StumbleUpon Evergreen Classification Challenge

(Website Classification Problem)

Sumit Roy (sumitroy@stanford.edu)
Shailin Saraiya(shailin@stanford.edu)

Abstract

Web classification is a very important machine learning problem with wide applicability in tasks such as news classification, content prioritization, focused crawling and sentiment analysis of web content. In this project, we primarily focus on developing prediction model using machine learning techniques for one such problem that classifies if a web posting is of eternal relevance, known as evergreen or is short-lived. We finally apply the optimized model to a different web classification problem, namely the subject classification of an website content.

1. Introduction

Classification plays a vital role in many information management and retrieval tasks. On the Web, classification of page content is essential to focused crawling, to the assisted development of web directories, to topic-specific Web link analysis, to contextual advertising and to analysis of the topical structure of the Web. Web page classification can also help improve the quality of the advertising, and help improve the quality of Web search. Customer reviews or opinions are often short text documents which can be classified to determine useful information from the review.. Automated methods can be very useful for news categorization in a variety of web portals.

The goal of our project was to study a specific instance of this broad and vital web classification problem and developing a successful prediction system. We selected the StumbleUpon Evergreen Classification Challenge, that requires building a classifier to categorize webpages as evergreen or ephemeral. Most news articles or seasonal recipes are relevant for only a short period of time while others are relevant forever and can be recommended to users anytime. Therefore, based on the relevancy of a webpage, it can be classified as “evergreen” or “non-evergreen”. In general such classification is made on the basis of users’ ratings. If machine learning algorithms are used then such distinctions can be made ahead of time which in turn improves the quality of recommendation engine. The rest of the project report is organized as follows: Section 2 describes our initial model based on Naïve Bayes classification, followed by feature selection and several other enhanced model. In section 3, we present a pipelined model that improve the accuracy over models in previous section. In section 4, we apply the model developed in section 3, to the subject classification problem. We conclude our report in Section 5 along with plans for future.

2. Naïve Bayes

Before presenting our classification models, let us first explain the dataset used for all our experiments. StumbleUpon.com[4] provided the parsed dataset of 7395 webpages along with user defined labels whether they are evergreen or not. The dataset contains 2 types of data

- Textual: Boilerplate that includes title, keyword and body
- 24 Meta-data: common-link ratio, spelling error ratio, HTML tag ratio, subject classification, etc

For comparing our models, we used 20-stratified cross-validation for training and testing on these datasets. We measured the progression on our improvement using ROC_AUC score. A receiver operating characteristics (ROC) graph is a technique for visualizing, organizing and selecting classifiers based on their performance. ROC-AUC score computes the area under the curve for ROC [3]. Accuracy score is based on one specific cutpoint, while ROC tries all of the cutpoint and plots the sensitivity and specificity. So ROC_AUC gives a more robust metric compared to accuracy score, hence we used that for qualifying all our models.

As suggested in the class [1], we started by implementing the Multinomial Naïve Bayes using the boilerplate as the feature set. Figure 1. shows the results using the Naïve Bayes running on 81079 tokens.

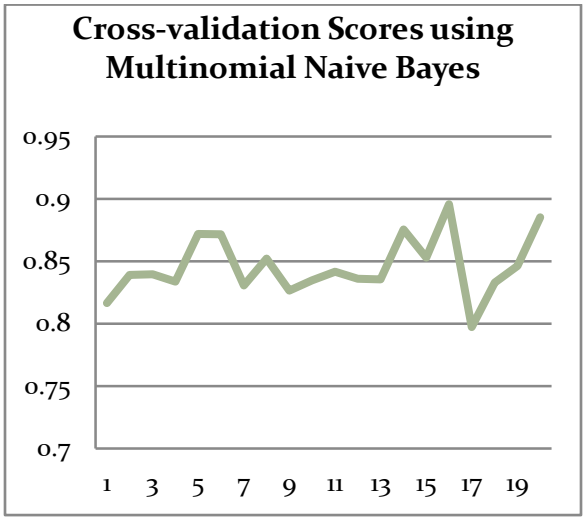


Figure 1. Results using Naïve Bayes

The average ROC_AUC score was 0.845. The first step in improving our model was to improve the feature selection. We started by analyzing the tokens with the largest positive log(⍰⍰ and negative log(⍰), which is shown in the Figure 2. below. The tokens in red (stop-words) are not good for distinction but Naïve Bayes gave high weightage to them.

