Ask Reddit

DEMO 3

S-A-M

Manaswitha (IMT2019511)
Samhitha (IMT2019521)
Amitha (IMT2019023)

Feature Engineering

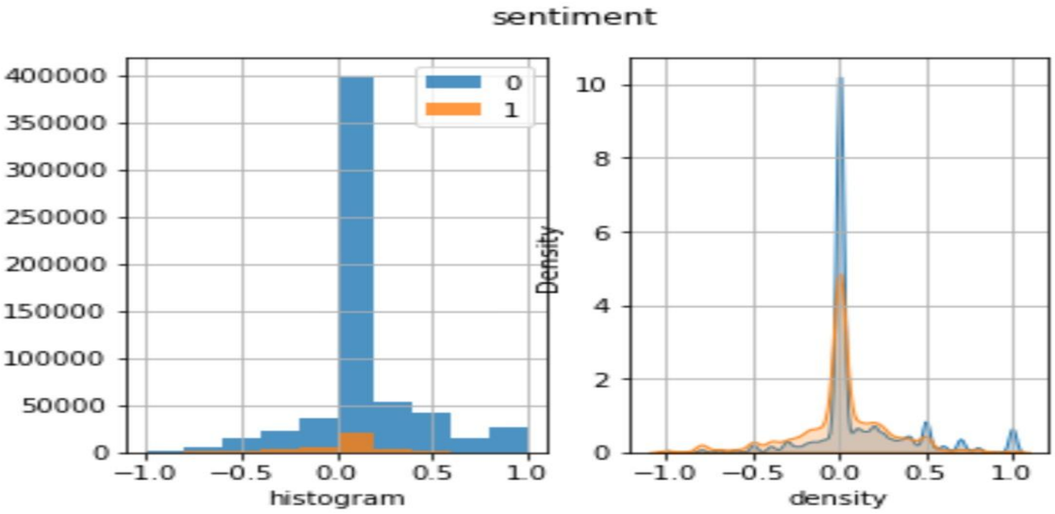
We extracted following features from dataset after preprocessing with removing punctuation, tokenization, removing stopwords, lemmatizing .
And also performed Standardization using StandardScaler library from sklearn.preprocessing .

1. Sentiment Analysis
2. Word Count
3. Sentence Count
4. Average word length
5. Average sentence length
6. Punctuation Count

