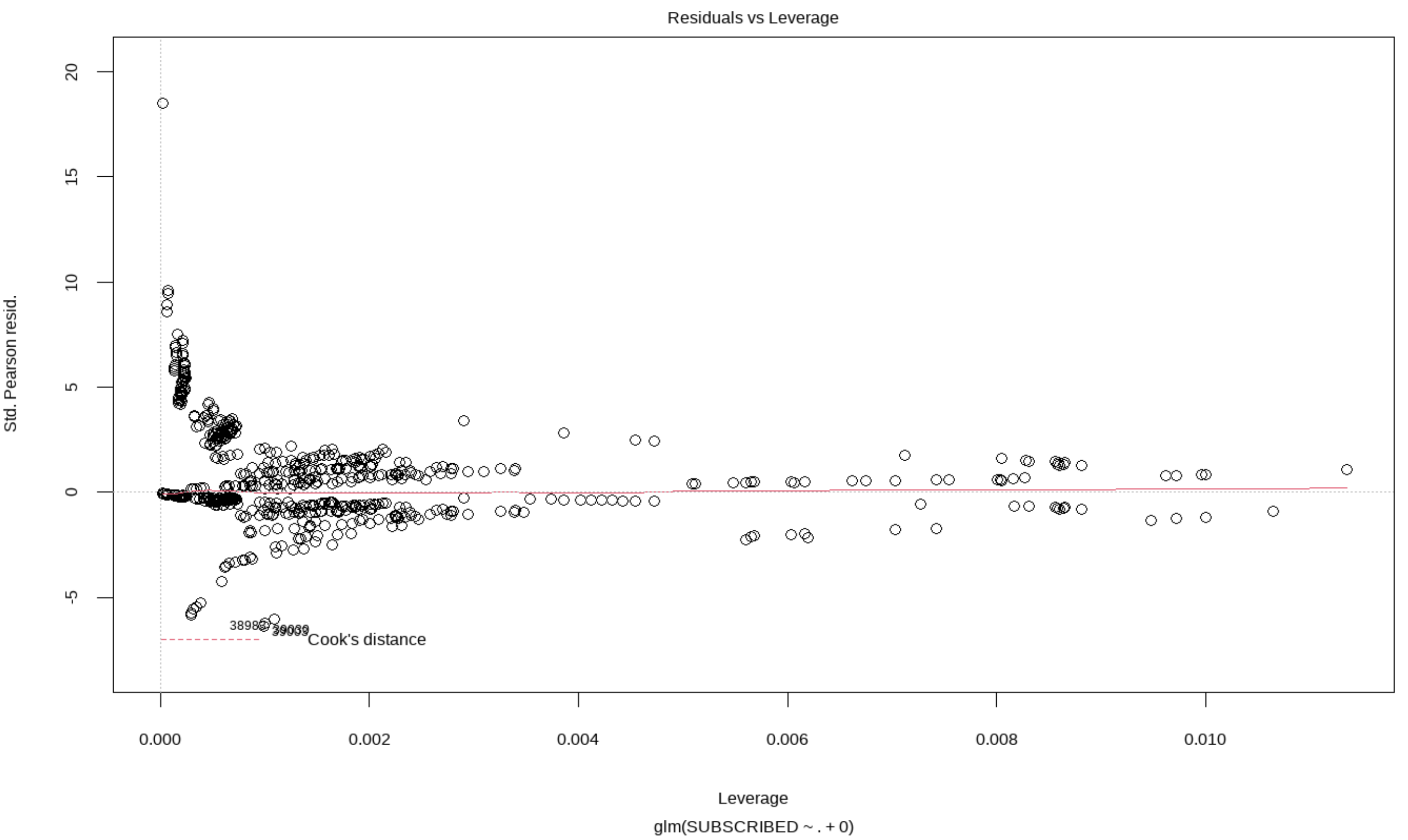


Figure 2-4 Residuals vs Leverage plot

3. Conclusions

In the first section we implemented three different classification methods with good results in accuracy and prediction of subscriptions. We experimented with many different methods in our code (r file attached) but the three presented gave the most sufficient results.

In the second part, we again experimented with many different combinations of different variable selections and clustering methods but the one we chose to use to represent our data was the only satisfactory beyond all other methods.