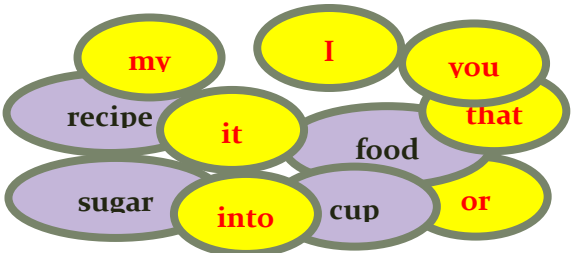
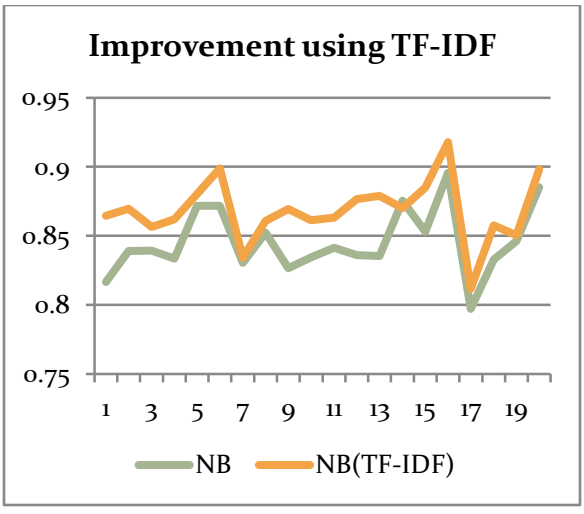


Figure 2: Tokens with highest log(ϕ),

2.1 Feature Selection

We implemented mutual information based feature selection to filter out the stop words as mentioned in the class. It did show some improvement in filtering out most of the stop words, but several of the stop words (like my, I, you) still got into top-100 log(⍰⍰ score. The key thing that was missing from mutual information metric is the importance of word frequency in a given document as well as the rarity of occurrence of that work. Hence we looked at TF-IDF (term frequency-inverse document frequency) [2]. We performed feature selection using TF-IDF and then applied that to the Multinomial Naïve Bayes Classifier. Figure 3. shows the comparison between that and the one without TF-IDF filtering. The average ROC_AUC score improved from 0.845 to 0.868.



2.3 Figure 3: Results using TF-IDF

After finalizing the feature selection methodology, our next step was to start analyzing ways to improve the model. We reviewed the boilerplate of the elements that was classified as false negative. We found the following types of information:

- Empty body (158)
- Advertisements (995)
- Tabloid news (1286)
- Bogus police report (1656)

where the element in the bracket is the item number that displays the behavior mentioned. Having an empty body or a tabloid news labeled as an evergreen clearly shows that there are human errors introduced during the manual classification. Given the noise in the dataset, we need to inspect models that perform regularization, like SVM or logistic regression. We applied logistic regression (referred as LR in rest of the report) with L2 estimator and tuned the parameter C (regularization factor) using GridSearch. Figure 4 shows the results compared to Naïve Bayes with TF-IDF. As you can see, the ROC_AUC score improved using LR for most of the cross-validation point, average ROC_AUC improved from 0.868 to 0.877.

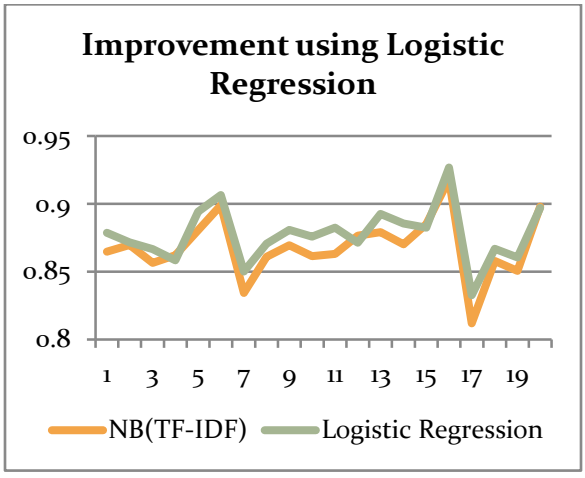


Figure 4: Results using LR

We continued our analysis to improve the model further. While investigating the false-positive and false-negative items, we noticed that certain subjects has more false positives than others and certain others have more false negative than others.

Since the training data had subject classification, we plotted the LR classified probability of each item against positive and negative label per subject. Figure 5 shows that classification, where each red line displays probability for all items with label 1 for each subject and similarly green represents for label 0. One can clearly see that each subject should have a different classification point. We decided to implement a Gaussian Discriminant Analysis (GDA) on the output of the LR fitted for each subject, the hope being that GDA would be able to identify the mean and variance to decide the classification point as well as the confidence level (probability of the classification) better. So the classification system was TF-IDF on the boilerplate, followed by LR on tokens for a given subject, followed by GDA per subject.

