Abstract

Using scraped data from the year 2018, this paper tries to get insight in the 2018 Venezuelan humanitarian crisis using specific keywords and text mining approaches. Also, it evaluates the performance of different classification algorithms against the category it was first searched for.

Keywords: mining, data mining, automatic text classification, supervised learning

Venezuela, and a data-driven overlook of the 2018 Crisis

Piero Ulloa

February 22, 2019

1 Dataset

The dataset is composed of 4 features, the news' title, the article content, the date of the article, and the URI used to retrieve it. The data was collected using our own scraper, with simple rules, so the data is a bit on the noisy side, but otherwise varied, since it was collected from the news search results of Google.

It was purpose built for this research, so there's an almost equal amount of articles for each category

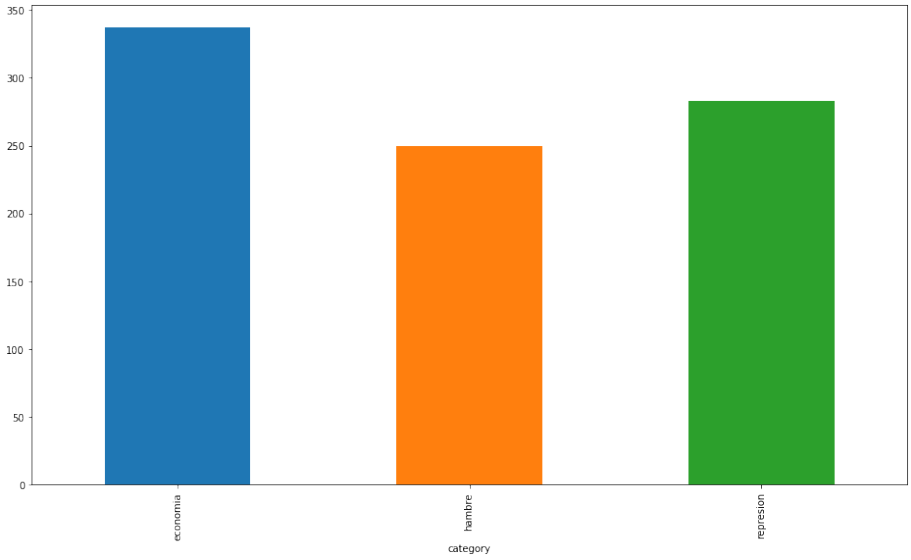

Figure 1: Plot of the absolute frequency of every category

1.1 Insights

After doing a simple correlation of the features in the td-idf matrix, we found some interesting bigrams and unigrams correlated with category words. Following next is a tree like structure, representing the most correlated bigrams and unigrams for every category.

- **"economia":**
  - Most correlated unigrams:
    - barril
    - pdvsa
    - petróleo
  - Most correlated bigrams:
    - fondo monetario
    - monetario internacional
    - banco central
- **"hambre"**
  - Most correlated unigrams:
    - alimentaria
    - fao
    - comida
  - Most correlated bigrams:
    - crisis alimentaria
    - alimentacion agricultura
    - seguridad alimentaria
- **"represion"**
  - Most correlated unigrams:
    - manifestacion
    - manifestante
    - protesta
  - Most correlated bigrams:
    - nacion bolivariana
    - cuerpo seguridad
    - guardia nacional

2 Methods

To build the dataset, we wrote a spider which scraped Google search results looking for news sites, and then it followed those links to the actual news, which then scraped using simple rules.

Then we processed the text of it into a TD-IDF matrix, and finally we used PCA to get a glimpse of how hard could a separation of these articles could be

In this plot we can appreciate that PCA projects all 10k features into a 2D space, and we immediately notice that separating all classes in the dataset wouldn't be an easy task. But perhaps we're looking at it wrong. Can you notice

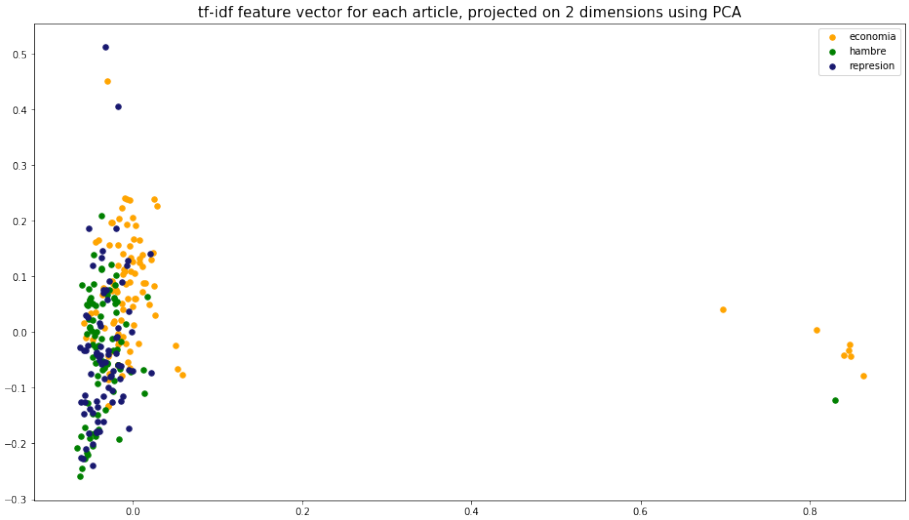

Figure 2: PCA applied to the TD-IDF matrix, grouped by category

that most of the data would group nicely in that corner, but it's overlapped with other datapoints. This means that more than 2D is needed to classify text.

