TOXIC CONVERSATIONS CLASSIFICATION

Abstract

This project proposes to implement, as part of a Kaggle competition, a metric to evaluate a machine learning algorithm operating on a population composed of individuals classified in subgroups. These subgroups mark a social or ethnic identity of the individuals making up the population.

Details of projects : https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

- An exploratory analysis of the corpus of texts has been conducted allowing to argue over the choice of learning model.

- Data-preparation process is described and leads to a digital representation of the input data to feed machine learning algorithm.

- The study of machine learning hyper-parameters is conducted and parameters used for building learning model are selected.

- Selected model is trained and results in term of binary classifications are exposed along with bias computation in compliance with formula described in https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/overview/evaluation

Project artefacts

- Project artefacts are located under http://bit.ly/FBangui_Datasciences_KaggleJigsaw

- These slides present the overall approach of the study : https://github.com/dataforcast/OC_Datascientist/blob/master/P8/report/Slides_P9_V1.pdf

Software architecture

Exploratory analyis

Features description

Content & target

target	**0**
comment_text	**0**

Toxicity indicators

severe_toxicity	0
obscene	0
identity_attack	0
insult	0
threat	0
funny	0
wow	0
sad	0
likes	0
disagree	0
sexual_explicit	0

Misc informations

id	0
created_date	0
publication_id	0
article_id	0
rating	0
identity_annotator_count	0
toxicity_annotator_count	0
parent_id	778646

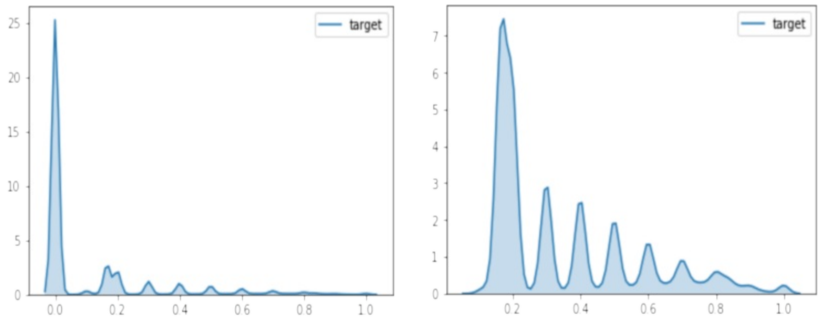
Identities

asian	1399744
atheist	1399744
bisexual	1399744
black	1399744
buddhist	1399744
christian	1399744
female	1399744
heterosexual	1399744
hindu	1399744
homosexual_gay_or_lesbian	1399744
intellectual_or_learning_disability	1399744
jewish	1399744
latino	1399744
male	1399744
muslim	1399744
other_disability	1399744
other_gender	1399744
other_race_or_ethnicity	1399744
other_religion	1399744
other_sexual_orientation	1399744
physical_disability	1399744
psychiatric_or_mental_illness	1399744
transgender	1399744
white	1399744

The dataset is composed of 45 columns described as following:

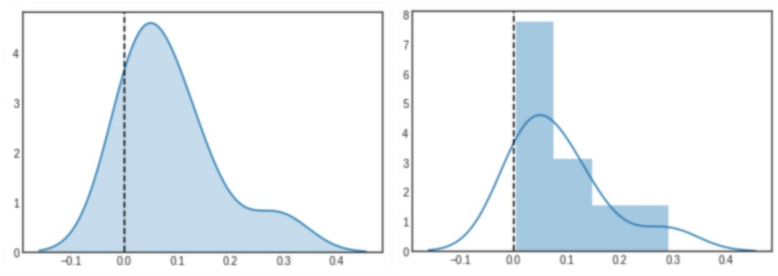
- Toxicity indicators : these features indicates the type of toxicity
- Identities : these features indicates presence into comments of words or expressions related to an identity. Bias of predictive model will be evaluate against some of these identities.
- Misc informations : these features provide additional informations over comments.
- Content & target : Content are used to feed machine learning model (after a data-preparation and a digitalization process) and target are used as labels to train models.

