January 21, 2016

Numerical Machine Learning

Visa Premier Classification: A Marketing Application

Gabriela HERNÁNDEZ

dkm²
Erasmus Mundus
master course in
Data Mining and
Knowledge Management
a european master

Contents

1 Introduction

In this study, we worked with the Visa Premier dataset that consists on the information about the customers' behavior of a bank. It specifies, for instance, their movements, account balances, personal information and whether they have the Visa Premier card or not. This is a premium payment card that seeks to strengthen the close relationship with the bank to retain wealthy clients. This dataset describes the behavior of 1073 clients by using 48 variables.

With this information, throughout this work, we created four different classification models, and among them we selected the best classification model to estimate the probability of a new customer to buy this card. Among the classification models, three of them only use continuous features and one of them uses both continuous and categorical features.

This work was done by using R 3.1.2 with the following packages: MASS, mclust, Rmixmod, caret, pROC.

The report is structured as follows: in section 2, the data preprocessing procedure is described. Section 3, briefly summarizes the method used for splitting the data into training and testing sets. In the section 4, we present three different classification models constructed only by using continuous features while 5 shows one model built by using heterogeneous data (continuous and categorical features). In section 6 we summarize the prediction results of all the models in order to compare them, and finally in section 7 we draw our final conclusions and thoughts.

2 Data Preprocessing

The first thing we did was to identify the missing values in our dataset. According to the features information, the variable *sitfamil* have the following categories: Fmar, Fcel, Fdiv, Fuli, Fsep, Fveu. However, in the dataset we found that it includes another category 'F.', thus we treated the values 'F.' in *sitfamil* as missing values. For the other features, the values labeled as "." were considered as missing values.

Afterwards, we identified the continuous features from the categorical ones, and we created two new datasets accordingly to the kind of feature. We stored the continuous features in a dataframe named "data_continuous" and the categorical features in another dataframe called "data_categ".

Continuous Features Preprocessing

For the "data_continuous" dataset, we replaced the missing values in each variable with its mean.

Subsequently, we looked for the constant features to be removed, and we detected that the variable *nbimpaye* was constant among the whole dataset, whereas the variables *tbon* and *nbbon* were almost constant, except for the second observation, so we treated the second record as a potential outlier and it was removed from both datasets ("data_continuous" and "data_categ"). Following, we removed the features *nbimpaye*, *tbon* and *nbbon* from the "data_continuous" dataset.

Finally, we just checked that all the variables in this dataset were identified correctly as continuous features.

Categorical Features Preprocessing

For the "data_categ" dataset, we first removed the following unnecessary variables:

1. *cartevpr*, it was a duplicated of the *cartevp* variable.

2. *sexer*, it was a duplicated of the *sexe* variable.

3. *matricul*, it is and ID for identifying each client and it is not needed for the classification task.

Afterwards, we treated the missing values, for each variable we replaced them with its mode.

Finally, we checked if each feature in this dataset was identified with the correct type (factor) and we found that the variable *departem* was registered as integer so we converted it to factor.

3 Data Splitting

We split the data into 70% training and 30% test sets:

Training Set: this dataset is used to estimate and to select the parameters of the models.

Test Set : this dataset is used to evaluate the performance of our models.

For this task, we set a seed in order to have reproducible results.

4 Classification based on Continuous features

In this section, we review the different models used to classify potentially customers to buy the "Visa Premier" card, with respect to only the continuous features. We evaluate each model with the following different metrics:

1. Confusion matrix: In order to check true positives, false positives, true negatives and false negatives predicted values.

2. Misclassification error: This metric can be used for selecting the best model.

3. Accuracy: To have an big picture or overview of the model's performance.

5. Specificity and Sensitivity: Since we have 2–class classification models, these measures are also good to test them.

6. ROC curve: With two classes the Receiver Operating Characteristic (ROC) curve can be used to estimate performance using a combination of sensitivity and specificity. The ROC curve plots the sensitivity, i.e. true positive rate by 1- specificity, i.e. the false positive rate.

7. AUC (Area Under the ROC curve): This is a common metric of performance.

4.1 Linear Discriminant Analysis Model

We carried out the Linear Discriminant Analysis by using the **MclustDA** function of package **mclust**. For selecting the best model among the parsimonious models, we used the training set in order to estimate their parameters and we evaluated the misclassification error for each model on the test set. We discovered that the model giving us the best performance, i.e. the smallest misclassification error on the test set, was the EEE = Ellipsoidal, Equal volume, shape, and orientation. Hence, we used this model to predict *cartepv* with the continuous features and prediction results are summarized below.