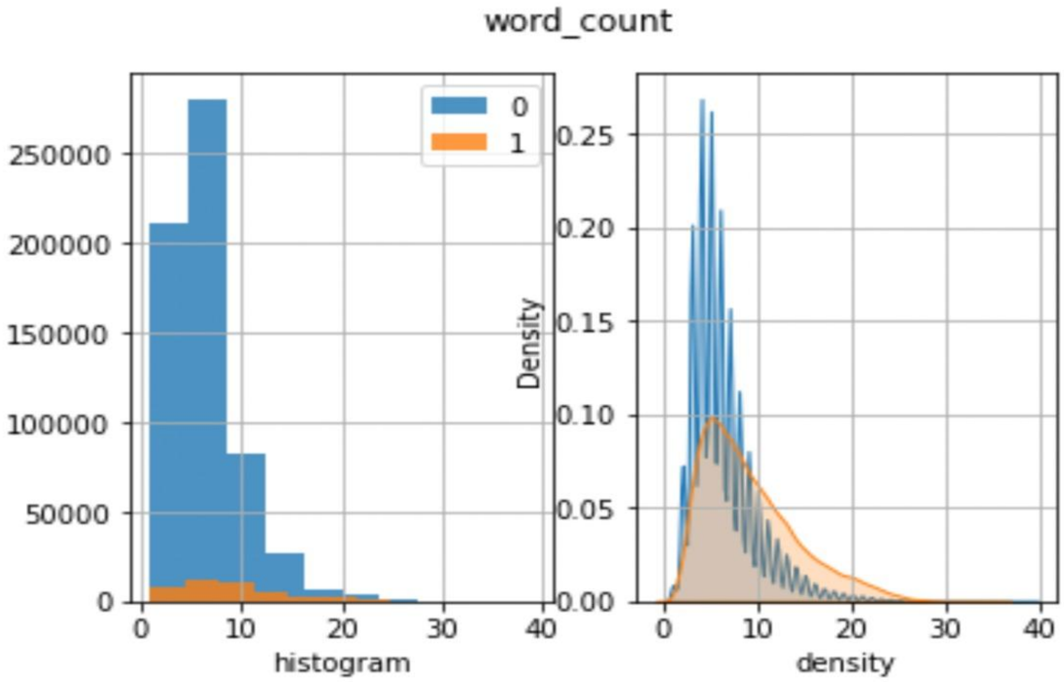
Exploratory Data Analysis

We plotted density and histogram curves for all the features to analyse the data.

Sentiment Analysis:

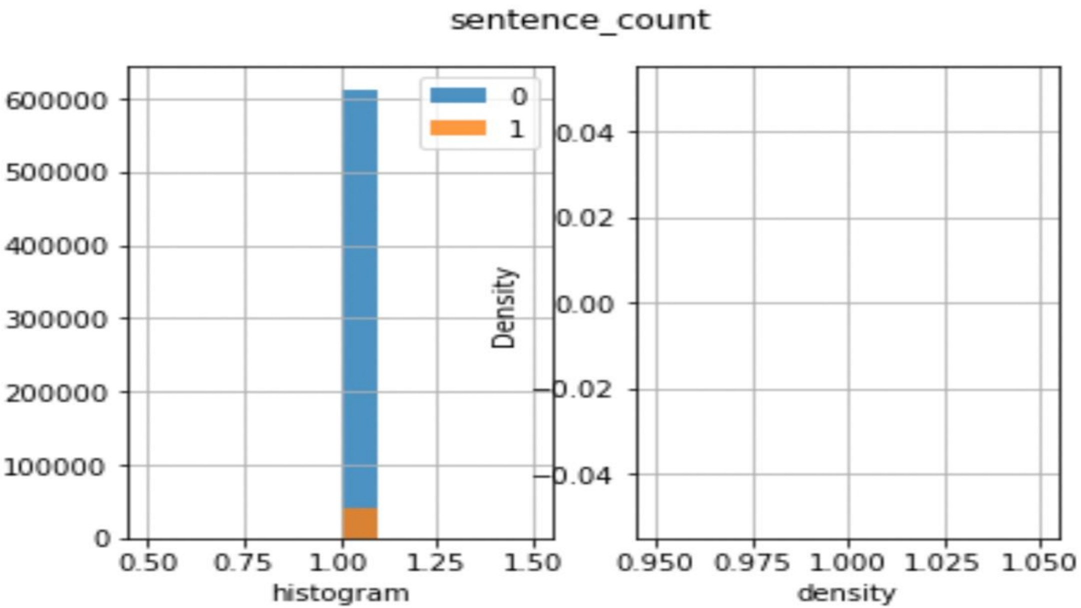


Word count :

