Jupyter notebook with Python code found in this link

Marketing Data

Predicting the Sale of Term Deposits

Ariel Serravalle z5264138

Marketing Data – Report

Abstract

A Portuguese bank has tasked us with analysing their Telemarketing data to predict the 500 customers most likely to purchase term deposits. These are financial instruments that pay a higher rate of interest in exchange for not being able to withdraw the funds for a fixed term. The data spans 2008-2010 and contains ~40k records on customer information, economic conditions, and call centre activity. The purpose of this report is to explain the process for building our model and analyse our findings.

Part 1 – Setup

Data scientists dream of a utopian society where information systems are perfect, and all data comes ready to be plugged into a neural network. Unfortunately, this is not the case, and we therefore require some modification to the dataset before we start building the model and making predictions. These include the setup of our analytics environment and packages, data cleaning, and dealing with missing values.

The analysis was done predominantly on Python, using Google Colab notebooks. The cloud storage, Google Drive Compatibility, and very large computing power made the task much easier. Python contains a variety of libraries needed for data manipulation, preprocessing, and model building. The main ones used were from the SciKit Learn machine learning package. SKLearn includes several models covered in the lectures, and evaluation metrics (AUROC, MSE).

Data Cleaning

Treatment of each variable is listed in Appendix 1. Variables which required no cleaning had no missing values, outliers, or irregularities. Cleaning was performed in a batch using a function which encoded categorical variables (binary and multiple), censored the *pdays* variable, and created a new *date* variable.

Table 1: Data Cleaning methods on each variable

No cleaning	ID, age, contact, year, month, day_of_week, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, nr.employed
Change Type	Campaign
Binary Encoding	Default, Housing, Loan, Y
Imputation	Job, marital, education, default, housing, loan
Regularisation	Pdays, poutcome, previous

Right censoring is necessary when the experiment does not run during an event we measure. In this case, *pdays,* indicating the number of days since previous contact, was set to 999 when the client hadn't been contacted. 97% of *pdays* values were like this. An MIT paper [2] suggests top-coding, or observing x_i only when it is less than a boundary ξ. In this case ξ is 22, the maximum value for *pdays* excluding 999. We also add another indicator variable d_i, *called* for variables that are censored.

KNN Imputation of Missing Values

The final step was to remove missing values. Based on our research[1] we deemed the best way to do this was through KNN imputation. This is because a very small portion of values were missing (see fig 1). The decision was corroborated by *Roychoudhury et al*. R can do this in one line of code, after we had converted all "unknown" values to NaN.

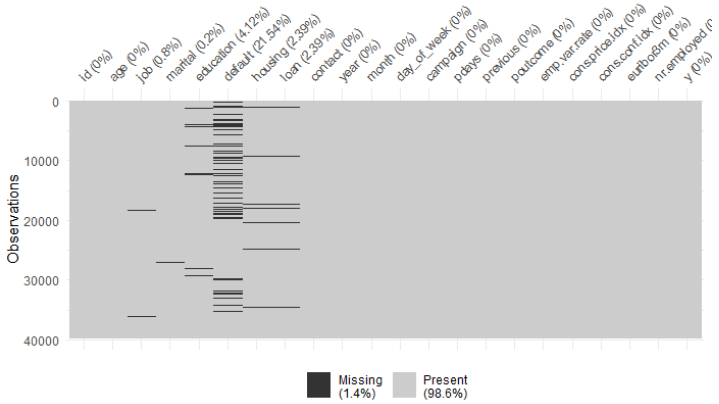

Fig 1: Highlighted missing values in the dataset

Before we continue, note:

$$P(success|missing\ data) = 1.7\%$$

$$P(success|no\ missing\ data) = 8.2\%$$

This means that data doesn't go missing at random. The absence of data likely means the salesman didn't get to the point in the conversation where they would collect that information. This would explain the discrepancy. Thus, we will create an indicator variable for observations which were imputed[7].

