Université Jean Monnet

Data Mining for Big Data: Project Report

Christopher JEAMME
Anthony DEVEAUX
Dimitri BRUYERE

January 2019

Contents

1 Introduction

The project consists in carrying out a study of a dataset. In this report, we present the structure of the dataset and the studies we made. The main objective of these studies is to provide a tool to automatically analyze and filter news articles in order to get only the ones that are related to the earnings of companies.

With this tool, we predicted the label for each article with unknown label that you can find in the file `test.cvs`. We also provide the source code of our studies.

2 Dataset

The dataset is composed of 5000 short news articles. These articles are in the XML format with the text inside a <BODY> tag which is the only tag of the document.

Of the 5000 articles, 4800 are labeled as 1 if the article contains information related to the earnings of a company or 0 otherwise.

3 Studies

3.1 Preprocessing

The first work we realized was to preprocess the data i.e. transform the original articles into vectors to give as input to our learning algorithms. We uses for this part functions of the python library `nltk` (Natural Language Toolkit)[1].

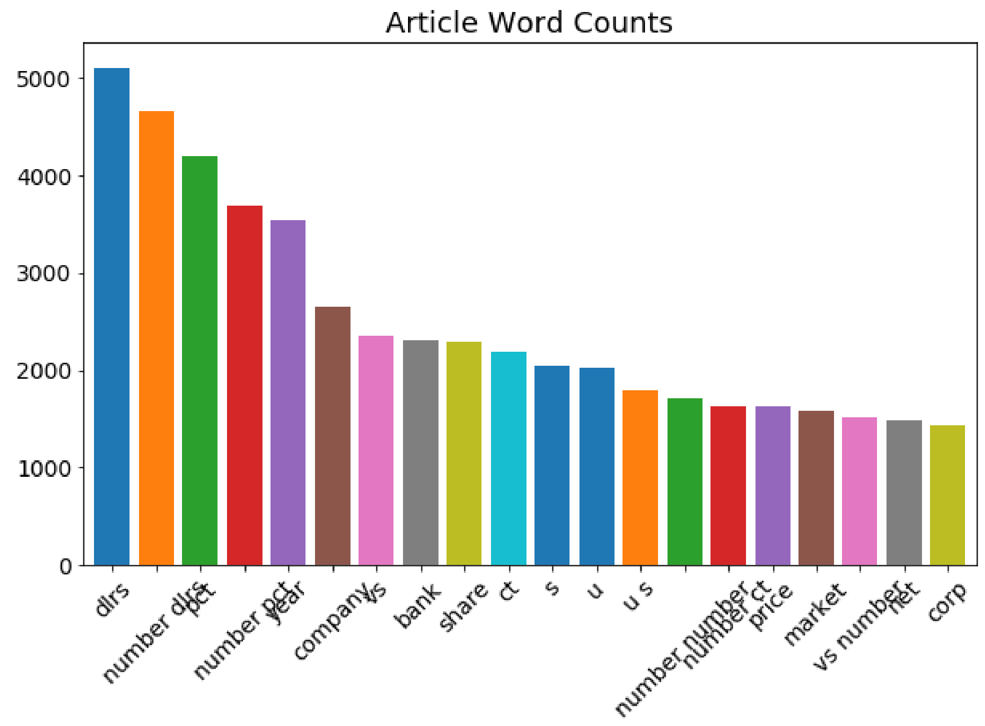
First, we remove punctuation symbols and we replace numbers by the string `number`. Indeed, we don't care about the value of the numbers, but we want to get the number of numbers in each article. This information can be very relevant, because we are interesting in the articles that are related to the earnings of companies, so in such articles they probably are several numbers.

Then, we realize a tokenization. This phase corresponds to split the text into a list of tokens. We realized a part-of-speech tagging on each tokens in order to apply an efficient lemmatization. This lemmatization consists on getting the inflected form of the word.

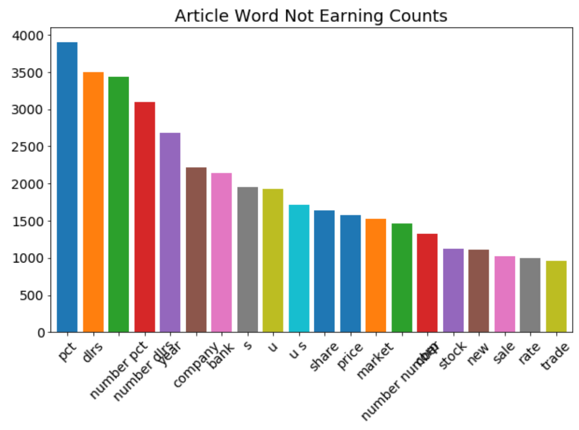
We also remove English stop words (most common words in English, which don't provide any relevant information on the article).

