

ML Project Presentation

Varsou Penny

Machine Learning - G.Gianakopoulos

16/02/2021

- 1 ML Project
 - Introduction and Overview
 - Project Presentation
 - Step-by-step
 - Text preprocessing - Feature extraction
 - Models creation
 - Models tuning
 - Results
- 2 Final thoughts
 - Conclusion
- 3 References

- Problem statement:
Detection of hate or offensive tweets in Greek language.
- Motivation:
Twitter is a major social networking service with over 200 million tweets made every day. According to a variety of researches nowadays more and more young people read news and form an opinion through social media

Step-by-step

- First step, found a csv with English tweets (<https://github.com/t-davidson/hate-speech-and-offensive-language>) and translate with google api code may be found in `tweet_translator.py`.
- Second step is about preprocessing our tweets using stopwords, find spaces, urls, retweets, tokensize.
- In third step we use Tfidf vectorize, pos tags and sentiment analysis in order to extract our final features..
- In fourth step we run our classification model and find for all classifiers run time and classification scores.
- In the last step we evaluate our scores with k-fold.

Text preprocessing - Feature extraction

During text preprocessing and feature extraction the below were done:

- We remove stopwords, identify urls, whitespaces and mentions and get standardized counts of them.
- We set data set to lower case, we stemming it, and we tokenize our text.
- We use POS (part of speech tagging), and TF-IDF in order to extract our features.

After all the aboves we have 18127 tweets and 5067 features.

For model creation the below four classifiers were used.

- Logistic Regression
- Gaussian Naive Bayes
- KNearest Neighbors
- XGBoost

- We split our dataset on train and test.
- We try our four classifiers, and calculate scores and time.
- We create confusion matrix for each one.
- We evaluate our scores using k-fold.

Results

In order to evaluate our models we use the below metrics:

- Accuracy: Due to the fact that our dataset is not a symmetric one and false positive and false negatives aren't almost same, we can evaluate our performance only with this.
 $(TP+TN/TP+FP+FN+TN)$
- F1-Score: Is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.
 $(2*(Recall * Precision) / (Recall + Precision))$
- Recall: Is the ratio of correctly predicted positive observations to the all observations in actual class - yes
 $(TP/TP+FN)$
- Precision: Is the ratio of correctly predicted positive observations to the total predicted positive observations.
 $(TP/TP+FP)$
- Time: We need to know and evaluate the cost of each classifier, in train and test part.

Results

		Logistic Regression	Gaussian NB	KNN	XGBoost
Accuracy		0.61	0.29	0.71	0.80
F1-Score	macro	0.44	0.28	0.39	0.44
	weighted	0.65	0.36	0.69	0.75
Precision	macro	0.43	0.40	0.47	0.61
	weighted	0.72	0.67	0.70	0.76
Recall	macro	0.48	0.39	0.39	0.42
	weighted	0.61	0.29	0.71	0.80
Time		17.44	1.77	107.33	211.92

Not a feasible project. The main problems occurs due to the fact that our dataset is inbalanced and the translation is not accurate. In general sentiment lexicons and word embeddings constitute well-established sources of information for sentiment analysis in online social media. Although their effectiveness has been demonstrated in state-of-the-art sentiment analysis and related tasks in the English language, such publicly available resources are much less developed and evaluated for the Greek language.

- <https://github.com/t-davidson/hate-speech-and-offensive-language>
- Machine Learning presentations
- <https://towardsdatascience.com/cross-validation-430d9a5fee22>
- <https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908>
- <https://pypi.org/project/googletrans/>