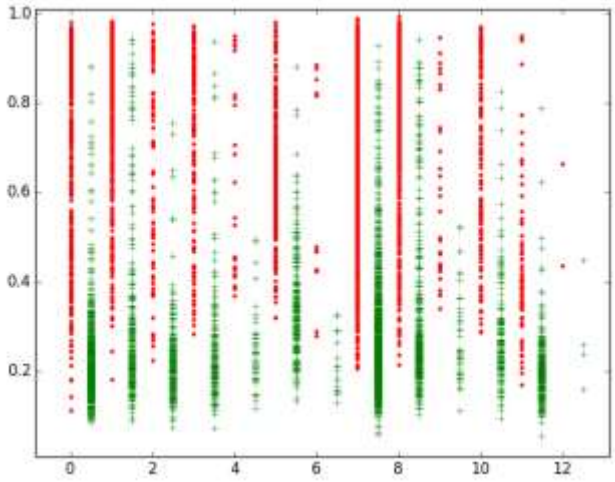


Figure 5: Subject vs LR probability

Unfortunately, the average ROC_AUC score did not improve, as shown in Figure 6. Although the accuracy was better because of the subject specific classification point, the probability of the false positive/negative data was pushed to the extremes (0 and 1).

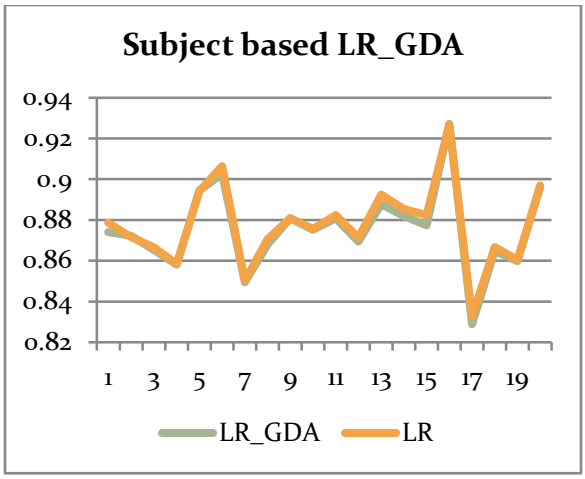


Figure 6: Subject based LR_GDA vs LR

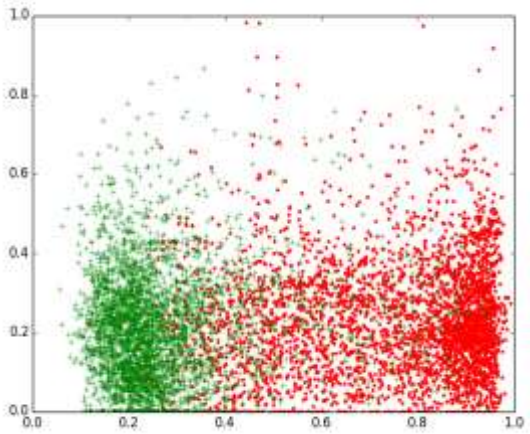


Figure 7: Link-ratio vs probability of LR

Next we started investigating if the model can be improved using a subset of the 24 non textual feature set. We plotted the probability of the LR classification of positive labels (red) and negative labels (green) against each of the non textual feature set. Figure 7 shows the plot using common-link ratio (y axis) vs the probabilities of boilerplate using LR(x axis). Visually, it clearly showed that none of the non-textual feature set had any additional discriminative power. We verified that by running a LR with featureset as the classification probability of the boilerplate based LR and each of the non-textual dataset. The average ROC_AUC scores from the 20-fold cross validation was much worse than just the simple LR. This clearly demonstrated that non-textual features are not very useful feature for our model. In the next section, we will talk about pipelining strategy using multiple textual feature set.