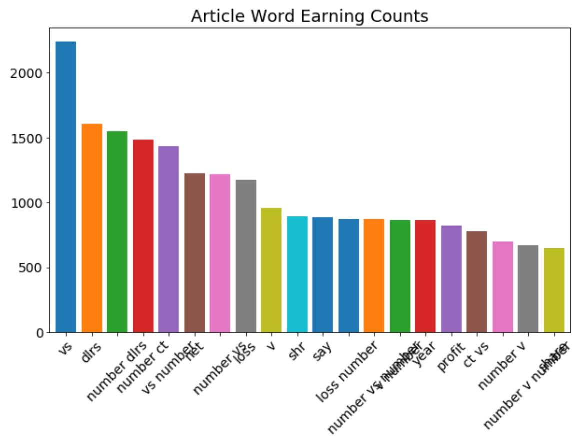
We decided to use N-Grams methods cause some usage like *num dlrs* to get vector of more efficient features.

In that way we could use a CountVectorizer to get a vector from each articles base on the number of words but this is not efficient so we use TfidfTransformer to get frequency instead of count which is more relevant.



As we can see in the graphic, inside our corpus lot of earning lexical but we need to see which word categorize better earning or not:

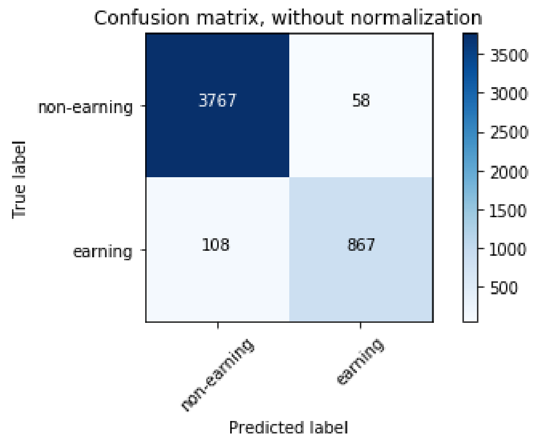


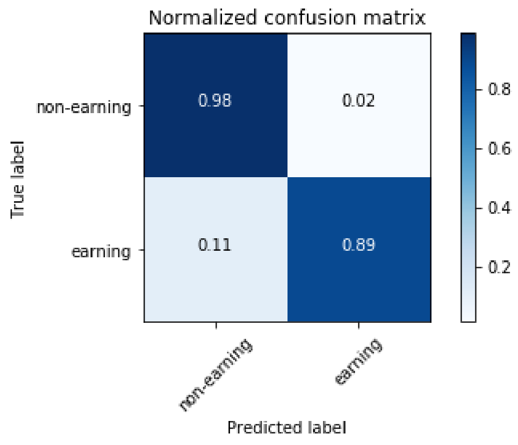


3.2 Predictions

We ran experiments on several classifier to select the one with the best accuracy to make predictions on the test set. For the experiments, we evaluate the predictions thanks to a K-Folds cross-validator (from scikit learn[2]) with 5 splits for each test.