Performance Evaluation:

1. Confusion table

		Observed	
		Cnon	**Coui**
Predicted	**Cnon**	198	13
	Coui	41	70

2. Misclassification error = 0.1677

3. Accuracy = 0.8323

4. Sensitivity = 0.9384

5. Specifity = 0.6306

6. AUC = 0.8357

7. ROC Curve:

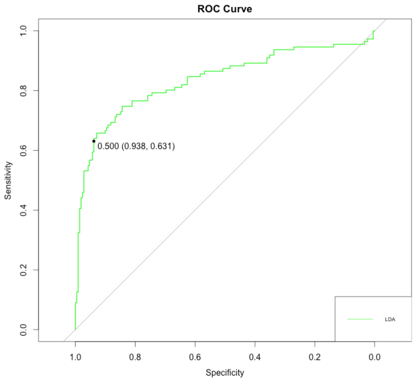


Figure 1: ROC Curve for LDA Model

4.2 Quadratic Discriminant Analysis Model

We performed this analysis by using the function **qda** in the package **MASS**. In this model, first we had rank deficiency issues due to exact multicolinearity, to solve this, we needed to jitter the data. Below, we summarize the prediction results for this model.

Performance Evaluation:

1.Confusion Table:

		Observed	
		Cnon	**Coui**
Predicted	**Cnon**	183	42
	Coui	28	69

2. Misclassification error = 0.2114

3. Accuracy = 0.7826

4. Sensitivity = 0.8673

5. Specifity = 0.6126

6. AUC = 0.8509

7. ROC Curve:

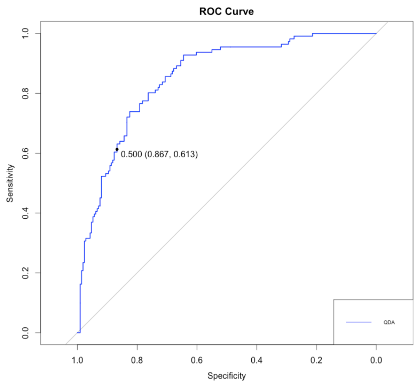
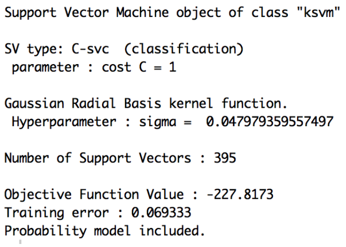


Figure 2: ROC Curve for QDA Prediction Model

4.3 Support Vector Machine Model

For this model, we used the **train** function in the **caret** package. This function contains 147 different models. It also has different resampling methods, metric and facilities for parallel processing. We wanted to take advantage of these capabilities and we repeated three 10-fold cross-validation, then the combination with the optimal resampling statistics was chosen as the final model and the entire training set was used to fit the final model. In figure 3, the parameters of the final svm model are shown, as we can observe the training error is very small compared with the error in the testing set, so by doing the 10-fold cross-validation in a small dataset tends produce overfitting conflicts. Nevertheless, when we applied the svm with our normal training set the overall predicted results obtained were worse, so we decided to kept this model.



```
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.047979359557497

Number of Support Vectors : 395

Objective Function Value : -227.8173
Training error : 0.069333
Probability model included.
```

Figure 3: SVM final model

Performance Evaluation:

1. Confusion matrix:

		Observed	
		Cnon	**Coui**
Predicted	**Cnon**	186	27
	Coui	25	84

2. Misclassification error = 0.1615
3. Accuracy = 0.8385
4. Sensitivity = 0.8815
5. Specifity = 0.7568
6. AUC = 0.8956
7. ROC Curve:

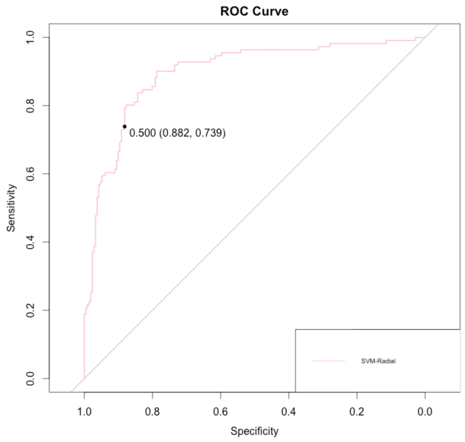


Figure 4: ROC Curve for SVM-Radial Prediction Model

5 Classification based on Continuous and Categorical Features

In this section, we present the model used to make the prediction task with both kind of features, continuous and categorical. With this aim, we merged the continuous and categorical features into one set named X. We split this dataset, in the same manner we explained in section 3, into X.train and X.test. The metrics used to evaluate the performance of our model were the same explained in section 4.

5.1 Mixture Model

We implemented this model by using the functions **mixmodLearn** and **mixmodPredict** of the **Rmixmod** package. One of the main advantages of these functions is that they have the capability to deal with heterogeneous data, i.e. data containing continuous and categorical features, without needing to convert or transform continuous features into categorical or vice versa. For achieving this, we centered, scaled and applied PCA to the 35 continuous features of the training set, in order to reduce dimensions. We kept 23 components to capture 95 percent of the variance. Afterwards, we applied the same transformation obtained in the training set to the continuous features on the test set. The categorical variables remained the same. The prediction results achieved by using this model are shown below.