Target distribution



Toxicity distribution

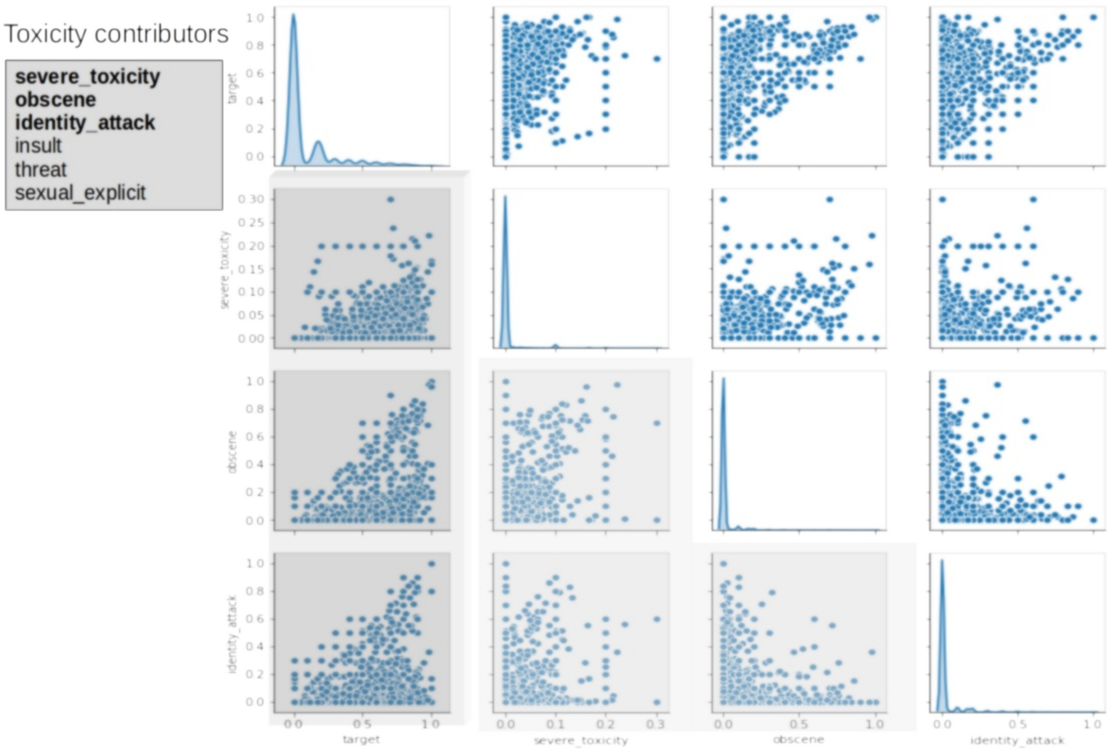
Toxicity > 0.1 : 30 %



Toxicity values (named target) are ranged from 0.0 (safe comment) to 1.0 (more toxic comment).
Toxicity disrtribution shows unbalanced classes. Around 30% of comments have a toxicity value greater then 0.0.

Note also that continuous values of toxicity are difficult to interpret and perceive distinctly for a human. E.g it is not clear to perceive how far a comment with toxicity value of 0.38 is fare from another comment with toxicity value fixed to 0.35.

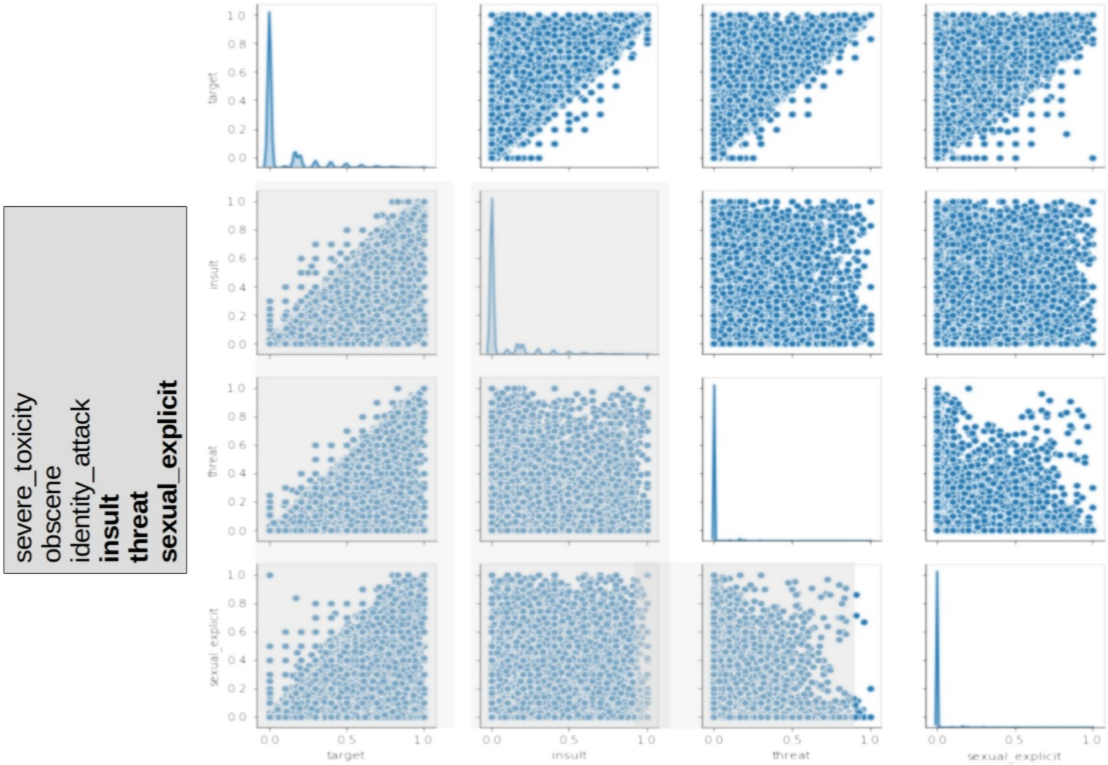
Toxicity contributors analysis



The group of features, named **Toxicity distribution** are analysed against target values in order to highlight associations between dependant variables and features.

This multivariate analysis shows that **identity_attack, obscene** and **severe_toxicity** in extruded area show a linear border of the shape of distributed points along with target. The border of the shape *increases linearly* with toxicity range of values. The fact that a dense area of points appears under the diagonal border let think that contrubution to toxicity may be due to a combinaison ot this three features.

We note also that in non extruded area, multivariate analysis between features **identity_attack** and **obscene** expose the same kind of linearly border shape that *linearly decreases*



On the display above, the multivariate analysis between features **insult, threat, sexual_explicit** and **target,** over the extruded area, shows a more accentuated and dense distribution of points below the diagonal border that *linearly increases* with target values.

ANOVA over identities

QUESTION: is it possible to infer a relation between toxicity and identities **?**

Identities are used in order to evaluate bias. Idea is to evaluate the contribution of identities over the toxicity variance

For doing so, ANOVA is proceeded over each one of the groups formed with identities.

Identities values are continuous values ranged from `0.0` to `1.0`. These values are reworked and are labelized with values `0,1,... 10`. Such transformation allows to consider identities as formed with qualitative groups. These labelized ranges of values may be interpreted as levels of the considered identity.

