Document Classification

AI approach

Stan Taov  
 Ontario, Canada  
 stastaov@gmail.com

Luiz Augusto de Carvalho  
 Ontario Canada  
 ziulcarvalho@gmail.com

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

ABSTRACT

Large part of the clerical work in this accountant organization consists of checking documents, classifying them and extracting some basic data from them.

This process today is basically manual, and therefore time consuming and prone to human error.

The idea behind this project was to create a code capable of based on the scanned PDF files classify the documents into six types:

* Invoices (Notas in Portuguese)
* Personnel payments (RPA in Portuguese)
* Receipts (Recibos in Portuguese)
* Bank transfers (DOCs in Portuguese)
* Bills (Boletos in Portuguese)
* Others (Outros)

Note that the class others will encompass basically everything else not covered in the previous five.

Other important point is the fact that data will be extracted only from the three first classes. Here there is another difference the type of data extracted from each type has similarities but also differences:

* Data and value will be extracted from all three types
* Company number origin and destiny of the Invoice (Just the invoices)
* Company paying number and social Insurance number of the person receiving (Just for Presonnel payments)

CCS

Computing methodologies, Artificial intelligence,NLP

KEYWORDS

Mask Linear Regression, decision three, Random Forest,.

INTRODUCTION

We will work with a dataset of 15.731 pdf files containing 16.475 documents (We may have a pdf file with more than one page on it). Note that the conversion pdf to jpeg for aligning also is the opportunity where the pdfs which contain many pages are broken down into individual files. The new files have the original name plus a number.

We had labels of 4.448 of these documents. The label of document was identified as revisadox.csv and encompasses the name of the pdf file and its class.

ANALYTICAL SCHEMA

This work has as objective to create an AI model that based on the PDF documents converted to TXT classify them into six categories and extract specific data from three of these categories.

The first issue identified in the database provided was the unbalance between the Invoices and the other five classes:



This unbalance adds difficult to model the problem, we have very few samples of some classes to work with.

DATA EXPLORATION

As already mentioned the data provided by Krypton BPO encompasses 4448 usable documents. The Pdfs were forwarded as they were provided by their clients. That means we have all sorts of types of scanned methods: From pictures taken to upside down scanned docs. Very few documents were 100% aligned,

The labeling ("C:/AI/krypton-01/csv-doc/revisadox.csv") file layout is as follows:

RangeIndex: 4448 entries, 0 to 4447

Data columns (total 2 columns):

# Column Non-Null Count Dtype

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

0 name 4448 non-null object

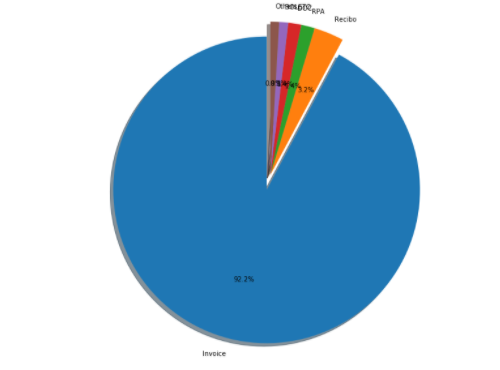
1 label 4443 non-null object

dtypes: object(2)

memory usage: 69.6+ KB

Five rows had Nan in the label

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



DATA PREPARATION

When cleaning the labeling file we identified five rows where the labels were missing.

In this particular project the data preparation encompasses basically aligning the pdf files, almost all of them had some small inclination (between -3 to +3 grades). We also found that some which were scanned up-sidedown or sideways. The code doesn´t treat these cases. It just aligns the ones whose inclination is between -90 and 90 grades.

To align the files we convert the pdf files into jpeg – using PIL library. Once aligned we convert the jpeg files into TXT using py.tesseract OCR.