Part 2 – Preprocessing

Once the dataset is in the desired format, we begin exploration and preprocessing. Here we discover all the nuances of the data and keep in mind the adage 'garbage in garbage out'. That is, even the most advanced Statistical Learning algorithms will be hindered by poorly processed data. And even simple models like logistic regression can be made very accurate with proficient feature engineering.

Exploratory Data Analysis (EDA)

Descriptive statistics allowed us to understand the relationship between the data and the reality. We can classify the variables into three types – call centre activity, customer demographics, and economic conditions. EDA should answer questions about these three types of variables and guide our further analysis.

The average client was 40 years old. Most owned homes had a university degree and were married. The top three jobs were in admin, blue collar work, and technicians. One begins to paint the picture of their demographic as professionals and family men. We must ask ourselves if the sample space is sufficiently varied for us to be able to accurately classify the customers. For example, since only 0.05% of targets we illiterate, information on that class will be less accurate.

We should also consider if customers seem more valuable because we target them more or because they have a high probability of buying term deposits. It is here that we discovered the target variable imbalance (<10% of calls were successful). This indicated that we would need to use oversampling later.

	y		success_rate
	sum	count	
previous			
0	2907	34985	0.083093
1	762	4181	0.182253
2	199	502	0.396414
3	49	86	0.569767
4	14	20	0.700000

Fig 2: More calls leads to more sales
P(Success | Previous = x)

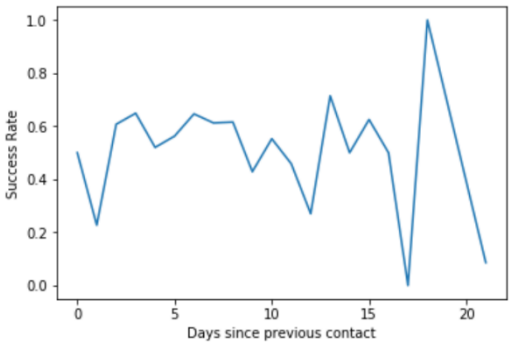

Fig 3: Follow-up time and success rate

We also see that the actions of the call centre play a major role in the success of the call. Of the eight factors most correlated with success, four are *previous, poutcome_success, called* and *pdays*. Refer to Appendix 2 for the full list. What this implies is that the rate of success increased with the number of previous contacts. Customers that had been called recently were more likely to buy than those who had not.

Table 2: Top 8 variables by correlation

Variable	Corr (y)
nr.employed	-0.299678
emp.var.rate	-0.280789
euribor3m	-0.27006
called	-0.264253
pdays	-0.26094
cons.price.idx	-0.206968
previous	0.17257
poutcome_success	0.258168

Economic indicators had a disappointingly large effect. The above findings can be translated into business decisions (e.g. be more proactive in following up with clients). The employment and Euribor data are factors outside of our control. However, we should still be aware of their effect on sales so that we can prepare accordingly. We cannot delve further into this without an analysis of the Global Financial Crisis, which began as this data was being collected.

Global Financial Crisis (GFC)

Combining year and month values allowed us to observe sales over time. We are particularly concerned with the effect of the GFC. We see that most of this data came from before the GFC, with low success rates. Call volume drops for ¾ months in October 2008 – March 2009, and success rates vary greatly. Calls made after 2009 and onwards have a much higher success rate but lower volumes.

According to *Reinforcement Learning and Savings Behavior*[8], there is 'an upward force on aggregate savings rates following a positive equity market return (and the reverse for a negative equity market return)'. Therefore, we would expect sales of term deposits to be higher in 2008. Another interesting anomaly is the success rate is negatively correlated with the EURIBOR – less people buy term deposits when they offer higher returns. This is a puzzling and counterintuitive outcome.

