CSCI 544: Applied Natural Language Processing
Course Project: Age Classification for Youtube Captions

This project attempts to annotate and classify Youtube videos taking into account the content of the video and the composition of the text/captions. While youtube flags content inappropriate for young audiences by requiring viewers to sign in, a lot of youtube content is generally unaudited if the uploader of the video does not flag it as appropriate for a particular age group. Also there is no distinction as to what content is appropriate for which age groups. In this project, we will classify content based the film rating system: *G, PG, PG-13 and R*.

Due to the large volume video data and thereby captions, most of which are accurate as they are uploaded by the uploader themself, and a dearth of any classification or publicly available corpus (a public corpus of youtube captions which analyses a large volume of data does not exist), this is a unique opportunity to tap into a very rich resource. There is a huge opportunity to annotate massive amounts of data and to gather insights with respect to the kind of content and the kind of audience it appeals to.

We have been able to successfully collect data from over a **1322** youtube videos from various channels spanning gaming, vlogs, talk shows , technology, and travel. We have used a third party service to get the captions of videos. The data will be labelled with respect to the film rating guidelines. Some of the parameters on which the age appropriateness is gauged are violence, abusive language, substance abuse, etc. In case of abusive language, the context in which this language is used is also considered by reviewing the article manually. (Eg. whether a threat or abuse is directly intended at a person or just used as a colloquialism or expression of angst, disgust etc.).

We have come up with the list of words which are divided into two categories: *profane* and *offensive.* Offensive words have a stronger implications and hence are restricted to only the R group or PG-13 in some contexts. We have labelled data manually and have also used helper scripts in case the class of data is very obvious. The basic classification algorithm is stated below:

Data Labeling Algorithm for classifying Documents:

Input: File ← List of caption words/subtitles for a video
Output: fileLabel

```
If offensiveWordsCount == 0:
    If profaneWordsCount <= 2:
        fileLabel ← 'G'
    Else if profaneWordsCount >= 2 and profaneWordsCount <= 4:
        Review file manually.
        If context not sexual, abusive or violent:
            fileLabel ← 'PG'
        Else:
            fileLabel ← 'PG-13'
Else if offensiveWordsCount >= 1:
    If offensiveWordsCount >= 1 and offensiveWordsCount < 4:
        Review File Manually.
        If context not sexual, abusive or violent:
            fileLabel ← 'R'
        Else:
            fileLabel ← 'PG-13'
    Else if offensiveWordsCount >= 4:
        fileLabel ← 'R'

Return fileLabel
```

In case of conflicts, as mentioned in the algorithm above, files are reviewed manually. In such cases, the file written by the automating script (which outputs a JSON) has a placeholder in place of the label and also writes a flag to another file which maintains the names of videos that are to be reviewed. The automating script always checks if this file exists prior to processing a video and skips the video if the video name exists in the file.

Inter-annotator agreements between team members: While reviewing the data manually, all team members have adhered to the following guidelines:

1. The definition of a text being sexual, consists of sexual (innuendoes or explicit) remarks being passed about an entity as opposed to just using a word that has the said connotations which might be very common in the colloquial parlance in a particular region.

2. The definition of a text being violent, consists of explicit violence, gory details about killing, suicide, etc. that may be intended about an entity. This does not refer to jokes or catch phrases occuring in a nonchalant context.
3. Substance abuse is treated carefully, with regards to the context. (Eg is the video promoting substance abuse or does it have a neutral standpoint regarding the issue, is it factual and presenting data for scientific or informational purposes or does it reflect the uploaders personal experiences, what kind of audience (age group) does the author have in mind as his intended audience, etc.)

Data Collection:

1. This list of offensive and profane words is collected from *CMU research group offensive/profane words list, noswearing.com dictionary, google's banned word list, youtube's banned word list.*
2. Text for captions is collected from a youtube's api.
   https://developers.google.com/youtube/v3/docs/search/list
3. Both the sources of data are cleaned to remove the timestamps and punctuation, capitalization and collated into single text from a file.

Classifier Approach:

To come up with the approach, we have tried several machine learning and text processing techniques and tested them using 5-Fold Cross Validation. With cross validation, we have evaluated the classifier on 5 different test sets. This strategy ameliorates the problem of overfitting. We have analyzed Precision, Recall and F1 scores of the classifiers for each of the ratings ["G", "PG", "PG13" and "R"] and reported the averaged scores for each configuration. Since, there is some imbalance in the classes, Weighted F1 score is a good measure to evaluate the One-Vs-All Multi-classification problem. Following are the approaches that we tried and improved:

1. **Baseline approach:**
   a. A simple Naïve Bayes algorithm that takes counts of unigrams as inputs.
   b. Only considering 1-grams.
   c. Laplace Smoothing (alpha = 1.0)
2. **Naïve Bayes with TF-IDF:**
   a. Here, we have used TF-IDF values instead of counts.
   b. We have also included n-grams ranging from 1 to 6.
   c. Total number of n-grams are 500 (To reduce variance)
   d. Laplace Smoothing (alpha = 0.2)
3. **Logistic Regression with Counts:**
   a. Input takes counts of unigrams.
   b. Logistic Regression with L1 penalty = 1.0 (Regularization)
   c. Only considering 1-grams.