Once we have the file converted into text we clean it through four sequences of cleaning

1. Clean – Removes the special characters
2. Clean1 – remove words with one and two characters
3. We convert the encoding to utf-8
4. Remove Portuguese stopwords

CREATE SYNTETIC VARIABLES

We create six synthetic variables which count how many times specific words appear in the document. Each class has its specific words:

DOC

['BANCO','FATURA','PROVANTE','VENCIMENTO','DEBIADA']

Invoices

['NFS','NOTA','FISCAL','PREFEITURA','MUNICIPAL']

Receipt

['RECIBO','RECEBEMOS','RECEBI','ALUGUEL']

Personnel payment

['RPA','NOMO','AUT','EMITENTE']

Bill

['BANCO','PROVANTE','BOT','DEBIADA','VENCIMENTO']

Others

['CONTRATO','AUXILIAR','DOCUMENTO','DANFPS']

Each time one of these words appear one of the six variables are added:

Sint1,Sint2,sint3,sint4,sint5,sint6

To get to these words we adopted the following logic:

1. Create a corpus with all text of each type of document
2. Count the most frequent used words in each type of document
3. Eliminated the words which appear in more than one type

CONVERTING TEXT TO VECTOR

The first step was to convert the text file into a vector with 200 features removing stop words in Portuguese. In sequence we added/combine the six synthetic variables creating a vector of 206 features. We put it into a matrix.

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

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. Ridge classifier (converts the labels -1 to 1) and uses regression
4. Decision tree
5. 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 of new documents. Therefore this project encompasses the following codes:

* Aligns the pdfs files (convert to jpeg)
* Converts the jpeg files into text files
* 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

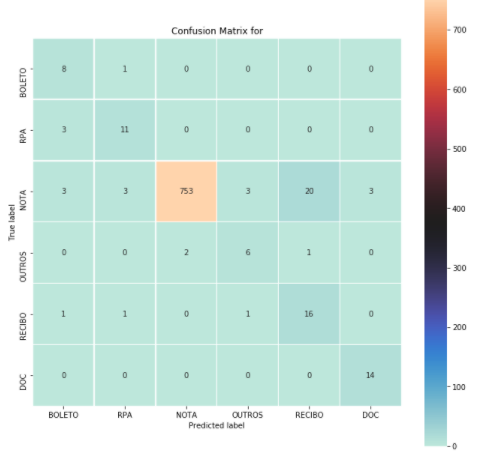
During the training of the model, we are going to separate the original database into training and test in a proportion of (80% training and 20% test). That means we are going to use 3.558 samples to train out model and apply this trained model to predict the output of 890 samples. Doing that we are going to check how many of these 890 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
* Ridge classifier
* Decision tree
* Random forest
* Ensembled combination of the previous 5

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

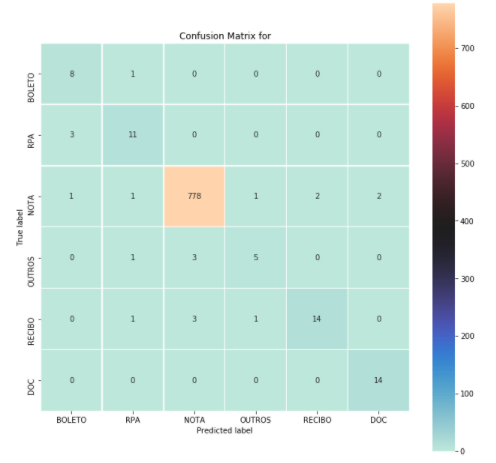
Technique 1 – Logistical regression



This model got it right 96% of the samples. 35 errors out of 890. Note that by far the class “OUTROS” got the higher percentage of errors: right just 66%.

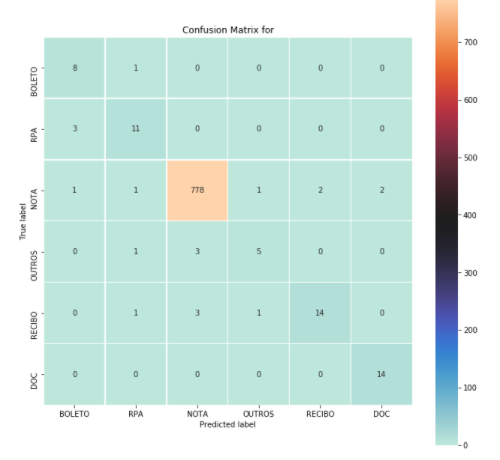
Technique 2 – Extra-tree classifiers

.