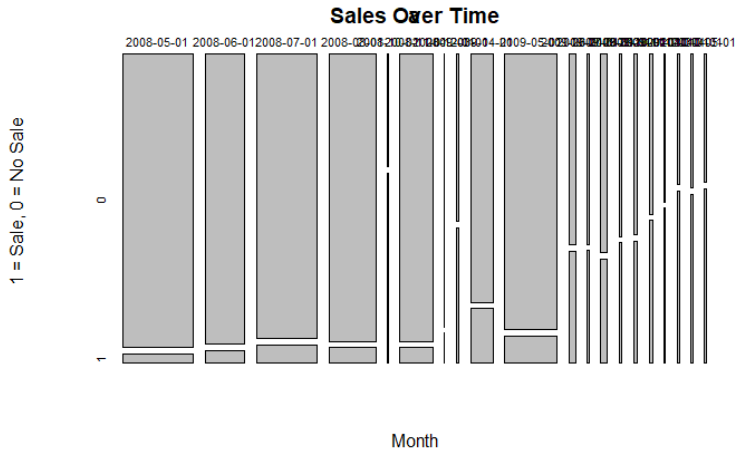

Fig 4: Success and Failure of calls over time. Thickness represents amount of data

Dealing with this variable is not so simple because the *Evaluation* dataset is for 2010 onwards – not included in the training set. Based on correlation, *date* is not a very powerful predictor. A lot of the time sensitive information is also captured in the economic data (particularly the Euribor). Thus, we will go for the simplest approach which is to convert the *month* and *year* variables into a *Timestamps* format, and then into a float.

Imbalanced Target Variable - SMOTE

He and Garcia suggest many techniques to deal with imbalanced data including random sampling and informed under sampling. The former involves repeat sampling of the minority class, random subset of the majority class. The latter uses supervised learning to systematically select which observations to under sample from the majority class. These methods are good to save computing power.

However, our use of cloud computing meant there was no need to conserve computing energy. The SMOTE method uses a KNN approach to consider the most similar observations from each sample. Then oversample those to create a symmetric data frame. A GitHub repository[3] gave us a simple way to do this, and we found several papers[5, 6] which gave it validity. The decision was thus quite simple.

Dimension Reduction - PCA

At the start of this stage, all of the variables were relevant, accurate, and in the data type that the model requires. We now seek to reduce the dimension of this data from 44 variables so as to accelerate training the model. Principal Component Analysis (PCA) is a method for doing this using unsupervised learning. It creates orthogonal vectors (PCs) which partially explain the information in all variables. It will continue to do this for as many components as we specify.

We found this method preferable to outright eliminating variables through selection. This is because it was very difficult to see patterns in the data, since most of the columns were encoded categorical variables.

To determine the number of PCs we wanted to work with, we fit a linear regression on each level and chose the one that had the best MSE. Rather, the best within a certain range, as MSE tends to increase as PCs decrease. We found the ideal value was around 20.

Part 3 – The Model

On the front end, the predictive model is actually the simplest part of any prediction challenge. SKLearn has made all sorts of models uniformly accessible in a few lines of code. The bulk of the work here is then in determining which model is more accurate for this data. Then we run the algorithm on the testing data and output our predictions. Currently our data looks like 10 principle components that represent the original 44 X variables (after encoding).

Model Evaluation

The appropriate models that have been considered for classification are:

- KNN
- LDA and QDA
- Logistic Regression
- SVM
- Random Forest

To save time on this large dataset, a simple 25% validation set was used to evaluate the six potential models. With data of this size, a K-Fold cross validation would take much longer and not be significantly more accurate given the law of large numbers. Despite these measures, model training in a loop still caused freezes. Therefore, we had to split the work up amongst different cells – a less elegant solution.

