

**Text Classifier**

Interview Challenge

Candidate : Hajer Mahjoub

Data Science Engineer Student

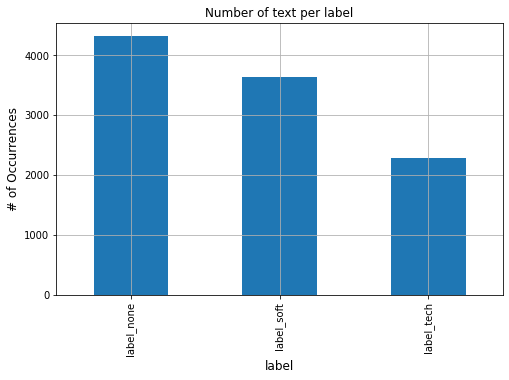
At ESPRIT Tunis

1. **Exploratory data analysis :**

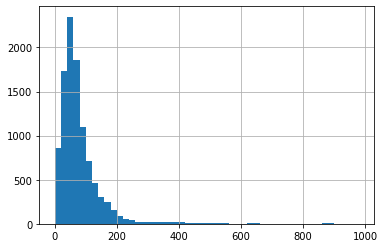
* Libraries: seaborn ,json ,pandas ,matplotlib
* Notebook : TalentBait\_EDA.ipynb
* The dataset source is a JSON file composed of 10 000 sentence labeled
* Labels :

“soft” ,”tech”,”none”

* Number of text per label :

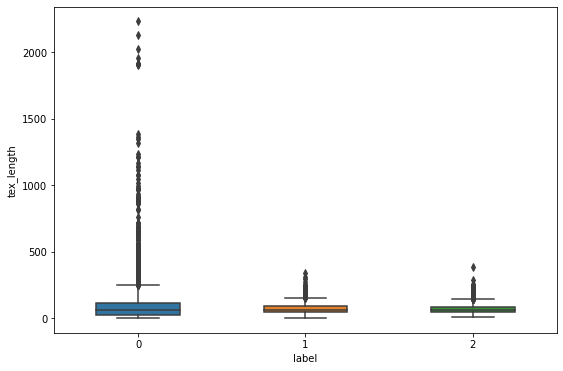


* The distribution of the number of words in text column :

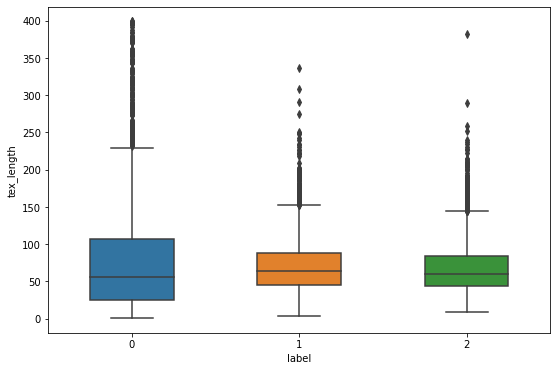


The most of the text are within 400 characters with some outliers up to 900 characters long

* Boxplot :

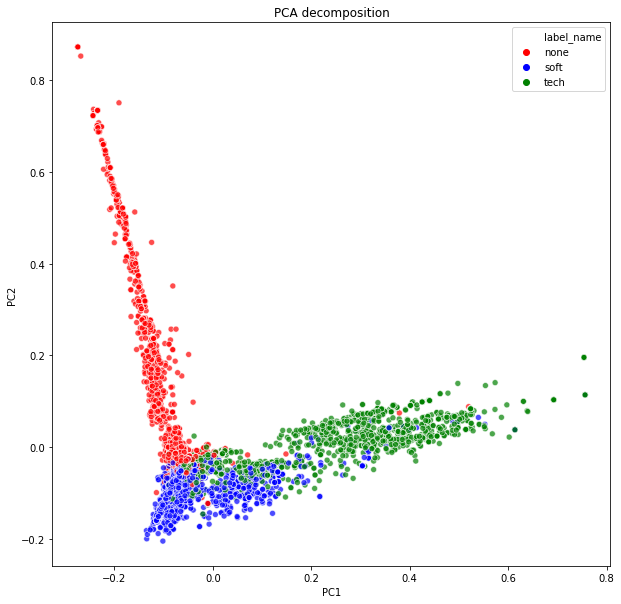


So we delete the long text

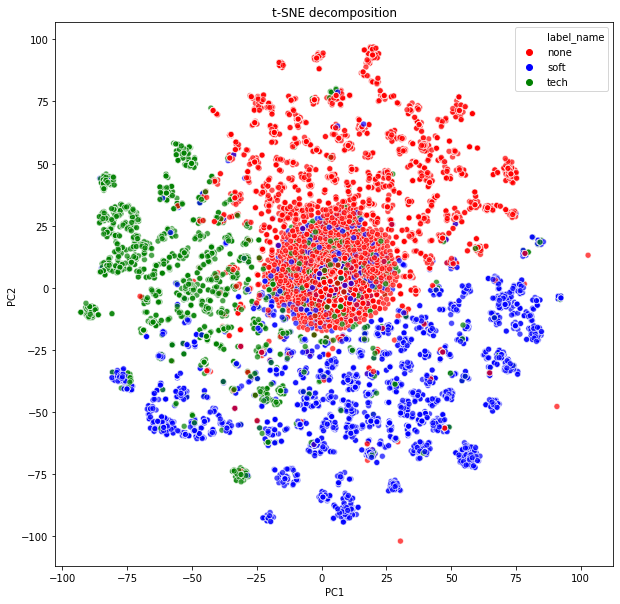


* We can see that, although the length distribution of our data is close similar to each category even that once we are creating the features with TF-IDF scoring, we will normalize the features to keep this distribution
* **Dimensionality Reduction :**
* Notebook : TalentBait\_ DimensionalityReduction.ipynb
* We'll perform a dimensionality reduction technique to plot the observations in 2 dimensions

1. **Principal Component Analysis :**

****

1. **t-distributed Stochastic Neighbour Embedding:**

****

We can see that the PCA decomposition works much better on our data.

**2. Feature Engineering :**

* Libraries : NLTK , Scikit-Learn , Regular Expression
* Notebook : TalentBait\_EDA.ipynb
* Process of feature engineering :
* Start by cleaning the text from (signs , special characters )
* Lemmatization & Stemming
* Label Encoding
* Split the train and the test data
* Text representation :

Choosing TF-IDF Vectors feature ([short for term frequency–inverse document frequency](https://en.wikipedia.org/wiki/Tf%E2%80%93idf))

* Chi squared test : see unigrams and bigrams most correlated with each label :
* 'none' label:
* Most correlated unigrams:

gute,standort,deutschland,gmbh,kenntnisse

* Most correlated bigrams:

ms office,gute kenntnisse

* 'soft' label:
* Most correlated unigrams:

umgang,sap,office, ms,kenntnisse

* Most correlated bigrams:

gute kenntnisse,ms office

* 'tech' label:
* Most correlated unigrams: flexibilität,hohe,kenntnisse,teamfähigkeit,arbeitsweise
* Most correlated bigrams:

hohes maß,wort schrift

* save the files

**3. Model (Train & Test) :**

* Notebook : TalentBait\_Model2.ipynb
* we'll try several machine learning classification models in order to find which one performs best on our data.
* We will try with the following models:
* Random Forest
* Multinomial Naïve Bayes
* Multinomial Logistic Regression
* Gradient Boosting
* The methodology :

1 choose hyperparameters

2 Search Cross Validation process

3 Grid Search Cross Validation

4 Training & Test

5 Calculate accuracy

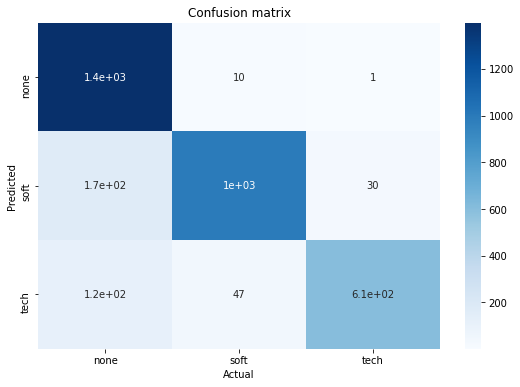
Note the prediction classify the text and in addition give the the conditional probability of belonging to every label

6 Reporting the performance of each model and plot the confusion matrix to summarize

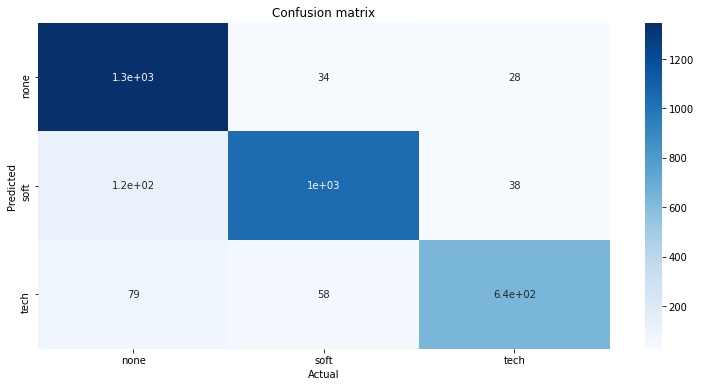
the performance

**. Confusion Matrix of each model :**

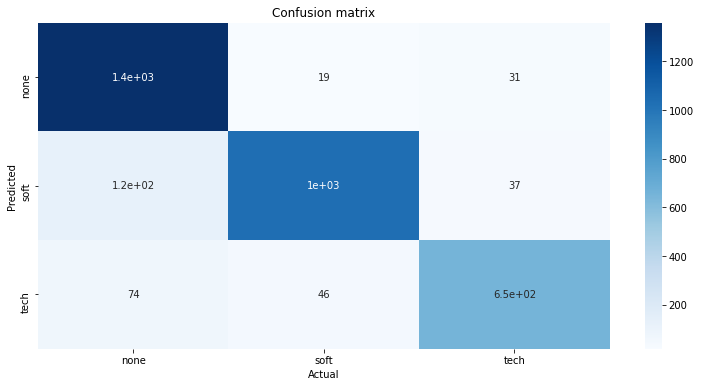
* **Random Forest**

****

* **Multinomial Naïve Bayes**

****

* **Multinomial Logistic Regression**

****

* The Summary table of performance metrics generated in every model :

|  |  |  |
| --- | --- | --- |
| **Model** | **Training Set Accuracy** | **Test Set Accuracy** |
| Multinomial Logistic Regression | 0.923536 | 0.902158 |
| Multinomial Naïve Bayes | 0.917856 | 0.893881 |
| Random Forest | 0.918584 | 0.887674 |

* I did choose the Multinomial Logistic Regression since it has the highest Test Set Accuracy.

**4. Prediction and application :**

* For test the chosen model and classify and Input text

Two solution were provided :

1. Notebook : Prediction.ipynb

You can run the cells and fill the text variable with the Input to classify

1. Notebook : theApplication.ipynb

Using Dash an Open Source Library i created simple web app with simple interface to inter text and get prediction to run the app currently just run the cells on the notebook

Note : the app can be optimise and deployed

We can use other methods for representation of the text

There is second methodology for the classification of the text i tried under the Notebook “TaletBait\_Modeling\_2.ipynb”