Clients Classification

AI approach

Luiz Augusto de Carvalho  
 Ontario Canada  
 ziulcarvalho@gmail.com

FirstName Surname  
 Department Name  
 Institution/University Name  
 City State Country  
 email@email.com

ABSTRACT

Here we describe a project developed inside a call-center operation of a large bank in Brazil whose objective was to sell credit cards for the clients of the bank. Therefore, the idea was to identify which clients were more easily acessible through the phone and which ones were more prone to accept buying a credit card.

Note that in this context the objective is to minimize the sales effort focusing the sales effort into the right people (more easily reacheable through the phone and prone to buy a credit card).

Today there is no process in place to select the best canditates to call (easy to reach and easy to sell), the company basically loads the mailling in a dialler and calls all people as if they were equaly reacheable and had equal interest in the product being sold (In tis particular case credit cards).

CCS

Computing methodologies, Artificial intelligence, Logistic regression, Extra tree classifier, Ridge classifier, Decision tree, Random Forest.

KEYWORDS

Mask Linear Regression, decision three, Random Forest,.

INTRODUCTION

We will work with a dataset of 108.002 clients. We have labels for all these clients regards if they were reached and regards if they bought the product being sold (credit cards).

ANALYTICAL SCHEMA

This work has as objective to create an AI model that based on the historical data, profile clients identifying how easy is to access him and how prone he is to buy.

The underline assumption is that a client with given profile will tend to behave as people of similar profile.

The first issue identified the fact that the positive labels (reach people and sell) were extremely unbalanced:



This unbalance adds difficult to model the problem, we have very few samples of some classes to work with. Note that only 30,97% of the people was actually reached and less than 1% actually bought.

DATA EXPLORATION

As already mentioned the data provided encompasses 108.002 clients.

The labeling ("C:/AI/alphavox-05/Training.csv") file layout is as follows:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 108002 entries, 0 to 108001

Data columns (total 51 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Unnamed: 0 108002 non-null int64

1 level\_0 108002 non-null int64

2 Unnamed: 0.1 108002 non-null int64

3 index 108002 non-null int64

4 ID 108002 non-null float64

5 NOME 108002 non-null object

6 DATA\_NASCIMENTO 108002 non-null object

7 SEXO 108002 non-null object

8 IDADE 108002 non-null int64

9 RENDA (R$) 108002 non-null int64

10 SCORE 108002 non-null int64

11 RECÃÂNCIA 108002 non-null int64

12 Cidade 100200 non-null object

13 UF 100333 non-null object

14 CEP 107566 non-null float64

15 LABEL\_ACESSO 108002 non-null int64

16 LABEL\_SUCESSO 108002 non-null int64

17 TABULACAO 108002 non-null object

18 STATUS 108002 non-null object

19 FONE\_FINAL 104570 non-null float64

20 Tels\_O 108002 non-null int64

21 Tels\_A 108002 non-null int64

22 Qtde\_Enriq 108002 non-null int64

23 FONE1 108002 non-null int64

24 STATUS\_TELEFONE\_1 108002 non-null object

25 FONE2 96105 non-null float64

26 STATUS\_TELEFONE\_2 96105 non-null object

27 FONE3 77153 non-null float64

28 STATUS\_TELEFONE\_3 77153 non-null object

29 FONE4 58336 non-null float64

30 STATUS\_TELEFONE\_4 58336 non-null object

31 FONE5 43121 non-null float64

32 STATUS\_TELEFONE\_5 43121 non-null object

33 FONE6 30527 non-null float64

34 STATUS\_TELEFONE\_6 30527 non-null object

35 FONE7 21971 non-null float64

36 STATUS\_TELEFONE\_7 21971 non-null object

37 FONE8 15772 non-null float64

38 STATUS\_TELEFONE\_8 15772 non-null object

39 FONE9 11181 non-null float64

40 STATUS\_TELEFONE\_9 11181 non-null object

41 FONE10 7666 non-null float64

42 STATUS\_TELEFONE\_10 7666 non-null object

43 idade 108002 non-null int64

44 idade\_range 108002 non-null object

45 classe 108002 non-null object

46 tipo\_tel 108002 non-null int64

47 classe1 108002 non-null int64

48 idade\_range\_x 108002 non-null int64

49 area 108002 non-null int64

50 sexo1 108002 non-null int64

dtypes: float64(12), int64(20), object(19)

memory usage: 42.0+ MB

The column name contains the name of the pdf file and the columns label identifies which type of document it is.

