**Stumble Upon Challenge**

By – Arkya Bagchi

**Problem Statement**

StumbleUpon is a user-curated web content discovery engine that recommends relevant, high quality pages and media to its users, based on their interests. In this challenge we would classify the webpage as "ephemeral" or "evergreen" by evaluating a large set of URLs.

**Data**

There are two components to the data provided for this challenge:

The first component is two files: train.tsv and test.tsv. Each is a tab-delimited text file containing the fields outlined below for 10,566 URLs total. Fields for which no data is available are indicated with a question mark.

train.tsv is the training set and contains 7,395 URLs. Binary evergreen labels (either evergreen (1) or non-evergreen (0)) are provided for this set. test.tsv is the test/evaluation set and contains 3,171 URLs.

**Approach**

We are using a traditional method to solve the Stumble-Upon Challenge. In the proposed work, it being divided into four stages.

1. **Data Extraction**

From the training dataset we extracted the useful features which included URL id, Boilerplate information which contains the title and body and finally collected all the labels associated with each URL.

1. **Data processing**

We dropped the rows with empty bodies or whose length is less than 20. Then the extracted quotations are processed before classifying by removing the stop-words and then stemming the quotations followed by removing the numerical characters and lowered case all the alphabets. We merged the title and text body for better representation of the data.

1. **Feature Extraction**

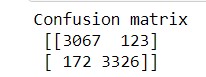
For feature extraction, I am using TF-IDF (Term Frequency and Inverse Document Frequency) model [1, 2, 3] for representing the words and characters in form of vectors. Since sparse representation is very cheaper to store as we only need to store the entries of the matrix, this is a very useful tool to reduce the space complexity of our problem. Finally, horizontally stacked the sparse representation of word and character vectors.

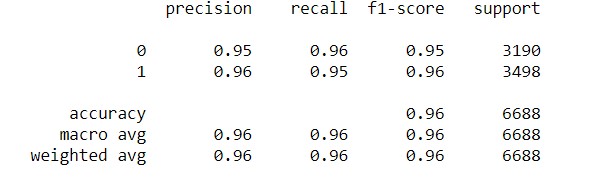
1. **Model representation**

Classification is performed with the use of Logistic Regression Model [4] with inverse regularization strength of 2 and class\_weight as ‘balanced’.

**Result**

Obtained the submit.csv file which contains the prediction of the classification problem.

 Obtained the confusion matrix and found the precision and recall for each class.



**References**

[1] K. S. Jones, “A statistical interpretation of term specificity and its application in retrieval,” *Journal of Documentation*, vol. 28, no. 1. MCB UP Ltd, pp. 11–21, 1972, doi: 10.1108/eb026526.

[2] P. Bafna, D. Pramod, and A. Vaidya, “Document clustering: TF-IDF approach,” in *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, Nov. 2016, pp. 61–66, doi: 10.1109/ICEEOT.2016.7754750.

[3] C. Z. Liu, Y. X. Sheng, Z. Q. Wei, and Y. Q. Yang, “Research of Text Classification Based on Improved TF-IDF Algorithm,” in *2018 IEEE International Conference of Intelligent Robotic and Control Engineering, IRCE 2018*, Oct. 2018, pp. 69–73, doi: 10.1109/IRCE.2018.8492945.

[4] “Logistic Regression for Machine Learning.” https://machinelearningmastery.com/logistic-regression-for-machine-learning/ (accessed Mar. 07, 2021).