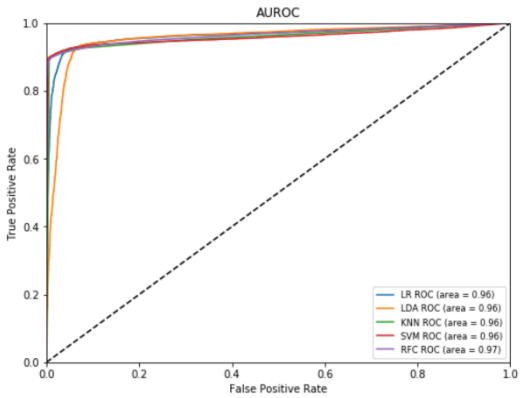

Fig 5: Training AUROC Curve for the 5 models evaluated here

	Model	AUROC	MSE
3	SVM	0.944742	0.055295
2	KNN	0.941277	0.058755
4	RFC	0.938153	0.061879
0	LR	0.934952	0.065060
1	LDA	0.922032	0.077949

Fig 6: Evaluation metrics

We used AUROC and MSE as evaluation metrics, as per *Beleites et al*[4]. The results in figure 6 come from a 50% threshold, which is why they differ from figure 5.

Prediction

From the above results, we determine the best model for this project to be the Random Forest Classifier (RFC). This is because of its AUROC and computational efficiency (relative to SVM). We also believe this model will be best for irregular decision boundaries. This was expected from the outset as RFCs tend to be the most robust models, particularly when used with an ensemble method such as Light XGBoost.

The algorithm was taken from an open source XGBoost application on Kaggle[9]. The algorithm will continue training random forests and adjusting parameters with Gradient Descent. It will continue to do this until no major improvements in our CV metric is made for 100 epochs. This occurred at a validation AUC of 0.9715.

After training, the function outputs the submission, feature importance, and evaluation metrics. The first of these is a dataframe with the customer ID and their predicted probability of sale. We took the ID's with the 500 best values as our predictions.

Identify 500 clients with highest P(Buy)

Conclusion

Throughout our analysis and report we have kept in mind the connection between numbers and reality. That is, considered the types of clients and economic conditions that relate to the highest probability of sale. This has been discussed in the sections about the GFC and exploratory data analysis, and the discoveries made here determined the analytical techniques we used.

This is important because we have produced a predictive model with high accuracy (measured by CV error of XGBoost), but absolutely no interpretative value. We have made this concession because the task given to us was to predict. Should the Portuguese banking client desire more interpretation, we would be able to provide it with more EDA and feature engineering (interactions between terms was not explored), or with a simpler model such as LDA.

While we succeeded in analysing the impact of major economic variables such as Euribor, customer data was less valuable. This is because it evidently provided little predictive value, and we suspect this is because they were mostly categorical. Individually encoded variables rarely have a high correlation with target variables. We outlined the majority of customers were married, university educated, etc. but this will not be able to guide future strategic decisions.

Bibliography

1. Edwin, D. and Mark, V., 2013. An introduction to data cleaning with R. 1st ed. Netherlands: Statistics Netherlands.
2. Rigobon, R., 2007. Estimation with Censored Regressors: Basic Issues
3. Github. 2019. KMeans Smote. [ONLINE] Available at: https://github.com/felix-last/kmeans_smote. [Accessed 7 August 2019].
4. Beleites, C., 2013. Validation of Soft Classification Models using Partial Class Memberships: An Extended Concept of Sensitivity & Co. applied to the Grading of Astrocytoma Tissues.
5. I. Mani, J. Zhang. “kNN approach to unbalanced data distributions: A case study involving information extraction,” In Proceedings of the Workshop on Learning from Imbalanced Data Sets, pp. 1-7, 2003.
6. M. Kubat, S. Matwin, “Addressing the curse of imbalanced training sets: One-sided selection,” In Proceedings of the 14th International Conference on Machine Learning, vol. 97, pp. 179-186, 1997.
7. The Analysis Factor. 2019. How to Diagnose the Missing Data Mechanism. [ONLINE] Available at: https://www.theanalysisfactor.com/missing-data-mechanism/. [Accessed 7 August 2019].
8. CHOI, J., 2009. Reinforcement Learning and Savings Behavior. Journal of the American Finance Association
9. Kaggle. 2019. XGBoost in Credit Data. [ONLINE] Available at: https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction. [Accessed 7 August 2019].