The columns LABEL\_ACESSO and LABEL\_SUCESSO are the labels which identify the outcomes:

Label\_acesso: 1 positive (manage to reach the person) or 0 negative (was impossible to reach this person)

Label\_sucesso: 1 Positve (The person bought) and 0 negative (the person did not buy)

DATA PREPARATION

The data preparation encompasses several steps. These steps are basically of two three:

1. Filling gaps in the database
2. Grouping the parameters into classes
3. Converting parameters into numeric value

Initially we manage to fill the gaps in the columns sex and income (renda).

To complete the missing sex the strategy used was to create a database with the first names identifying the number of male and female using this name. Once we find a person where the sex is not identified the code checks the first name and fill the field with the most prevalent sex of this name.

To complete the missing income the strategy used was to create a database with the income associated to each zip code, each time we found a person whose income wasn´t informed we check the zip code of the person and based on the prevalent income of the area define the income of the person.

In the sequence we group the age into 15 categories and the income into 5.

In the sequence we identify if the phone number (FONE1) is a mobile of fix line (1-0) and the area code of the number.

Finally we convert the parameters income range and age range into numeric.

CREATE SYNTETIC VARIABLES

We created six synthetic variables based on the original database:

* Idade\_range-x: range of age (15 ranges between 15 years and 95 – 5 years per range)
* Classe1: Income range, we have 5 income ranges (1-5)
* Area: Area code of the telephones (FONE1) – the two first numbers of FONE1.
* Tipo\_tel: identifies if the number belongs to a mobile (in Brazil number starts with 9,8 or 7)

CONVERTING TEXT TO VECTOR

We also converted the name of the person into a vector with 200 features. In sequence we added/combine the eight features with the four synthetic variables with the name vector creating a vector of 212 features. We put it into a matrix.

PREPARING THE TRAINING AND TEST SETS

Once we did that we separate the database into two 80% for training our model and 20% to test if the model is predicting the outcome properly.

Proceeding in this way we manage to create a training set which can be fitted in several models. We used four:

1. Logistic regression (categorical)
2. Extra-tree classifier (random forest which instead using subsets of the data – bootstrap uses the whole data set) – extra trees choose randomically
3. Decision tree
4. Random forest

After we did that we ensemble them into only one combined model. The whole logic of this was to create a trained model which could be used to predict the labels clients. Therefore this project encompasses the following codes:

* Preparing the database
* Train the model (using five techniques)
* Make predictions.



As we can see, there are several intermediary databases. The final outcome is a csv file with the predicted labels.

EVALUATING EACH MODEL

During the training of the model, we are going to separate the original database into training and test in a proportion of (80% training and 20% test). That means we are going to use 86.401 samples to train out model and apply this trained model to predict the output of 21.601 samples. Doing that we are going to check how many of these 21.601 the model predict correctly.

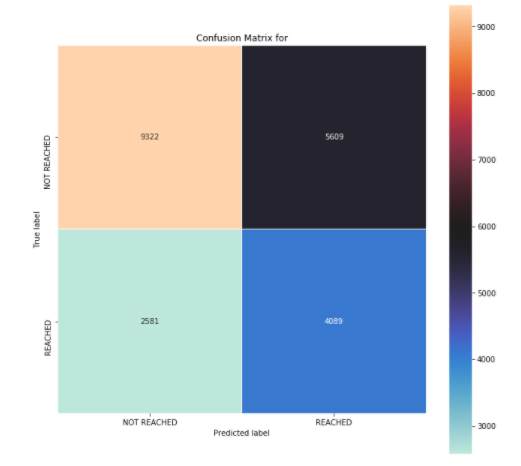
As mentioned we are going to use five techniques and ensemble them using majority vote:

* Logistic regression (categorical)
* Extra-tree classifier
* Decision tree
* Random forest
* Ensembled combination of the previous 4

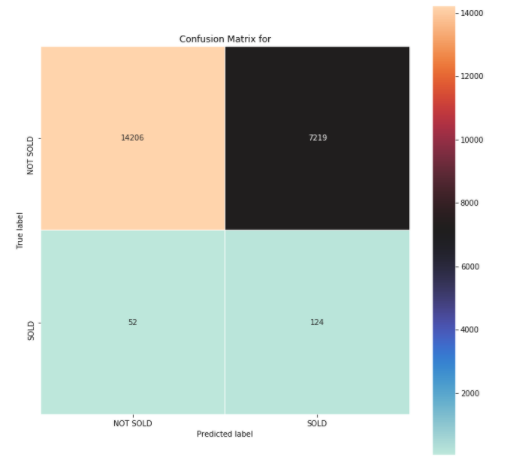
We are using GridSearchCV to find the optimized hyper-parameters.

