International Institute of Information Technology
Bangalore

Machine Learning

AI511

# Course Project

*Name of the student: B Laxmi Sreenivas*
(IMT2020510)

*Name of the student: Ganga Sagar Reddy*
(IMT2020134)

# Contents

# 1 Pre-Processing

## 1.1 Null Value Check

No Null Values Were Found in the Given Dataset.

## 1.2 Duplicate Value Check

Every sample has a unique ID associated to it. So, comparision of rows is done without including the column "id".

## 1.3 Expanding Contractions

Contractions are replaced their by their Expanded Counterparts in the dataset. This operation is done using "contractions" library.

## 1.4 Handling Emoticons

Emojis contain meaning information about the tone of the comment.So, they are not removed.They are replaced by a word(s) which represent the same emotion. This is done using "contractions" library.

## 1.5 Removing Special Characeters

At this point, special characters such as punctuation,.. etc carry no information. Hence they are discarded.

## 1.6 Removing Stop Words

Words such as "a","an","the" contain no information about the tone of the comment. Hence they are removed. Along with stop words Links are removed as well. (Because there is no automated way to check the tone of the redirected web page).

## 1.7 Removing Stop Words

Words such as "a","an","the" contain no information about the tone of the comment. Hence they are removed. Along with stop words Links are removed as well. (Because there is no automated way to check the tone of the redirected web page).

## 1.8 Lemmetization

Comments are now tokenized and then lemmetized (Removing -ing,.. suffixes). Parts-of-Speech Tagging is also done alongside lemmetization. This is the final step of pre-processing.

# 2 Feature Extraction

## 2.1 Word Vectorizer

TF-IDF Vectorizer is used. The Hyper parameters are as follows :

- lowercase = True
  All operations are done after the words are down casted to lower case.
- strip accents = "unicode"
  Accents on letters are stripped. They carry to little to no information about the tone.
- ngrams = (1,2)
  ngrams are used because they carry information of order of words in a comment.
- sublinear tf = True
  Logarithm is applied to Text Frequency. This makes sure that a word doesn't get too much "importance" just because it repeated more in text.
- min df = 2, max df = 0.5
  A word should occur at least twice in the document in order for the vectorizer to consider it (Removes unneccessary vocabulary).If a word occurs too many times (i.e at least half of comments have a word in common) then it carries little to no significance.

## 2.2 Character Vectorizer

TF-IDF Vectorizer is used. The Hyper parameters are as follows :

- lowercase = True
- strip accents = "unicode"
- ngrams = (2,6)
- sublinear tf = True
- min df = 2, max df = 0.5

## 2.3 Final Features

The features extracted from both word and character vectorizer are horizontally stacked. These columns are used to training in further stages.

** Bag of Words Vectorizer is considered but TF-IDF was found to be better. So, features extracted from this vectorizer were discarded.

# 3 Model Testing and Training

## 3.1 Multi Label Classification

We have 6 different labels to predict ("harsh","extremely harsh","vulgar","threatening","disrespect", "targeted hate"). All the labels are independent of each other. So, our problem can be viewed as 6 different binary classification problems.

## 3.2 Naive Bayes

Naive Bayes model is used to predict outputs of all labels. The final output accuracy was found to be 0.94 . (So, gaussian is not a good fit in our case).

## 3.3 Logistic Regression

Since gaussian seems to be not a fit, Logistic regression is used. Logistic Regression proved to be much better than Naive Bayes with intial accuracy of 0.962. This was further improved to 0.984 by tweaking the hyper parameters.

## 3.4 Random Forest

The focus was then shifted to "Tree Models" from "probablistic models" . Random Forest (with max depth of 100) gave an prediction accuracy of 0.96. The max depth was then increased to 1000 but this didn't increase the accuracy much. The training time of Random Forest is very high and results weren't satisfactory.

## 3.5 Chain Classifiers

Two Chain Classifier models were trained. (Naive Baye Chain and Logistic Regression Chain each of size 10). Chain Classifiers didn't improve accuracy much but increased training time. So, we moved from using Chain Classifiers.

## 3.6 Ridge Classifier

The focus was now shifted to linear classfiers. So, Ridge classifier was trained. Though the train accuracy and F1 score out performed all the models. The test scores was found to be the worst. It gave an accuracy of 0.68. This was a clear indication that the test data is not linear seperable. So linear classifiers such as perceptron are put on hold.

## 3.7 Support Vector Machine

The training of Support Vector Machine lasted 7hrs long and ended in a crash. So, no further trails were done using this model.

### 3.8 Logistic Regression CV

In order to boost the performance of Logistic Regression model, cross validation is done (K-fold Stratification). But this didn't improve it's performance.

### 3.9 Grid Search

Grid Search was performed on both logistic regression and ridge classifier in order to find the best hyper parameters combination. The performance did improve in the case of Logistic Regression.

### 3.10 Ridge Regression

There is no predict proba for ridge classifier. We can't make our own inference algorithm. So, ridge regression is used. Now, we set a threshold and predict the class accordingly. The best accuracy of this model is 0.981.

## 4 Results

### 4.1 General Model Trend

| Model | Accuracy |
|---|---|
| Logistic Regression | 0.982 |
| Naive Bayes | 0.94 |
| Random Forest | 0.969 |
| Ridge Classifier | 0.683 |
| Naive Bayes Chain | 0.941 |
| Ridge Classifier (Grid Search) | 0.611 |
| Ridge Regression | 0.981 |

### 4.2 Tuning Best Model

| Model Variation | Accuracy |
|---|---|
| Logistic Regression (StratifiedKFold) | 0.98111 |
| Logistic Regression (Grid Search) | 0.98412 |
| Logistic Regression (Chain Classifier) | 0.97933 |
| Logistic Regression (solver = liblinear,C=2) | 0.98411 |
| Logistic Regression (Updated Max Features) | 0.98414 |

### 4.3 Best Kaggle Score

The Best Kaggle Score : 0.98414. The Model was Logistic Regression.

## 5 References


- NLP
- contractions
- Naive Bayes
- Logistic Regression
- Chain Classifier
- Random Forest
- Ridge Classifier
- Support Vector Machine
- Logistic Regression CV
- Grid Search CV