3 Models

The models we used were:

- Linear Support Vector Classifier
- Logistic Regression
- Random Forest
- Multinomial Naïve Bayes
- Random Forest

Each of these classification algorithms were chosen because they are able to handle high dimensionality data, which is a must on text mining.

Each of these classifiers were given a TD-IDF matrix for every document in the dataset and the category of each news article as outcomes.

3.1 Model selection

First, the models were pitched against each other and had their accuracies measured under cross validation on the dataset as a way to know which ones could perform competently with the dataset.

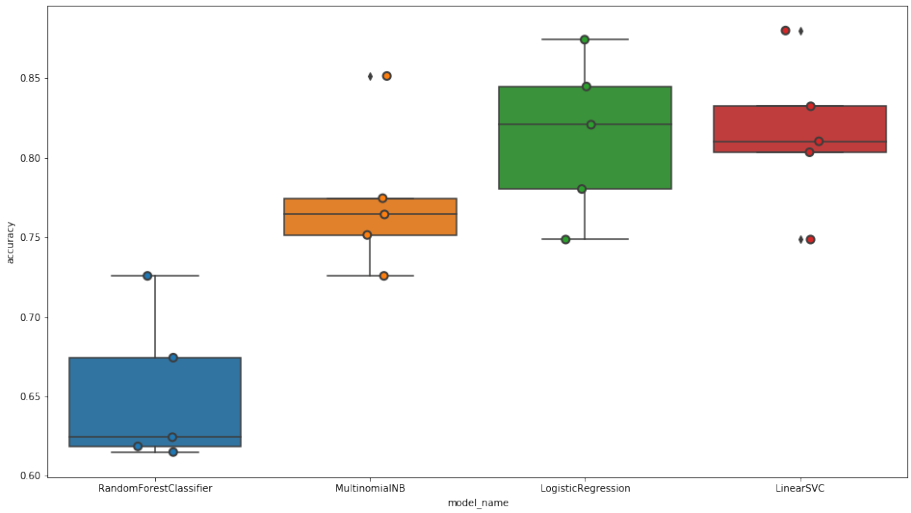

Figure 3: Box plot for every Cross Validation accuracy measure for each model

After that, we pitched the best performers, Logistic Regression, and Linear SVC, using different regularization parameters. This is a hyper-parameter that is easily adjustable in the tools used for this model selection which effects can be easily understood by looking at CV results.

We noticed stronger regularization (lower C value) provided lesser accuracy, and this is proof that the model hasn't overfitted. We're going to pick Logistic Regression, since it's the simplest model of both, and according to occam razor, it is preferable.

4 Results

The chosen algorithm was Logistic Regression, and the test results were favorable, with 81% accuracy on test dataset.

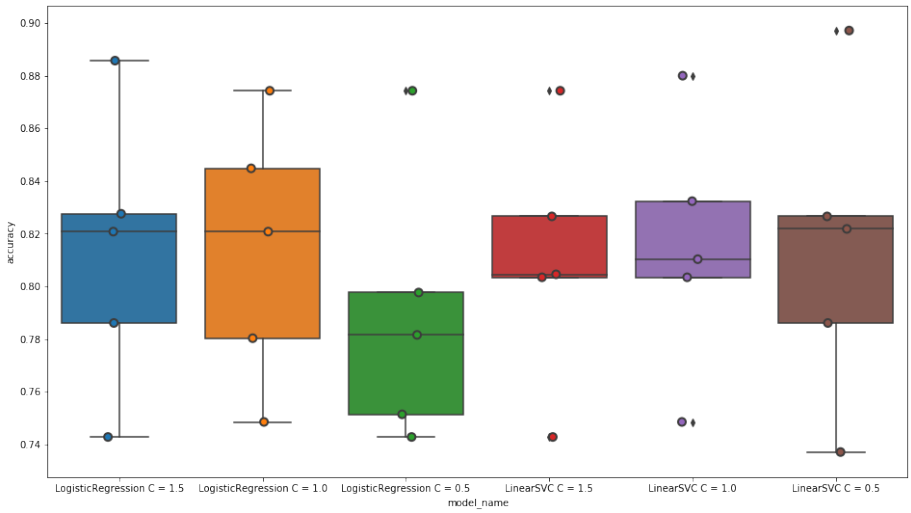

Figure 4: 5-fold CV for Logistic Regression and LinearSVC with different regularization parameters