Technique 1 – Logistical regression

Access:

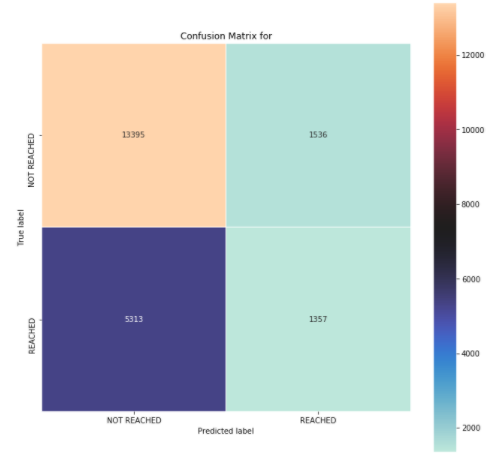


Sales:

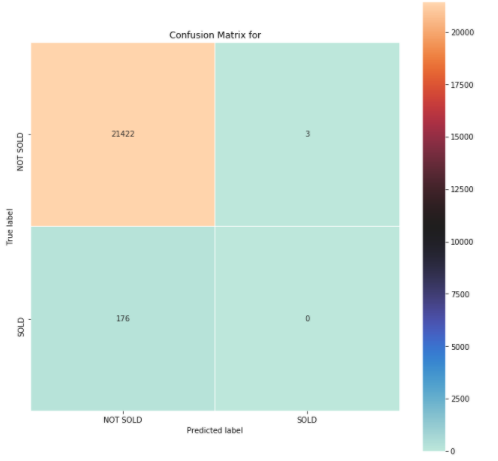


Technique 2 – Extra-tree classifiers

Access:

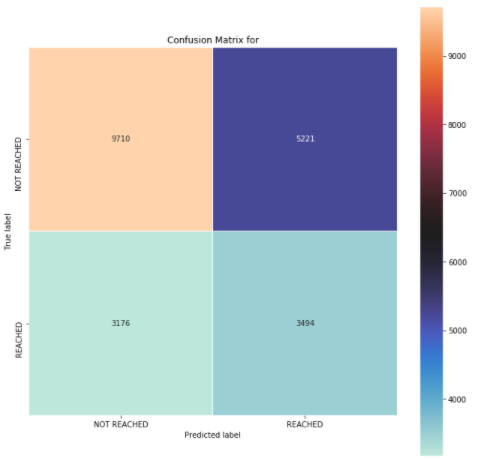


Sales:

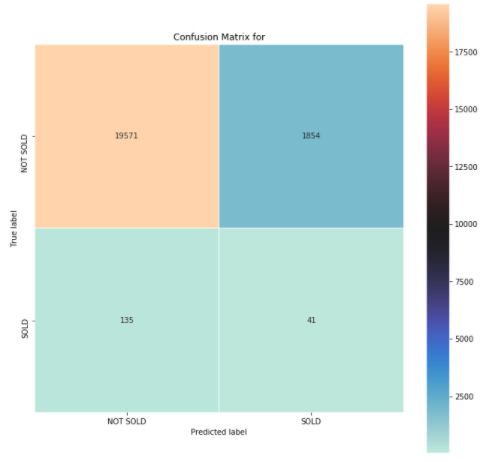
.

Technique 3 – Decision Tree

Access:

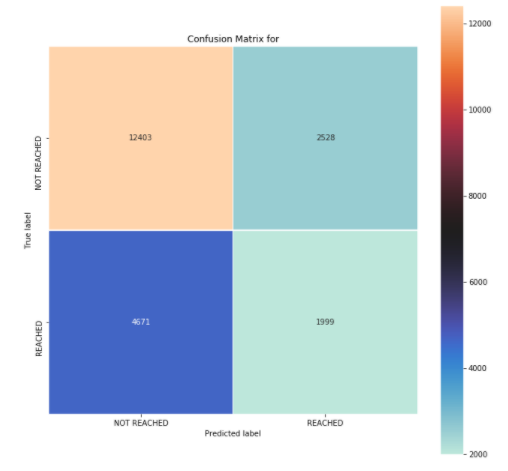


Sales:

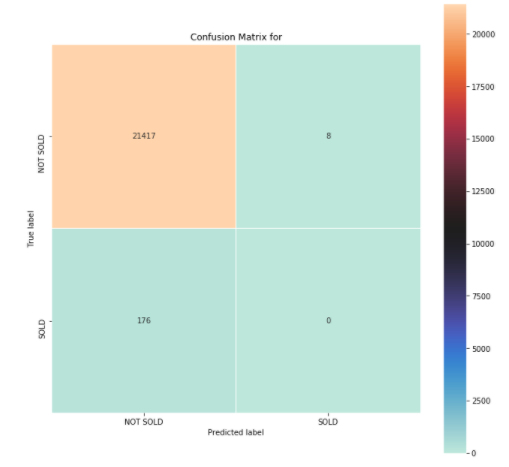


Technique 4 – Random forest

Access:



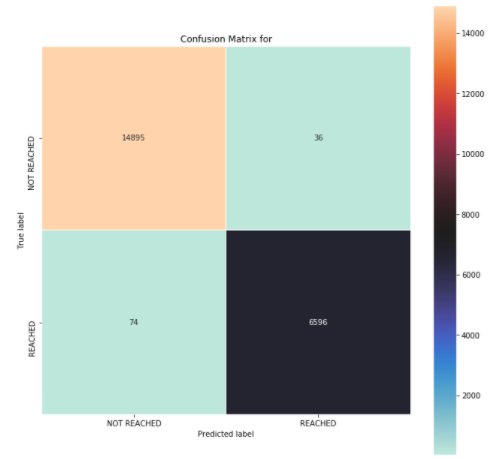
Sales:



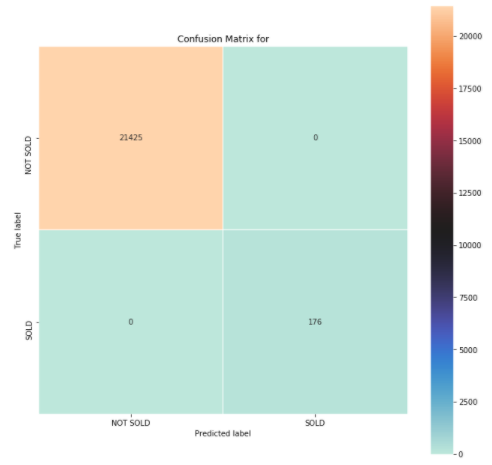
After we got the four models we combined them into one:

Technique 5 - Combined model

Access:



Sales:



Note that the combined model is away better predictor. It got it right 99.10% of the cases and got right to the minority classes 100% of the times. **A really good predictor**.

Here it is interesting to note that the combination of the models is better than the individual models.



Note that two of the models tend to generate false positives and two false negatives. In this tug of war when we combine the models they tend to stabilize in the right position, generating a very low error rate.

THE IMPORTANCE OF THE PARAMETERS

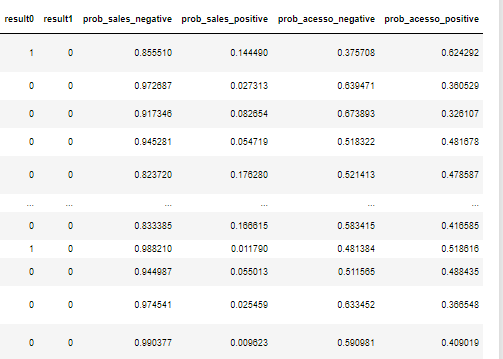
Here we are going to evaluate the impact of each one of the parameters in the decision process of the model. This is an important evaluation because tells us what is really being important when taking the decision to classify the client regards access and regards success in selling:



We can note that the area code , age, Tels\_A and quant\_enrri tend to be a much better indicators about if the person is going to be reached and going to buy than RECENCIA, SEXO, SCORE or RENDA (Income). That is a bit contra intuitive.

THE RESULTS

The results are presented through creating six columns the database being evaluated:



Note that the column “result0” represents if the code predicts that the person will be reached or not (1 – Yes and 0-No) , the colum “result1” predicts if the person will buy or not (1 – Yes and 0-No).

Of course these two columns would be enough if the final goal was to predict in advance who would buy a product and therefore just try to sell to these people. However, reality is abit more complicate, we intend to try to sell to these people discharging just the ones where the likelihood of success is very small.

In order to overcome this problem, and living to the managers of the sales operation the decision of “to whom” try to sell, we generated more four columns:

* Prob\_acesso\_Positivo => probability to reach the person
* Prob\_acesso\_negativo=>Probability to not reach the person
* Prob\_sales\_positve=>Probability to sell to the person
* Prob\_sales\_negative=>Probability not sell to the person