The labeling scheme is as following :

Range of group values	Label
0	0
]0.0 , 0.1]	1
]0.1 , 0.2]	2
]0.2 , 0.3]	3
]0.3 , 0.4]	4
]0.4 , 0.5]	5
]0.5 , 0.6]	6
]0.6 , 0.7]	7
]0.7 , 0.8]	8
]0.8 , 0.9]	9
]0.9 , 1.0]	10

E.g. for female identity, such transformation leads to consider **11** groups, ranged from **0** to **10**, with statistics values describe below for female identity :

Female	N	Mean	SD	SE	95% Conf.	Interval
level_0	2782	0.132164	0.216442	0.004104	0.124120	0.140209
level_6	2782	0.160483	0.225222	0.004270	0.152112	0.168854
level_8	2782	0.166551	0.218262	0.004138	0.158438	0.174663
level_7	2782	0.166963	0.220442	0.004179	0.158770	0.175156
level_2	2782	0.167104	0.239145	0.004534	0.158216	0.175993
level_10	2782	0.169066	0.223008	0.004228	0.160778	0.177355
level_5	2782	0.169404	0.230365	0.004368	0.160842	0.177966
level_3	2782	0.170108	0.236792	0.004489	0.161308	0.178909
level_9	2782	0.178096	0.227038	0.004304	0.169658	0.186535
level_4	2782	0.179424	0.243967	0.004625	0.170356	0.188491
level_1	2782	0.186221	0.232819	0.004414	0.177568	0.194874

Means values are the means of toxicity computed for each group.

Question : are differences between these means significant **?** Is there an inferable relation between identities labels, ranged from **0** to **10,** and toxicity level of a comment **?**

For any of the groups in the list above, **Levene** test is conducted. This test allows to validate **homodestaticity** hypothesis : variance in each group has equal value. This is the **H0 hypothesis stating over variance equality** between the groups of an identity. If Levene test is significant, considering p-value, then null hypothesis will be accepted and homodestaticity hypothesis will be valid.

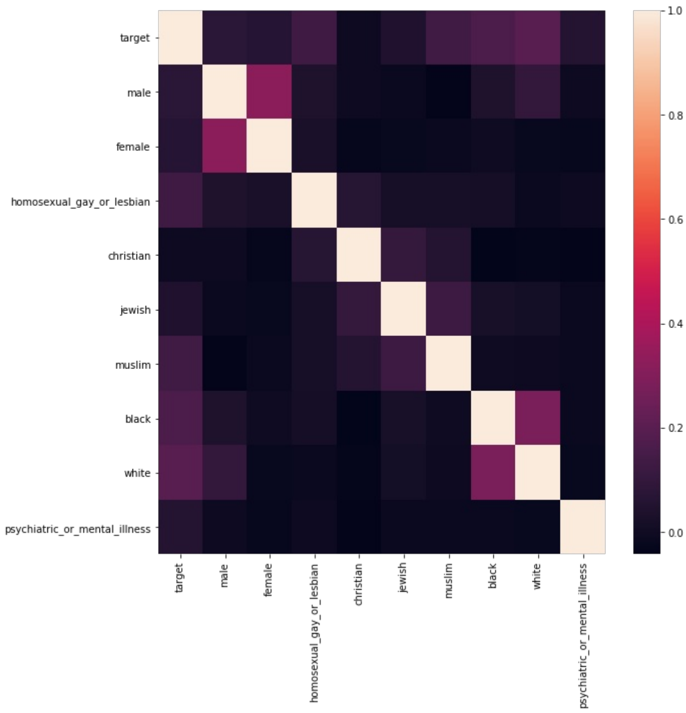
Result are produced on the figure below:

	Sampling per group	Levene		ANOVA		
Identities		F-stat	p-value	R^2	F-stat	p-value
male	3473	23	0.0	0.00	18.53	0.00
female	2782	10.5	0.0	0.00	10.20	0.00
homosexual_gay_or_lesbian	708	13	0.0	0.05	42.54	0.00
christian	2218	36	0.0	0.01	36.02	0.00
jewish	**214**	**1.48**	**0.14**	**0.01**	**2.805**	**2.e-3**
muslim	1045	19.4	0.0	0.03	33.82	0.00
black	780	13	0.0	0.06	52.86	0.00
white	835	14.5	0.0	0.06	56.35	0.00
psychiatric_or_mental_illness	576	14.5	0.0	0.02	10.76	0.00

Levene test seems to be not significant for all identities except jewish group. Nevertheless, for making F-stat robust considering **normality and homodestaticity hypothesis**, all groups inside each one of the identities have been sampled with the name number of observations. Then, for each identity, considering F-stat magnetude (that measures a ratio of the variability of the mean inside groups and the variability of the mean between groups) and associated p-value << **5.e-2** (that indicates how significant is the result of F-stat), test over differences of means between groups are **regarded as significant.**

As a conclusion of ANOVA :

- Considering `F-stat` values for all identities, there is a significant difference between means of groups, groups organized as levels of an identity.
- Considering R^2 values for all identities, variance is very weakly explained by any idntity, leading to the conclusion that none of the identities have effect over comments toxicity. Correlation matrix here-under partially illustrates this claims. This correlation matrix has been built over the whole observations. We note a symetrix aspects of correlations for couples of features `(male, female)` a well as `(black, white)`. These features show a **middle size effect** for explaing target variance.



In addition, diagram belows exposes the distribution of a sample of `5000` points for identities that are taken into account for bias evaluation along with toxicity. There is no intuitive evidence of informational shape as pointed for toxicity contributors features on previous section.

For jewish identity, having the lowest number of observations among all other identities, the F-statistic and related p-value, respectively

2.8 and 2.e-3 shows that there is a significant difference between means of groups inside jewish identity.

Model built based on the list of identities, allows to conclude that each identity has a significant effect over toxicity level.

Table below shows, for jewish identity, means and standard deviation of any of the groups built on labeled indices ranges of values. For the need of test reliability, all groups have been sampled with the same number of bservations, 214.