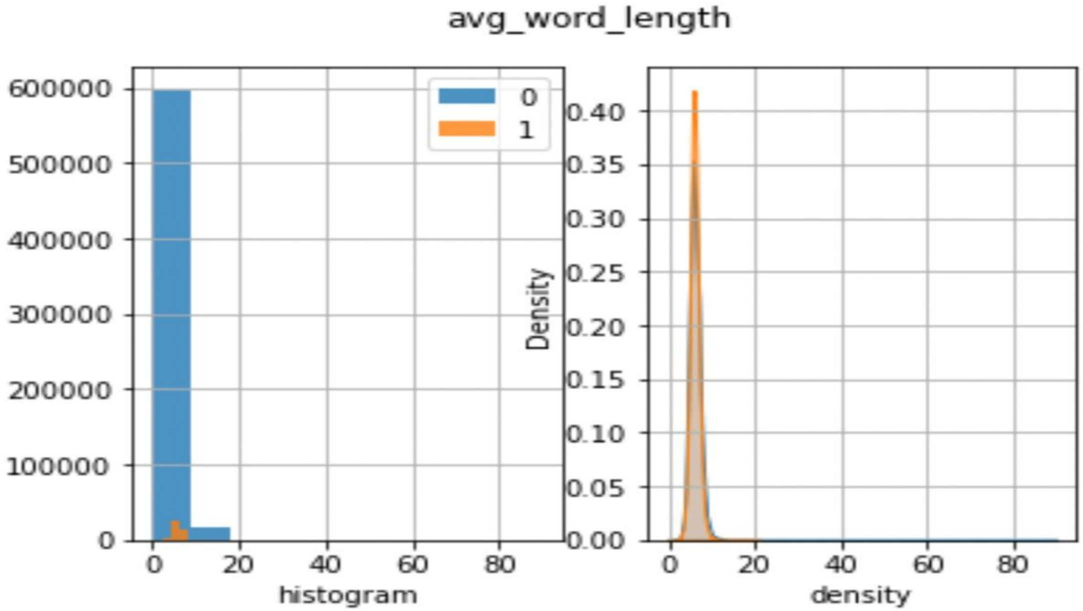


word_count
0
1
250000
200000
150000
100000
50000
0
0
10
20
30
40
histogram
Density
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
density

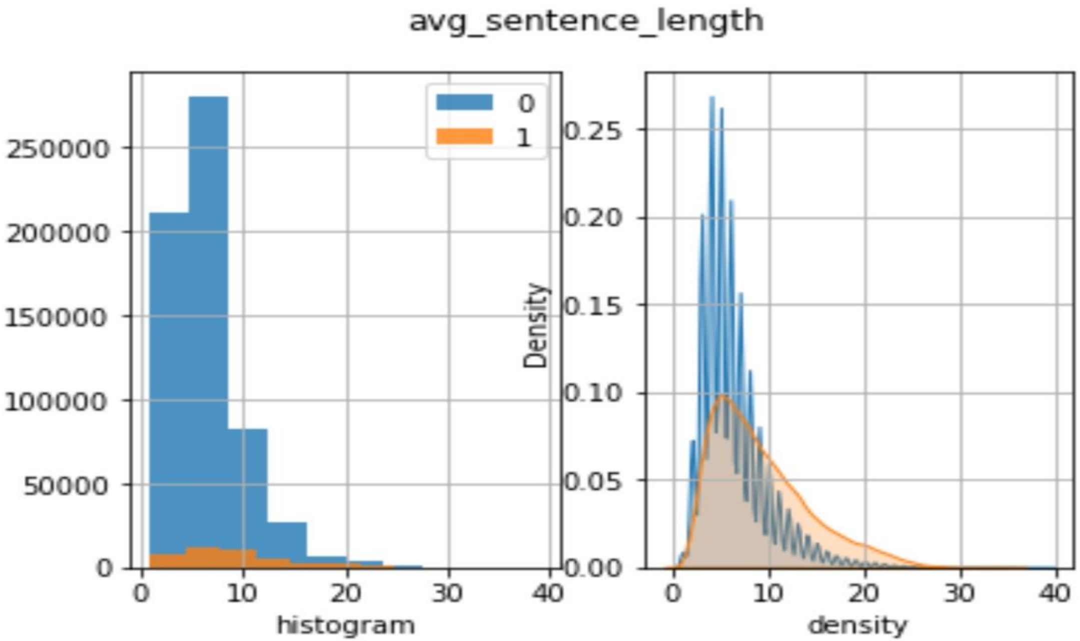
Sentence Count:



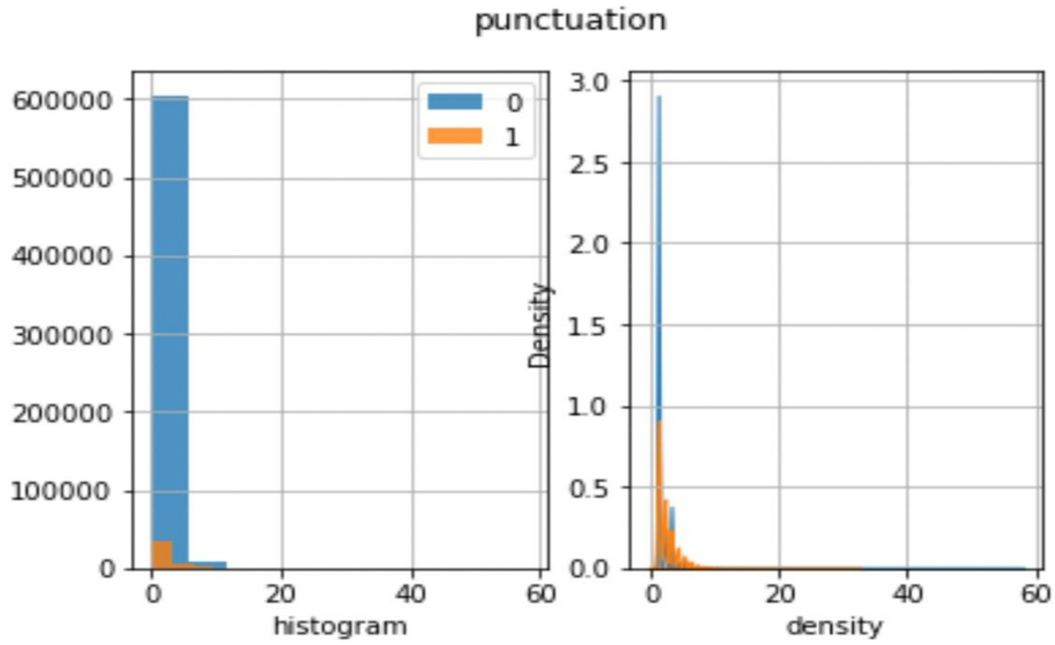
Average Word Length:



Average Sentence Length:



Punctuation Count :



Model Selection:

We have used a pipelined model because we had both text and numeric data.

For text features we tried the following Bag of Words models to convert them into numerical data - Count Vectorizer, TFIDF Vectorizer, Hash Vectorizer.

We sent both raw and preprocessed data to above three models and found that Count Vectorizer with raw data works best.Now this data is added to the numerical data which we got from feature enginnering

Pipelined models used on complete data- Logistic Regression, Linear Support Vector Classifier, Stochastic Gradient Decent, Random Forest Classifier, AdaBoost Classifier, Gradient Boosting Classifier.

We have tried out all the above classifiers using GridSearchCV to find the best fitting hyper parameters for the given data. Logistic Regression with the following hyper parameters gave best accuracy and f1score.

- class-weight = {0:0.19,1:0.81}
- maximum iterations = 7000
- tolerance = 0.002
- intercept scaling = 11
- solver = liblinear
- random state = 10

Class-weight

The purpose of class-weight is to penalize the misclassification made by the minority class by setting a higher class weight and at the same time reducing weight for the majority class.

There is a threshold to which you should increase and decrease the class weights for the minority and majority class respectively. If you give very high class-weights to the minority class, chances are the algorithm will get biased towards the minority class and it will increase the errors in the majority class.

We tried with class_weights = ‘balanced’ also but taking ratios gave better results.

In our dataset majority class is non-troll(label - 0) for this we assign less ratio and for minority class we assign higher ratio. After doing many trails for class-weight = {0:0.19, 1:0.81} in logistic Regression we got better results.

Thank you