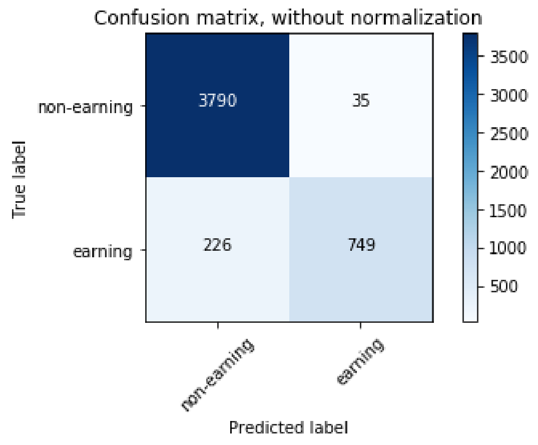
3.2.1 SGD Classifier

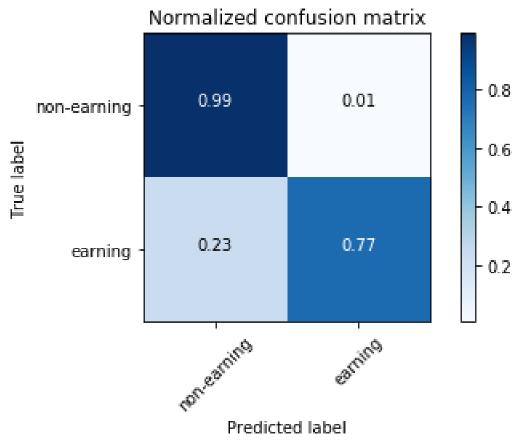




The average accuracy we obtained with the SGD classifier was 96.54%.

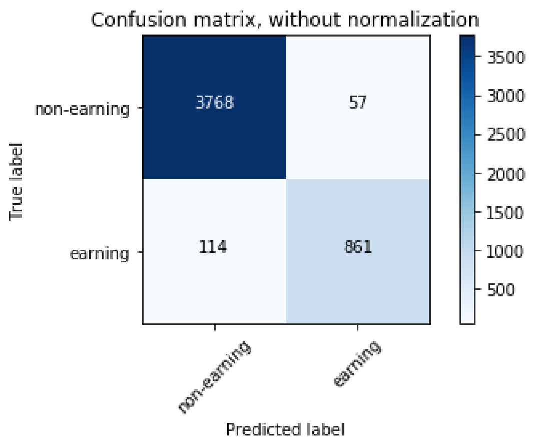
3.2.2 Multinomial Naive Bayes Classifier

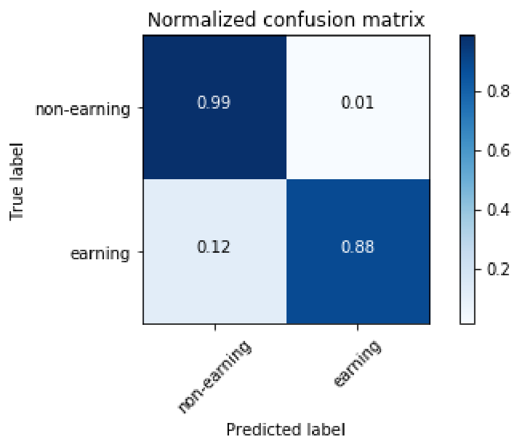




The average accuracy we obtained with the Multinomial Naive Bayes classifier was 94.56%.

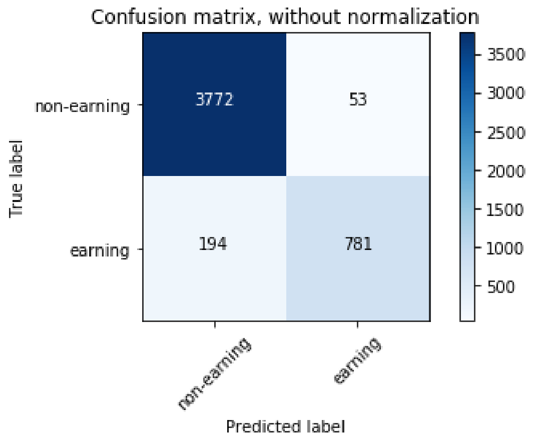
3.2.3 SVC linear kernel

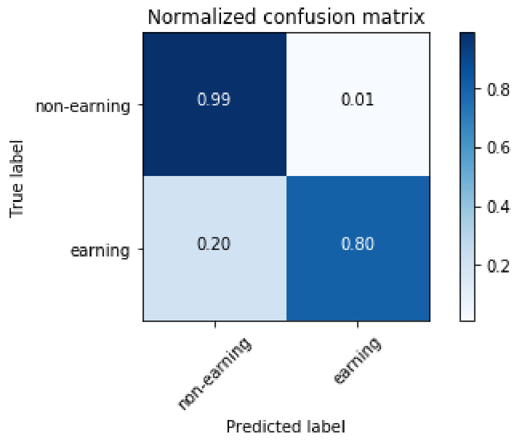




The average accuracy we obtained with SVC with linear kernel was 96.44%.