Jewish identity						
	N	**Mean**	**SD**	**SE**	**95%**	**Conf. Interval**
level_0	214	0.137278	0.225208	0.015395	0.107034	0.167523
level_1	214	0.146550	0.204520	0.013981	0.119083	0.174016
level_5	214	0.165121	0.207500	0.014184	0.137254	0.192987
level_2	214	0.167728	0.213974	0.014627	0.138992	0.196464
level_3	214	0.168272	0.203585	0.013917	0.140931	0.195613
level_4	214	0.168771	0.221208	0.015121	0.139063	0.198479
level_7	214	0.177465	0.217144	0.014844	0.148303	0.206626
level_6	214	0.180769	0.199595	0.013644	0.153964	0.207574
level_10	214	0.199158	0.224799	0.015367	0.168968	0.229348
level_8	214	0.209509	0.229285	0.015674	0.178716	0.240301
level_9	214	0.218742	0.237592	0.016241	0.186834	0.250649

Means show differences in between labelized groups. Are these differences significant? If yes, then it will be allowed to conclude that such groups, based on jewish identity, do explain variancy of toxicity. In that case, it will be stand that a relation between toxiciy level and jewish identity exists.

Table below shows the result of ANOVA over jewish identity.

	Coef	std err	t	P>|t|	[0.025	0.975]
Intercept	**0.1373**	**0.015**	**9.251**	**0.000**	**0.108**	**0.166**
C(jewish)[T.level_1]	0.0093	0.021	0.442	0.659	-0.032	0.050
C(jewish)[T.level_10]	**0.0619**	**0.021**	**2.949**	**0.003**	**0.021**	**0.103**
C(jewish)[T.level_2]	0.0304	0.021	1.451	0.147	-0.011	0.072
C(jewish)[T.level_3]	0.0310	0.021	1.477	0.140	-0.010	0.072
C(jewish)[T.level_4]	0.0315	0.021	1.501	0.134	-0.010	0.073
C(jewish)[T.level_5]	0.0278	0.021	1.327	0.185	-0.013	0.069
C(jewish)[T.level_6]	**0.0435**	**0.021**	**2.072**	**0.038**	**0.002**	**0.085**
C(jewish)[T.level_7]	**0.0402**	**0.021**	**1.915**	**0.056**	**-0.001**	**0.081**
C(jewish)[T.level_8]	0.0722	0.021	3.442	0.001	0.031	0.113
C(jewish)[T.level_9]	0.0815	0.021	3.882	0.000	0.040	0.123

ANOVA has been conducted over 214 observation and shows :

- R^2 value, `1.e-2` is a weak value. Correlation between toxiciy and identity is weak. There is no **linear relation** between jewish identity and toxicity. Correlation matrix here-under supports this conclusion :
- From the column `P>|t|` the groups labelized `6, 7, 10` have significant difference with intercept, that is group level `0`. **Post-hoc** test will allows to compare groups level `6, 7` and `10` in between each-other.

Conclusions over features analysis

Features have been splitted into two categories : contributors to toxicity of coments and identities. It has been shown that contributors to toxicity has a strong effect over the toxicity magnetude, whereas identities have a weak effect over toxicity. Unintended bias in machine learning classification comes with the fact that despite that last point, comments tend to be classified as toxic when they embedd some identities terms or expressions.

Absence of identity effects over toxicity also let think that the process involved in toxicity and identity metrics can be safely taken into account. This lead to consider as reliable comments labelization used in a supervized machine learning model.

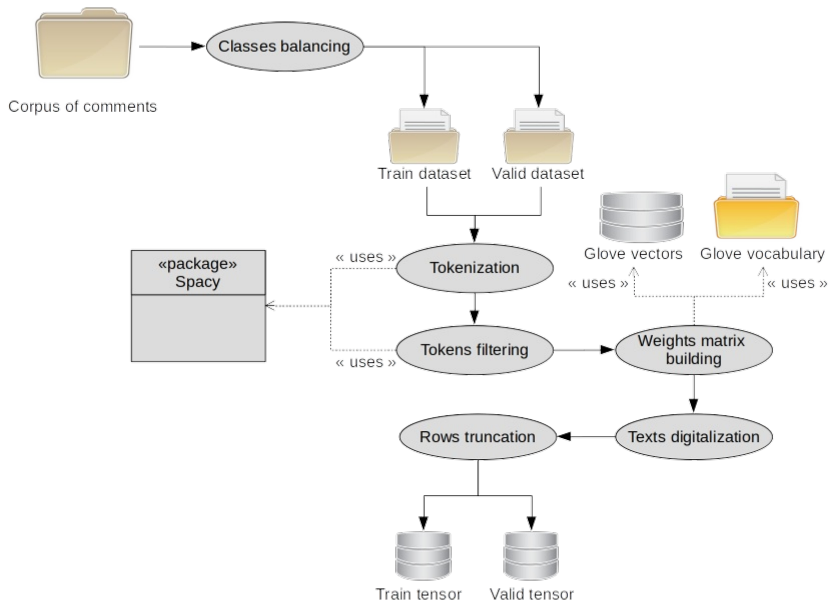
Same remark also stands for toxicity contributors. Their co-variancy behaviour along with toxicity is compliant with the intuitive sens, stating that a comment that embbeds insult terms or expressions is welling to render it toxic.

Proper algorithms should be able to take into account the presence of toxicity contributors along with identities in comments in order to proceed to classification.

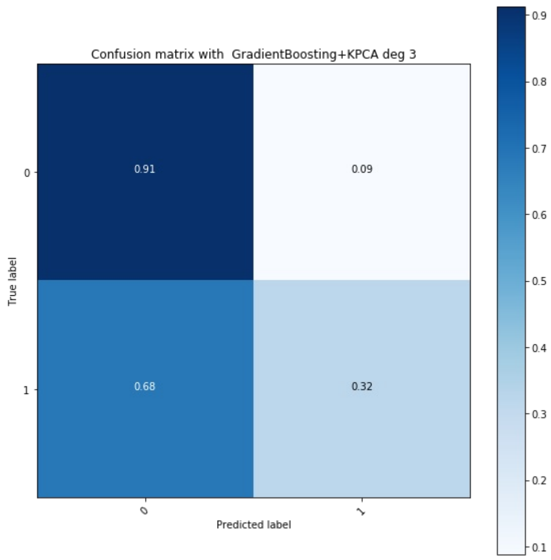
Also, post-hoc tests shows no evidence of an existance of treshold effect identifying differences between means of different groups inside an identity.

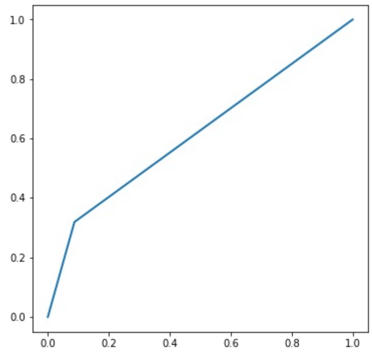
Data preparation

This process allows to transforms a corpus of comments into a digitalized representation, allowing to feed an algorithm. The global process is described below :



- Spacy package is used for NLP processing. Using it, sequences of words are preserved using.
- Due to toxicity distribution study in previous sections, balancing dataset is required in order to ensure toxic and non toxic comments to be processed "equaly". Otherwise, contributions to cost function will be mainly due to non toxic comments that will lead to a biased prediction model. The picture above do represents a model obtained with Gradient boosting algorithm without dataset balancing. All non toxic comments are very well classified while predictions for toxic comments are not better then random prediction leading to weak performances as shown over the RAUC curve.



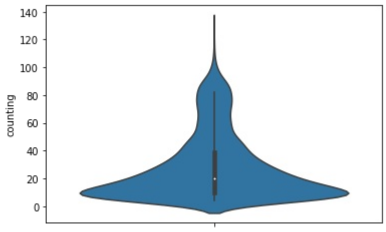


- As the result of the digitalization process, a tensor with 3 dimensions is produced, for train and validation dataset. It is represented as the figure below :

		Dim 0	...	DIM D
Text t	Token 0	$\text{coeff}_{t,0,0}$		$\text{coeff}_{t,0,D}$
	Token 1	$\text{coeff}_{t,1,0}$		$\text{coeff}_{t,1,D}$
	...			
	Token N	$\text{coeff}_{t+1,N,0}$		$\text{coeff}_{t+1,N,D}$
Text t+1	Token 0	$\text{coeff}_{t+1,0,0}$		$\text{coeff}_{t+1,0,D}$
	Token 1	$\text{coeff}_{t+1,1,0}$		$\text{coeff}_{t+1,1,D}$
	...			
	Token N	$\text{coeff}_{t+1,N,0}$		$\text{coeff}_{t+1,N,D}$

Dimension 1 : Number of texts | Dimension 2 : Max text length | Dimension 3 : embedings

- Texts truncation : all tokenized texts should have the same number of tokens. This allows to feed NN algorithm with **same embedding layer size.** This operation takes place after the filtering process, for keeping the model with usefull informations. The max length size for the number of tokens has been fixed to 100. This value comes with the text word length distribution described on diagram below :



Trunc of texts after digitalization allows to drive this process based on an objective criteria, such as magnetude of the vectors. Vectors with the smallest magnetude will be pushed out of digitalization representation. While doing so, relevant information that will contribute to decrease of loss function will be kept.

- Texts padding : digitalized texts with number of tokens less then max length will be padded with zero vector.

Word embeddings

Glove allows to use different kind of model language for Naturel Language Processing. The one used here is `en_core_web_lg`, in which, each word in vocabulary is represented as a vector in a `300` dimensions space. Words vectors have been built using web texts and comments issued from variou social networks.

A dictionary structured as `{word:glove_coefficient}` is built from Glove source.

Once built, dictionary allows to build a vector for every word in vocabulary issued from tokenizer.

Endly, weights matrix is built from vocabulary issued from tokenizer.

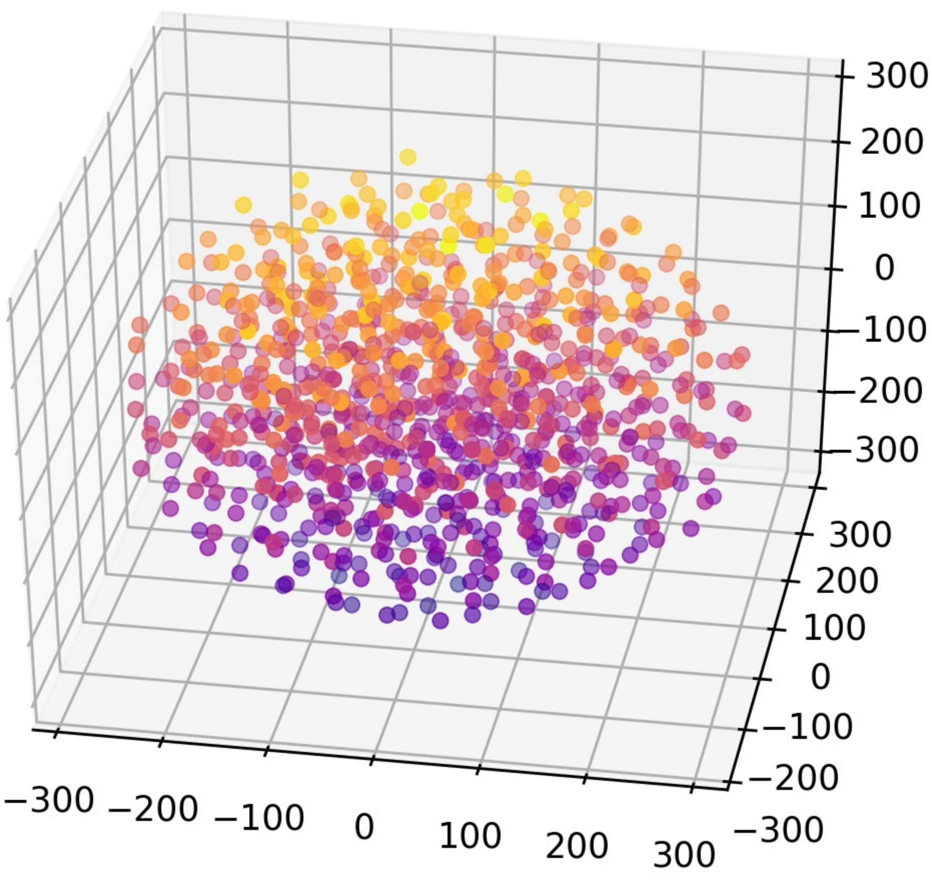
Such process is summarized with sequences here-under :

- `dict_glove_word_coeff <-- processing Glove file name`
- `vocabulary_word, index <-- tokenizer`
- `weight_vector = dict_glove_word_coeff[vocabulary_word]`
- `weight_matrix[index] = weight_vector`

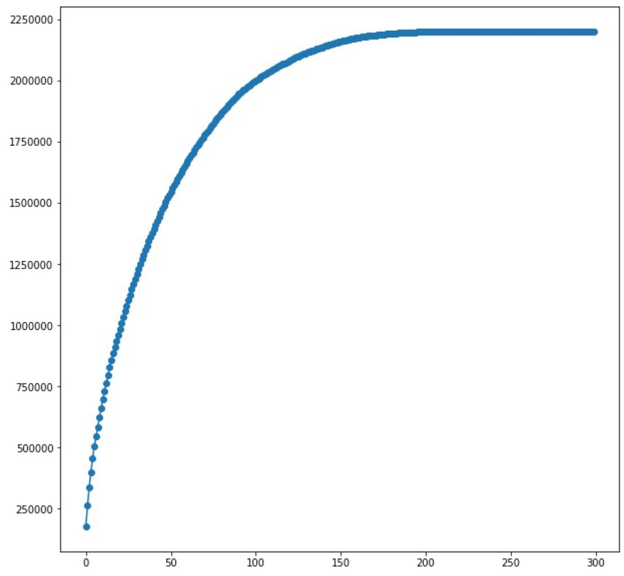
Features correlations

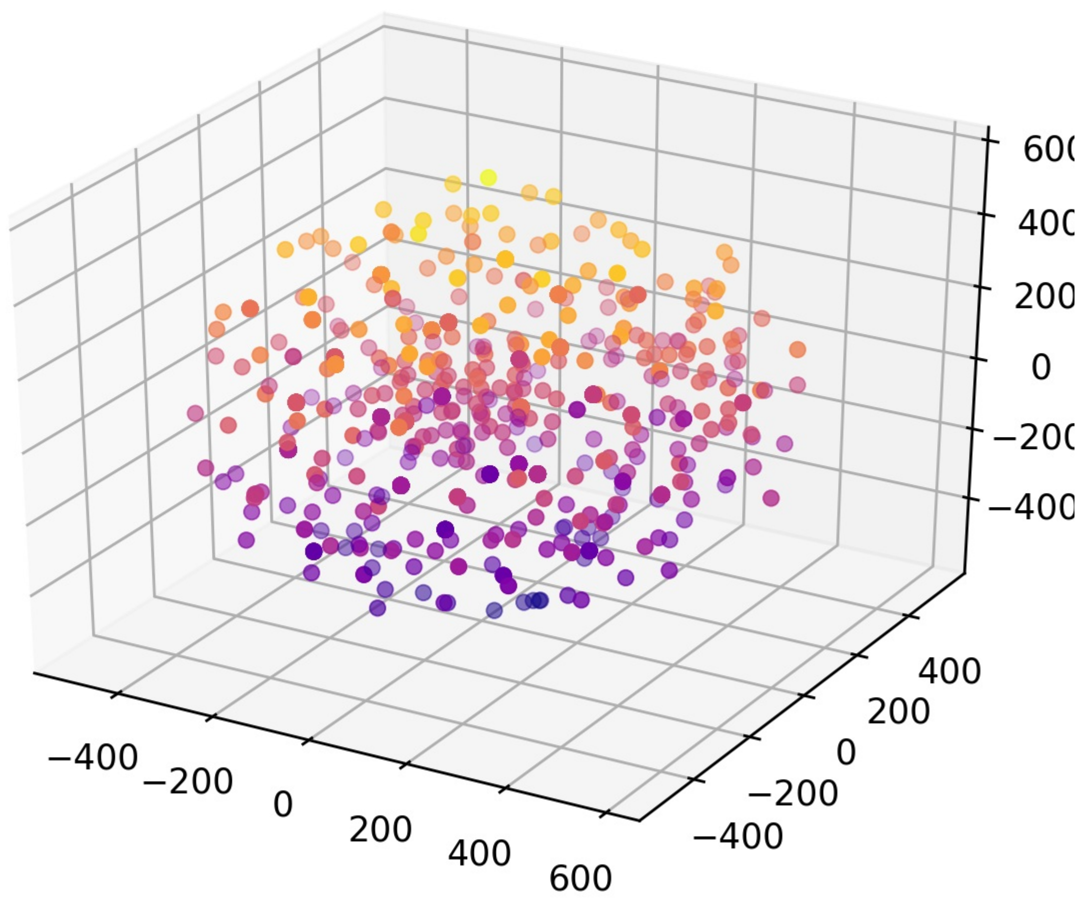
Statistic analysis as shown that some features expose linear correlations with toxicity values.

Figure below shows 1000 digitalized comments with process described in previous section. t-SNE operator is applied over 3 dimensions. There is no evidence that an hyperplan exists that is able separate groups of points.



On figure below, a kernel PCA operator has been applied with kernel of order 5. The kernel trick allows to process a non linear model as a linear model in an Hilbert space of larger size then the original space.





The 3D reduction of model issued from kernel trick operator shows group of separated structures. The complexity of such NLP model forbids the use of linear based models.

CNN classifier

From statistic analysis conducted in previous sections, it is shown that the presence of some type of words or expressions in a comment is correlated to the toxicity level. The problem can be formulated as the way to identity in a comment, what has been named toxicity contributors and identities and relate them to a level of toxicity. Due to that, the expected prediction model should be able to learn such relations connecting identified structures with toxicity level.

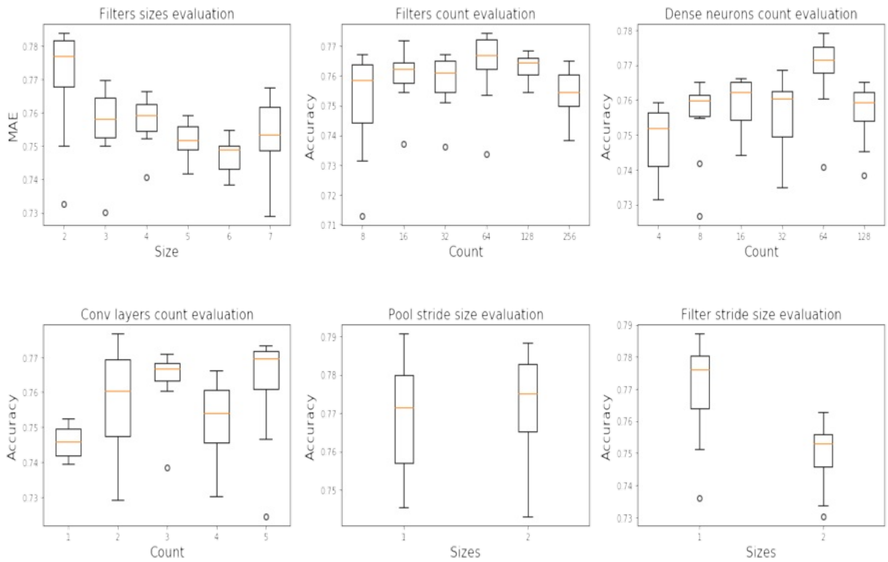
The ground problem is focused on textual structures identification. CNN architectures are famous and relevant to accomplish a such task.

In addition, a bias metric defined in https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/overview/evaluation will be applied over the model in order to evaluate how fare the model is able to distinguish toxic comments from non toxic comments that embbeds identities terms or expressions.

Rather then predict toxicity level thanks to a regression algorithm, a classifier is used to complie with the model bias evaluation.

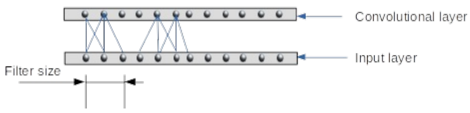
CNN hyper-parameters selection

CNN hyper-parameters selection for convolutional layers



These parameters are related to CNN structure :

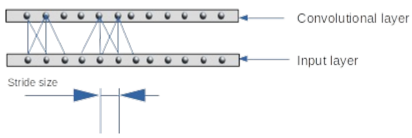
- The size of the convolution filters. When processing text, this parameter amounts to setting n-grams for the neighborhood of words. Such a structure in word processing is related to the semantics of words, indicating that words of the same neighborhood have the same meanings.



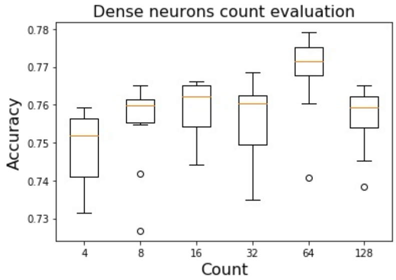
- The number of convolutional filters, leading to features maps. This will give the model the potential for identitying all kinds of textual structures in the corpus of comments. A single feature map identifies the same structures that may be repeated in a comment.

- The number of convolutional layers.

- The filter stride size. This parameter fixes the window stride for N-GRAM structures detection.



CNN hyper-parameters selection for dense layers



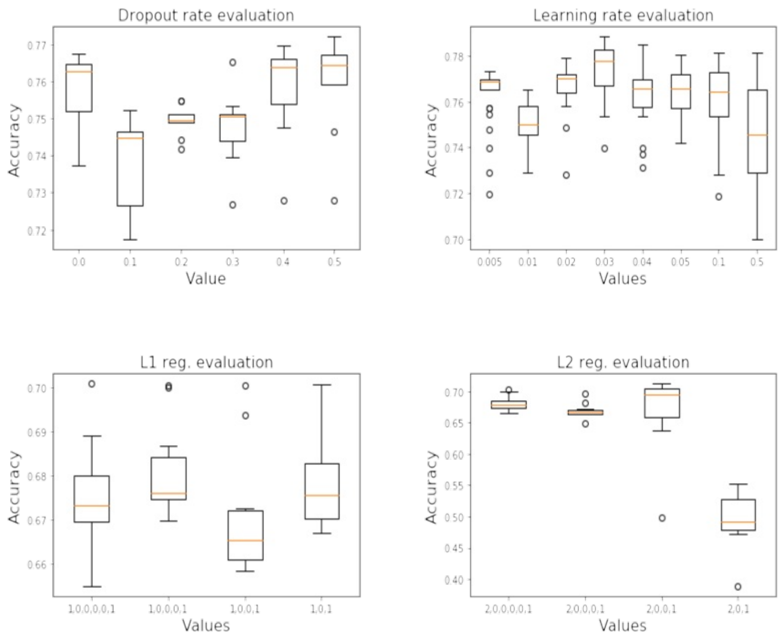
These parameters are related to dense layers structure and they are:

- The number of dense layers. It has been fixed to 1.
- The number of neurons in the dense layer. Dense layers after convolutional layers allows to classify features issued from convolutional layers.

Algorithms selection :

- Gradient descent algorithm is RMSprop.
- Loss function algorithm is binary cross entropy.

CNN hyper-parameters selection for optimization :



- Dropout rate. This parameter controls the model complexity and hence, its capacity to generalization.

- Learning rate. This parameter controls the speed of the learning.

- L1 regularization. This parameter controls the complexity of the algorithm, hence its bility to generalize. This regularization process is applied to the cost function. Complexity regularization is achived while killing some features in the learning process. L1 regularization is knwon to be less stable as L2 regularization, because loss function convexity is impacted with L1 terms.

- L2 regularization. This parameter controls the complexity of the algorithm, hence its bility to generalize. This regularization process is applied to the cost function

Classification results

Binary classification

This step has been performed by spliting toxicy values into 2 classes :

- 0.0 <= toxicity < 0.5 : safe comments
- 0.5 <= toxicity <=1.0 : toxic comments

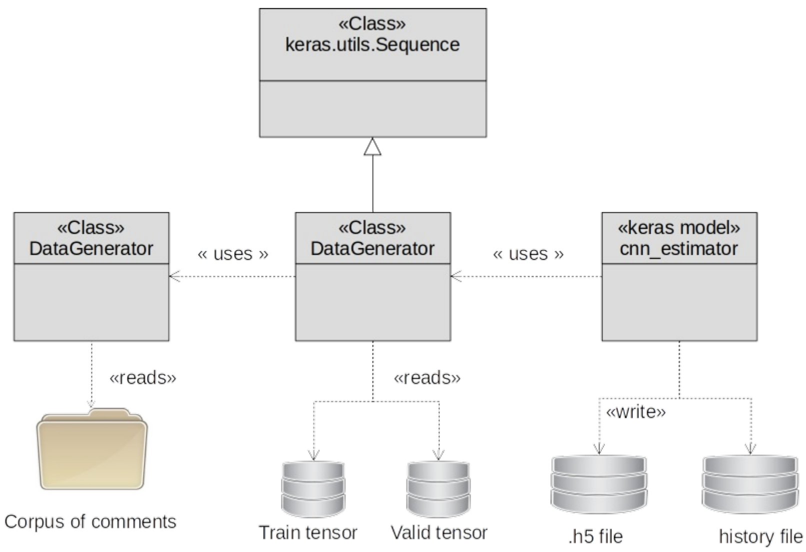
Data generators

Because of memory resources issues, `DataGenerator` objects have been created and used for both, trainig an validation operations.

Such objects allow to pump bulk of data stored on harddisk. Architecture below shows how the model works :

`DataGenerator` class inherites from `keras.utils.Sequence` This allows Keras model to use object issued from this class as a source of data, pumping it iteration after iteration in the train or validation operations, until all data partitioned into files over hard disk have been processed.

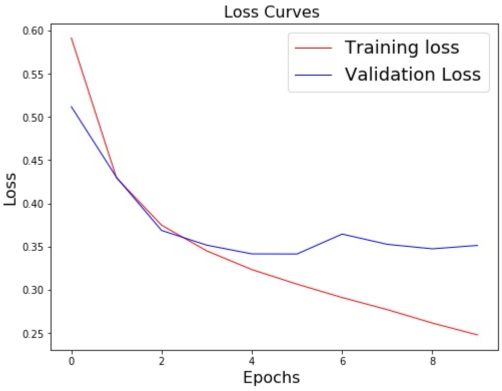
DataGenerator is created using DataPreparator object. Such objects contains all operations for data preparation and digitalization process along with configuration parameters used inside these process. This architecture allows an **automation digitalization process** to take place.

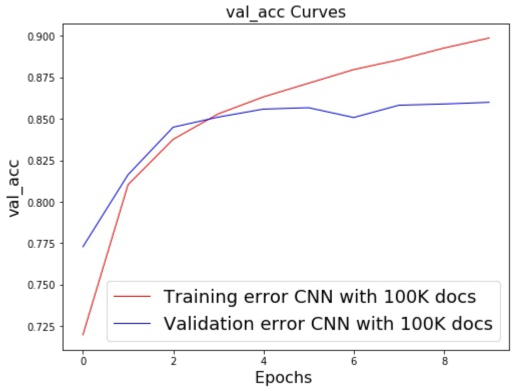


Depending on object configuration, data sources may be train dataset or validation dataset. A callback function, issued from object of type `keras.callbacks.ModelCheckpoint` allows to save the best model issued from train operation.

Training / validation performances

The number of trainable parameters is closed to 715K Model tends to overfit, despites dropout rate of 0.3



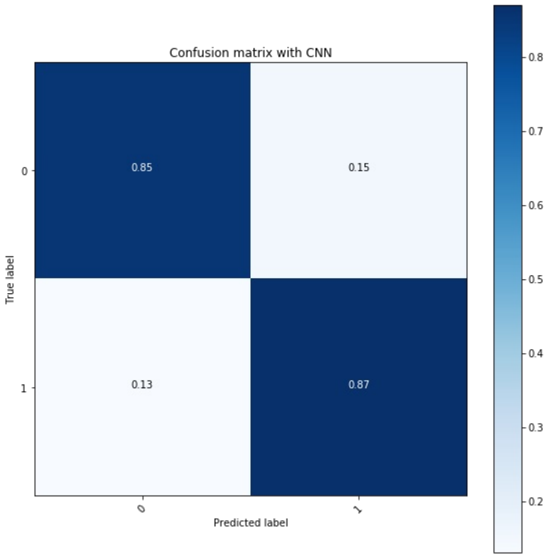


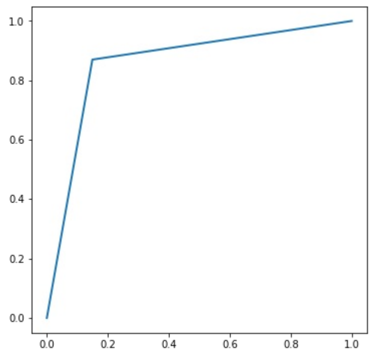
Classification performances

Results below have been obtained with training near to 100K comments and validated over closed to 10K comments. Confusion matrix below shows that model is able to properly classify :

- 85% of non toxic comments
- 87% of toxic comments

Area Under RO Curve is AUC=0.86





Bias results