Performance Evaluation:

1. Confusion matrix:

		Observed	
		Cnon	**Coui**
Predicted	**Cnon**	189	58
	Coui	22	53

2. Misclassification error = 0.2484

3. Accuracy = 0.7516

4. Sensitivity = 0.8957

5. Specifity = 0.4775

6. AUC = 0.8271

7. ROC Curve:

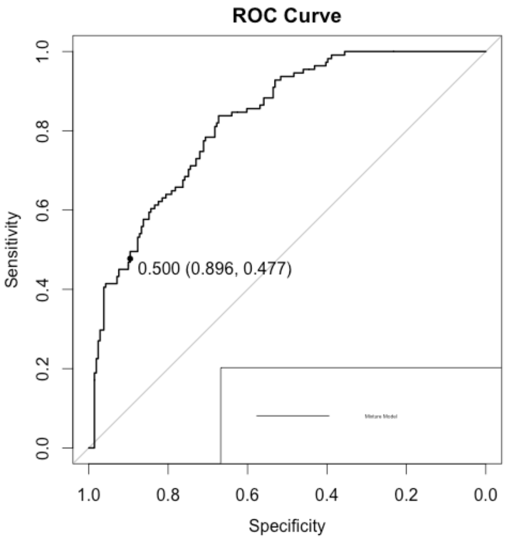


Figure 5: ROC Curve for Mix Model Prediction Model

6 Results

In this section, we summarize the results obtained in all the proposed models in order to compare them, and emphasize some observations.

Table 1: Comparison of different metrics among the models

Model\Metrics	**Misclassification error**	**Accuracy**	**Sensitivity**	**Specificity**	**AUC**
LDA	0.1677	0.8323	0.9384	0.6306	0.8357
QDA	0.2114	0.7826	0.8673	0.6126	0.8509
SVM Radial	0.1615	0.8385	0.8815	0.7568	0.8956
Mixture Models	0.2484	0.7516	0.8957	0.4775	0.8271

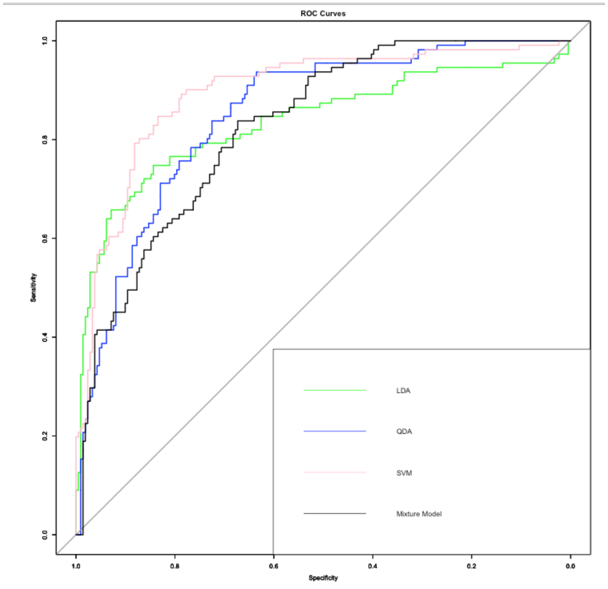


Figure 6: Comparison among the ROC curves of the models

As we can see in table 1, the SVM-Radial model is the one that performs the best among all the evaluation measures. Meanwhile, the mixture model gets the worse results, except for the sensitivity, it means that it has a high rate of positives events that are actually correctly identified as such, in this case it can be translated as the people who are identified as not Visa Premier card holders are actually no visa premier card holders, although the specificity is very low, i.e. this model tends to classify a significant proportion of card holders as no-card holders. Quadratic Discriminant Analysis get very similar results to Mixture model, however it tends to perform a little bit better. Finally, LDA gives a very reasonable result, taking into account that the preprocess of the variables for this model was quite easy with respect to the others.

These results are fairly consistent with the ROC curves shown in figure 6. As we can observe, the SVM-radial model is the one closer to the coordinate (1,0), please notice that in our ROC plots the x-axis ranges should be interpreted as (1-specificity), whereas QDA and mixture model have very similar performance.

7 Conclusions

One of the most surprising results we got was for the Mixture Model when we used heterogeneous data, we expected this model to have a better performance, even though we looked forward to take advantage of its unique property to deal with mixed type of data trying to make further preprocess in the data or change some variables, the function crashes, i.e. it did not support it. We do believe that there are more ways to improve this model and we would like to work on tuning it in a future. The Linear Discriminant Analysis was a good model, easy to implement with not further preprocess nor sampling techniques and with good enough results, giving the fact that we only considered continuous features. Support Vector Machine is a well recognized robust classifier, the only downside is the time it takes for training, we consider that doing 10-fold cross validation helped to improve the predicted results in this model. The Quadratic Discriminant Analysis could not be a good model for this dataset, since it has exact colinearities among the covariables, so adding jitting to the data could have made the results worse.