Classification on bank marketing data

Martí Cardoso i Sabé, Meysam Zamani Forooshani
marti.cardoso@est.fib.upc.edu, meysam.zamani@est.fib.upc.edu

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH
FIB

Problem

We have data related with direct marketing campaigns of a Portuguese banking institution [1]. We want to predict if a client will subscribe a term deposit or not, so it is a binary **classification** problem. The dataset has **20 explanatory** variables (10 continuous, 7 categorical and 3 binary) and the binary response variable (called y). It has **41.188 observations**

Preprocessing

The main preprocessing tasks performed in this project are:

Dealing with missing values: Our dataset does not have *NA* values, but some variables have values with the meaning of being missings: *pdays* (we decided to discretize it) and categorical variables (we keep the *unknown* modality).

Transformation of variables: First, we apply logarithms to the variables that seem to be exponential (*campaign* and *previous*). And secondly, for the continuous variables that seem to be irregular, we discretize them (we will analyze if this discretization is useful for prediction tasks).

Feature selection: We used tests of independence (Fisher's F and Chi-square test) and we removed the variables that seem to be independent to the target (*loan* and *housing*).

Feature extraction: We applied some feature extraction techniques to our datasets. We tried two approaches:

1. Apply PCA to the concatenation of PCA using the continuous variables and MCA using the categorical ones.
2. Apply MCA to the dataset with all continuous variables discretized.

We will analyze if these new features are helpful for prediction tasks.

Datasets after preprocessing

After the pre-processing, we have generated four datasets (they will be analyzed in order to see the one that performs better):

- **D1:** Dataset with all continuous and categorical variables.
- **D2:** D1 but discretizing the continuous variables that are not Gaussian.
- **D3:** The projections of PCA applied to the concatenation of PCA and MCA (feature extraction approach 1).
- **D4:** The projections of MCA applied to the dataset containing all variables discretized (feature extraction approach 2).

References

[1] UCI Machine Learning Repository. Bank Marketing Data Set. `https://archive.ics.uci.edu/ml/datasets/bank+marketing`. [Online; accessed 03-04-2019].

Visualization

Using visualization techniques we made a plot in 2d of our dataset and we understood how the variables are related between them. We have used the two feature extraction approaches explained in the preprocessing to make the visualization, and with both, we found similar relations between variables (e.g. *education* and *job* are highly correlated) and that the two classes are difficult to classify in two dimensions, so our problem is not easy.

Protocol of validation

We split the dataset into learning and test sets using the **holdout method** (2/3 for learn and 1/3 for test). With the first one we select and create the best model, and with the second one we use it to get an estimation of how good or bad is the final model. Then, for the model selection step, we use the **10-fold cross-validation** resampling method on the learning set to compute the validation error, compare the tested models and select the best one. Also, as we have unbalanced classes, we decided to use the **F1 score** because it gives more importance to the minority class.

Model selection

We have tested several machine learning techniques that are suitable for our problem (mixed data and binary response). Most of the hyperparameters of each method have been optimized by testing several values (or a grid search). The following table shows the best results of each method (together with the dataset and hyper-parameters that perform better):

	F1	**Dataset**	Hyper-parameters
Logistic regression	0.7305	D2	
Lasso	0.7310	D2	$\lambda = 0.0002$
Ridge	0.7323	D2	$\lambda = 0.0373$
LDA	0.7125	D1	
Naive Bayes	0.7109	D1	
SVM	0.7331	D2	kernel=RBF, $C = 10^{-1}$,$\gamma = 2^{-2}$
MLP	0.7309	D1/D2	$neurons = 30$, $decay = 10$
Decision tree	0.7202	D4	
Random forest	0.7297	D1	$nTree = 63$

LDA and Naïve Bayes are the two methods that obtain the lowest results ($F1 \approx 0.71$), then the Decision tree ($F1 \approx 0.72$) and all others methods get an F1 score near to 0.73. In our experiments, mostly D1 and D2 perform better than D3 and D4, so the use of the projection of the PCA and MCA do not help on the prediction in our problem. Between D1 and D2, D2 seems to be a step above too, that means that our discretization of the irregular variables was useful. Finally, we decided to select the **SVM with the RBF kernel,** $C = 0.1$ **and** $\gamma = 2^{-2}$ **using the D2 dataset**. as the best method to predict if a client will subscribe or not.

Final model

Once we have selected the best method, we refit the model with all the training set ($D2$) and then we do the prediction of the test set. With these predictions, we can estimate the generalization error of the model because this data was not used in the training stage.
The confusion matrix of the test set (left) and some interesting performance statistics (right) are:

		Real labels no	Real labels yes
predictions	no	10.027	540
	yes	2.156	1.007

Accuracy	80.36%
Recall^{yes}	65.09%
Recall^{no}	82.30%
Precision^{yes}	31.83%
Precision^{no}	94.88%

The final model gets an accuracy not very high knowing that 90% of the observations are of type *no*, but this is the price that we pay to give more importance to the *yes* class. For our problem, we want to have a high recall on the *yes* case because we do not want to predict clients that will subscribe as a *no* (the bank will lose the client), and this recall on our model is not bad.
So, this final model could be useful for the bank company because it would filter a lot of real *no* subscription users and it would keep most of the *yes*.

Conclusions

In this project, we have performed a full machine learning process on a real dataset related to bank marketing. Doing it, we have learned how to develop a project of this type, starting by doing a pre-process of the data, then making the visualization of it, defining the protocol of validation, the model selection and finally creating the final model and estimating its generalization error.
Our final model does not have very high accuracy, but the recall in the *yes* label is not bad, and for this type of problems, it is interesting that this recall is as higher as possible. So, we think that this model could be useful for the bank company.