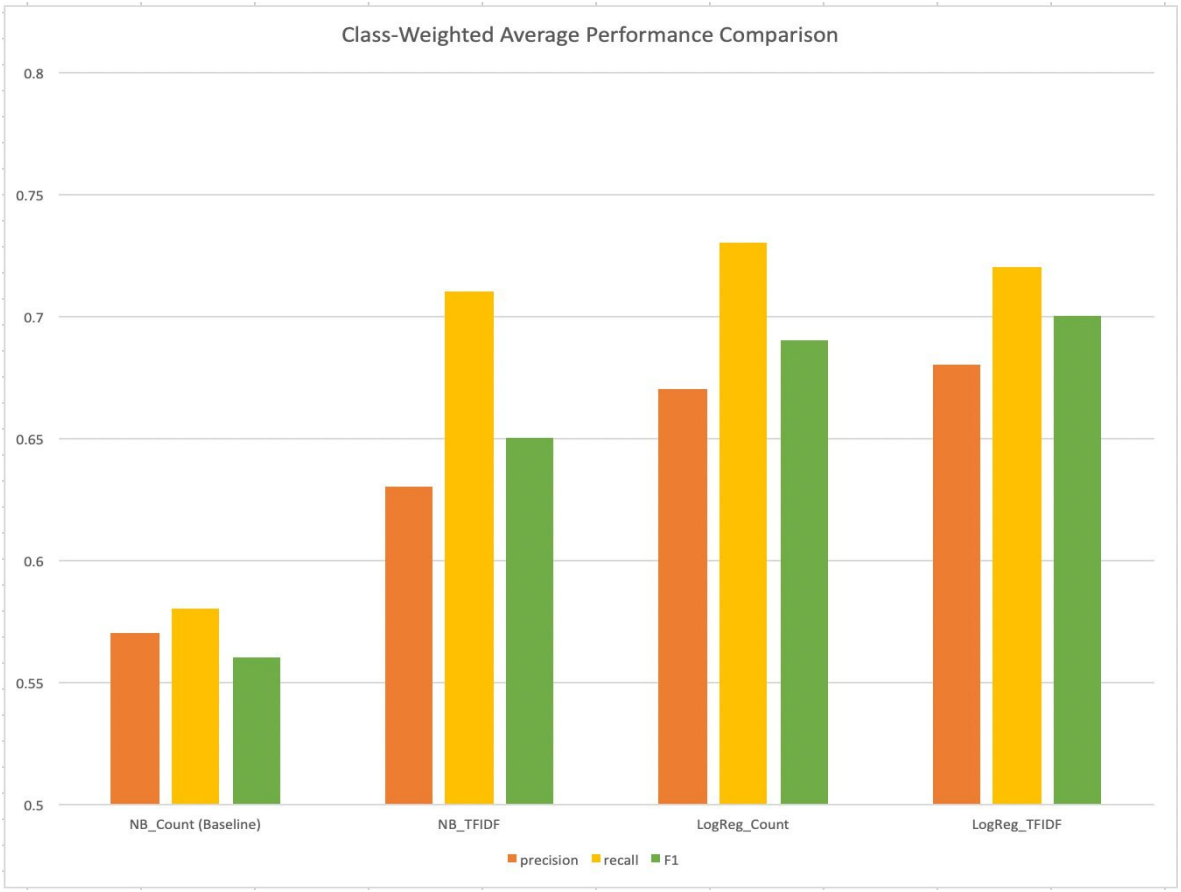
4. **Logistic Regression with TF-IDF:**
    a. Here, we have used TF-IDF values instead of counts.
    b. We have also included n-grams ranging from 1 to 3.
    c. Logistic Regression with L1 penalty = 10 (Regularization)
    d. Total number of n-grams are 500 (To reduce variance)

Observations:

Here are the observations after performing *Grid Search Cross Validation*:

- Decreasing smoothing parameter helps to increase the performance as the model now deals better with underflow.
- Including n-grams in the model helps in increase the performance.
- Decreasing the feature space improves performance significantly by making the model more robust to overfitting.
- Logistic Regression performs better with a higher L1 penalty because it deals with overfitting by selecting the most important features.

Analysis and Benchmarking of results:



Class Distribution (%):

```
Class Distribution:
G       41.98
R       28.44
PG13    21.86
PG       7.72
```

Train - Test Split: 3:1

Result comparison for each approach used:

1. **Baseline Approach:**

```
Running Naive Bayes Classifier with Word Counts


Cross Validation Weighted F1 Score: 0.590920535001


Classification Report:
             precision    recall  f1-score   support

          G       0.78      0.57      0.66       139
         PG       0.00      0.00      0.00        26
       PG13       0.31      0.39      0.35        72
          R       0.62      0.91      0.74        94

avg / total       0.57      0.58      0.56       331
```

2. Naïve Bayes with TF-IDF:

```
Running Naive Bayes Classifier with TF-IDF Vectors, Laplace Smoothing and N-Grams


Cross Validation Weighted F1 Score: 0.630946894109


Classification Report:
             precision    recall  f1-score   support

          G       0.77      0.92      0.84       139
         PG       0.00      0.00      0.00        26
       PG13       0.56      0.25      0.35        72
          R       0.67      0.94      0.78        94

avg / total       0.63      0.71      0.65       331
```

3. Logistic Regression with Counts:

```
Running Logistic Regression with Word Counts


Cross Validation Weighted F1 Score: 0.685936223447


Classification Report:
             precision    recall  f1-score   support

          G       0.75      0.91      0.82       139
         PG       0.00      0.00      0.00        26
       PG13       0.55      0.42      0.47        72
          R       0.83      0.91      0.87        94

avg / total       0.67      0.73      0.69       331
```

4. Logistic Regression with TF-IDF:

```
Running Logistic Regression with TF-IDF Vectors, L1 Regularization and N-Grams


Cross Validation Weighted F1 Score: 0.708991478197


Classification Report:
             precision    recall  f1-score   support

          G       0.77      0.86      0.82       139
         PG       0.00      0.00      0.00        26
       PG13       0.54      0.44      0.49        72
          R       0.85      0.90      0.88        94

avg / total       0.68      0.72      0.70       331
```