4.1 Classification errors

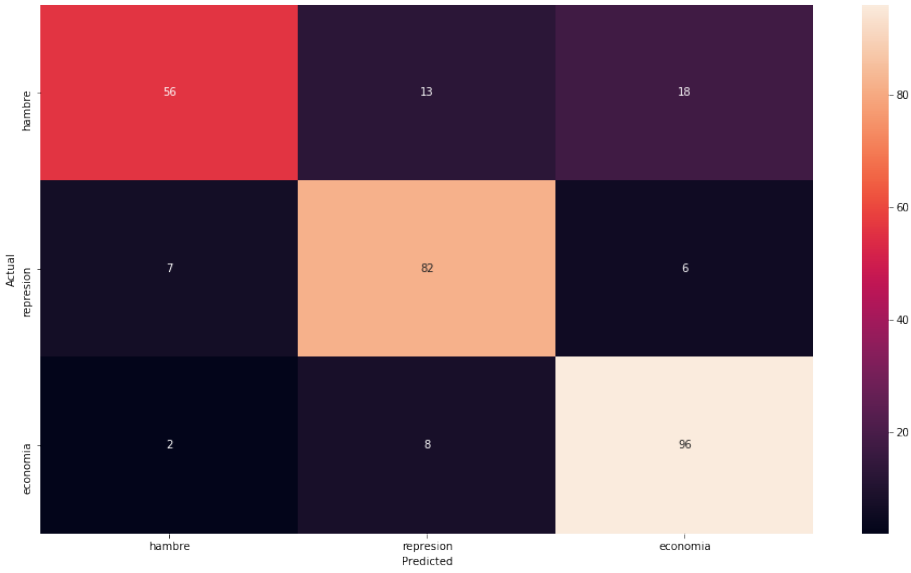

Figure 5: Confusion Matrix for linear regression

On figure 5 we see that this classifier performs great on the test set, but it has some problems classifying documents from the "hambre" category, with 31 of these samples were misclassified. After further inspection of the results we found that most of them were really noisy articles, and with a more thorough data cleansing, the model could eventually predict their class properly.

We also put the model in a validation test, with the validation data collected from another source, and transformed it using the TF-IDF vectorizer used while training. The performance was noticeably hurt.

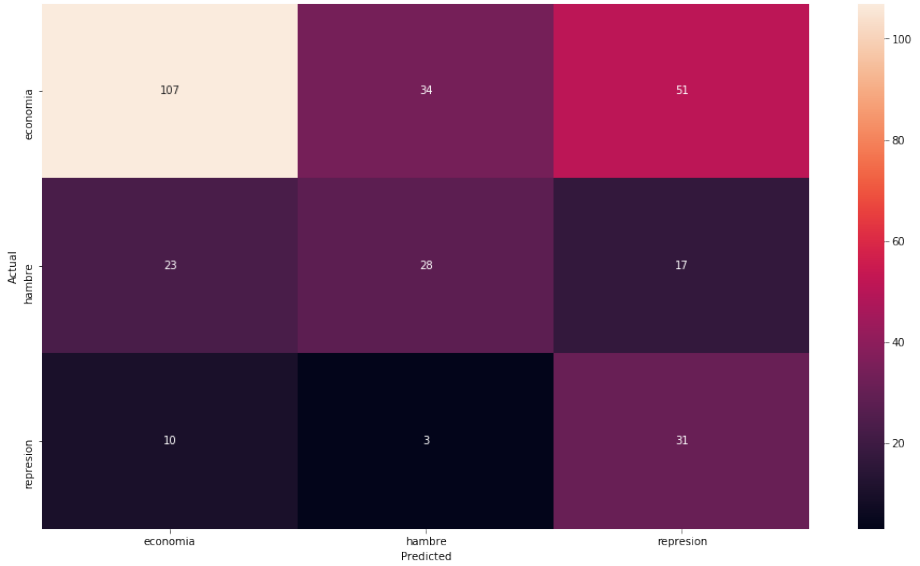

Figure 6: Confusion matrix for validation dataset

The figure 6 shows that in general it performs worse in every category, indicating that the model might have learned some characteristics in the training dataset that weren't present in the test dataset (as they were collected differently)

5 Discussion

After looking at the accuracy results, we want to discuss about what it actually means to be wrong. In a posterior examination, we found an article classified as "represion", yet the classifier predicted it was economy. After a careful examination of the news article[1], it was found that this article actual category was "represion",even though the article wasn't about it, but instead it was about China's economic interests and the political instability of Maduro's term and

[1] https://www.bbc.com/mundo/noticias-america-latina-47170209

what it meant for Chinese investors, a topic which clearly can't be classified in just one category.

In hindsight, RandomForest wouldn't have been a good performer, since automatic document classification is highly dimensional, and we limited every tree to have little depth. On the opposite side, letting the depth increase would have made those decision trees overfit the data.

As a final recommendation, we think the models are promising, but if we could feed them with cleaner data, it might generalize better.

6 Acknowledgements

We wish to acknowledge with Carlos Orellana, as the source of our validation data.