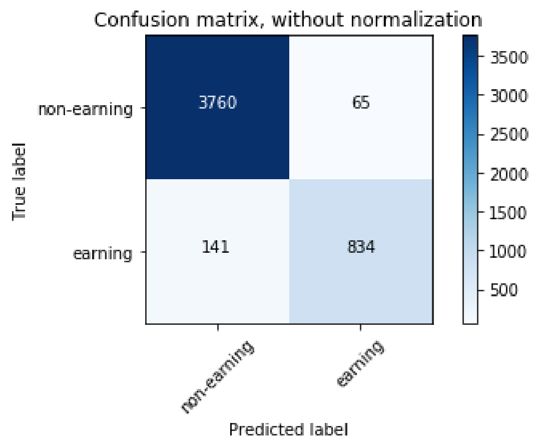
3.2.4 Random Forest

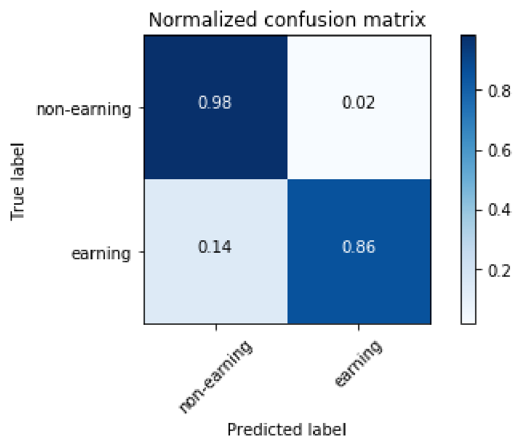




The average accuracy we obtained with random forest was 94.85%.

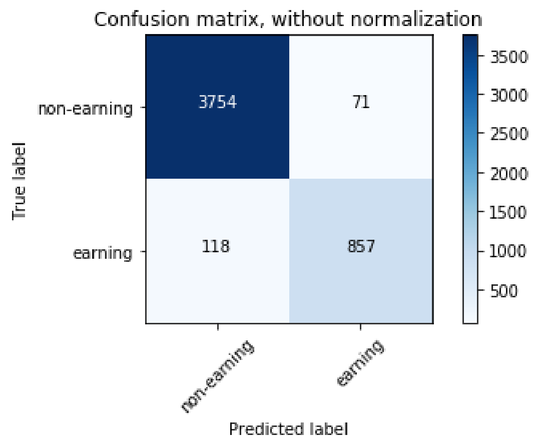
3.2.5 XGB Classifier

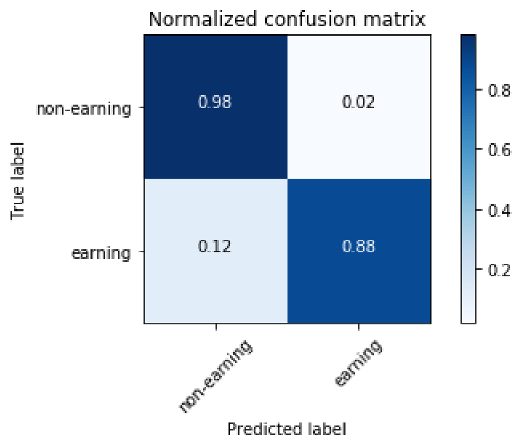




The average accuracy we obtained with the XGB classifier was 95.71%.

3.2.6 Multi Layer Perceptiron





The average accuracy we obtained with the MLP classifier was 96.06%.

3.2.7 Results

The best results were obtained with the SGD classifier. So it is with this classifier that we make predictions on the test set with unknown labels.

4 Improvements

The model should be improved with a more advanced preprocessing. We should deal with the abbrevations and synonyms. Furthermore, we think we could get better results if instead of a simple lemmatization, we realize an entity linking. For example, the tokens like *dollars*, *euros* or *zolty* should be replaced by `currency`.

Actually we also have a lot of potential problem inside the corpus vocabulary that could be format with an inverse index of feature. To detect potential feature. Like if we replace each city name by *city* maybe it will appear some unknown stuff

5 Conclusion

For the model selection we focus on a classifier that could well classified a lot of earning even if we got a lot of false positive. Because we want to know every articles of that subject to possibly deal with Stock Exchange for example, every information could be primordial. So we probably can extract a lot more stuff that could probably improve the preprocess part.

References

[1] Natural language toolkit. `https://www.nltk.org/`.

[2] Scikit-learn. `https://scikit-learn.org/`.