3. Pipelined Model

Given that the non-textual features lacked differentiability, we started investigating meta data within the boilerplate that could potentially improve the classification accuracy. We generated a LR model with keywords as the feature set. The ROC_AUC score for the model was good (in the range of 0.85), which indicates that keywords have good potential of classifying the dataset. We performed similar experiments based on words in the title of the webpage as feature set and got similar ROC_AUC score. Encouraged by this, we implemented a combination of ensemble and pipelined model as shown in Figure 8. We start with 3 distinct feature sets, namely title, body and keywords of the webpage, transformed it through TF-IDF filter and then trained 3 LR models for each respectively. Finally, we took the classification probability of each of those models and trained the final LR model to provide the final classification. We used our

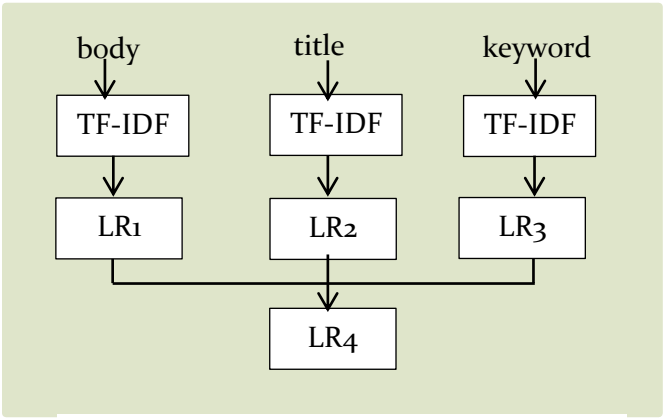


Figure 8: Final Classification Model

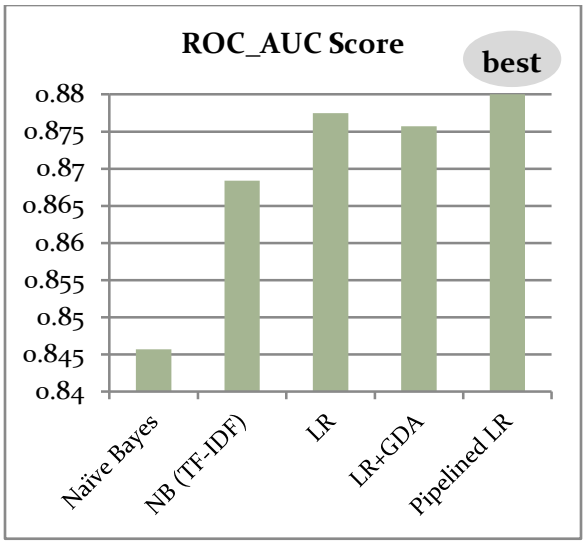


Figure 9: Comparison of various models

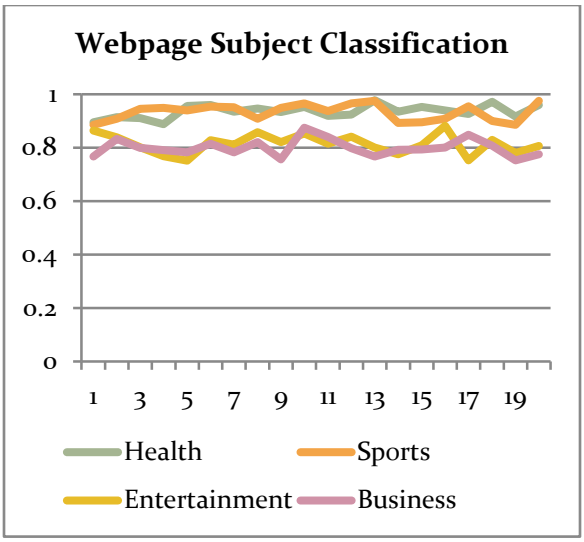


Figure 10: Pipelined LR model

dataset to train and test this model and the results are shown in Figure 9. As can be seen, this combination of feature-set along with pipelining and ensemble strategy gave us the best ROC_AUC score.

4. Subject Classification

As mentioned in the introduction section, the goal of our project was to develop model for a specific web classification problem and then extend it for other website classification problem. The problem that we decided to explore was classifying the subject of the content of a webpage. The StumbleUpon dataset had that classified as one of its meta-data. We used that as the label and applied our model developed in Section 3. The results are shown in Figure 10. We observed that subject contents that are very precise like "Health" and "Sports" were classified with more than 0.9 ROC_AUC score, but subjects like "Business" and "Entertainment", which have more broader definition ended up with lower scores.

5. Conclusion

The webpage classification problem is vital to many web-mining applications and we presented a method to effectively solve the StumbleUpon Evergreen Challenge. We were encouraged by the results seen on applying the model to a different webpage classification, namely subject classification. In future, we plan to extend our model for sentiment analysis of webpages and emails as well as study ways to model noise better.

6. Reference

[1] Andrew Ng, CS 229, Class Lecture Topic

[2] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries", ICML, 2003

[3] Tom Fawcett, "An introduction to ROC analysis", Pattern Recognition Letters 27 (2006)

[4] www.stumbleupon.com