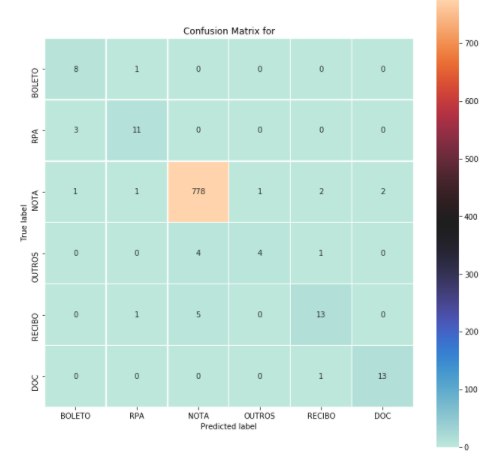
Note that this model seems to be much better than the previous one. It got a much higher percentage of right predictions 99 % but in terms of predicting the minority class “OUTROS” it was worse than the previous one. Got it right in just 56% of the cases. The other minority classes remained pretty much the same (A bit worse in class “RECIBO” and “DOC” from 3 to 4 and 3 to 5 wrong).

Technique 3 – Ridge-classifier



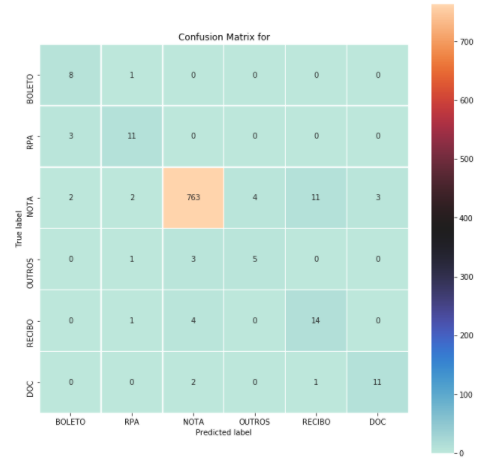
This model got exactly the same performance of the previous one.

Technique 4 – Random Forest



This model proved to be a little better than the previous one in terms of predicting the minority class, but not to the point to make a significative difference. Got 97.4% right and 83% right in the minority classes, although just 44% right in the class “OUTROS”.

Technique 5 – Decision tree

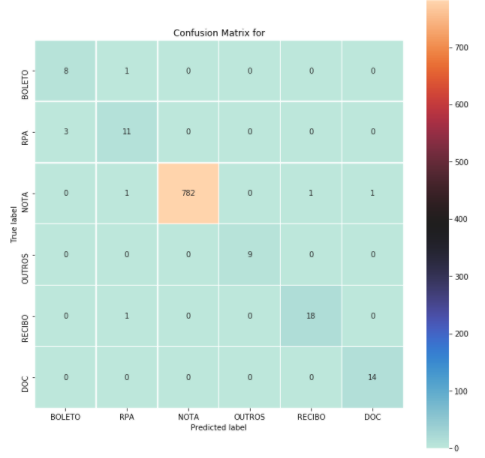


This model proved to worse to the previous one in terms of predicting the majority got 97%.33 right and 83% right in the minority classes, although just 33% right in the class “OUTROS”.

Here it is important to understand that in practical terms not being able to classify properly the class “OUTROS” is not a big problem. The three classes which actually matter are “NOTAS”, “RPA” and “RECIBO”. Those are the documents which actually matter, the other three types are basically bank demonstrations.

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

Technique 6 - Combined model



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 97% of the times. **A really good predictor**. Even the class “OUTROS” it got right 66% of the samples.

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



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 a document:

EXTRACTING DATA FROM THE FILES

As already mentioned before, in addition of classifying the documents we also extract some information from three of these sex labels. The information extracted differs a bit depending from the type:

RPA (Recibo de professional autonomo) – payment receipt for contractors: CPF of the receiver (Kind of Social Insurance Number in Brazil) and CNPJ of the payer (Company number in the Brazilian revenue agency) – the value paid and date of the document.

NOTAS (Invoices) and RECIBOS (receipts) we extract both CNPJ (company numbers in Brazilian revenue agency), the value paid and the document date.

The efficiency of this extraction process was as follows:

