Accuracy of machine learning models used to predict Reddit comment origin

Group 59 - Matt D'Iorio, Daniel Borisov, and Joseph Szymborski

Abstract—Machine learning serves an important role within the field of natural language processing and classification tasks. Reddit, a popular internet forum, can provide many insights into different social groups and their sentiments. With different themed subforums, termed subreddits, various topics are discussed across the platform. Understanding the differences between the comments on these subreddits can give insights into the different mentalities and cultures curated by the different population groups, and serves as a good testing ground for the incorporation of machine learning models for the use of natural language classification. A training dataset of 7000 comments, evenly distributed across 20 different subreddits, was used to test different preprocessing steps and machine learning models and their ability to classify the subreddit from which the comments originated from. Tokenization, URL analysis, stemming, and term frequency-inverse document frequency transformations were assessed for their use in origin classification and were analyzed through a variety of different sci-kit learn models, a deep learning model, and an internal implementation of naive Bayes. It was found that utilizing a sci-kit learn implementation of complement naive Bayes produced the greatest accuracy, beating out every other model tested with a cross-validation accuracy of 57% and a test set accuracy of 56.20%. Compared to other groups, it was found that this model underperformed in comparison. Future work includes utilizing more preprocessing steps to look at sentiment, text readability, and semantic analysis to improve classification accuracy.

1 Introduction

In this paper we evaluate language processing, feature selection, and classification methods to predict the origin of comments written on the reddit website. Reddit is a popular social media website containing categorized interest groups called subreddits where individuals can post and comment on content related to each interest. We have developed and evaluated several pipelines for data cleaning, feature selection, classification, and validation to determine the efficacy of each strategy.

Text classification performance relies heavily on the processing of speech data into numeric features. The transformation from paragraphs to vectors involves basic data cleaning, tokenization, and feature selection. To further improve the input data, feature engineering techniques are used to rank the importance of each variable and reduce the dimensionality of the overall dataset. Methods such as the term frequency-inverse document frequency (Tf-Idf), and linear support vector classification (SVC) are implemented to reduce dataset noise and therefore increase the accuracy and speed of each model.

The actual prediction task of assigning each comment to a subreddit was split between several different custom and pre-built learning models that were evaluated under different conditions. This by implementing classification models from the Sci-kit Learn library such as logistic regression, decision trees, support vector machines, and Naive Bayes classifiers [1]. We also developed and implemented a custom naive Bayes and a convolutional neural network classifier. Ultimately, the complimentary naive bayes classifier from sci-kit learn had the best accuracy compared to the other models tested.

Project submitted Oct 20, 2019

1.1 Related Work

Techniques for classification of comments from social media platforms have been extensively applied to improve context-aware machine learning pipelines. Given the large and growing proportion of open-source text data from platforms such as twitter and reddit, classification algorithms have been developed to predict a wide range of endpoints. For instance, Twitter data been used to train models that can predict relevant categories of information during crisis events [1]. Collections of reddit comments have been used to create models that

Subreddits	Comments	Average Word Count
anime	3500	48
AskReddit	3500	48
baseball	3500	36
canada	3500	54
conspiracy	3500	54
europe	3500	52
funny	3500	34
gameofthrones	3500	46
GlobalOffensive	3500	36
hockey	3500	37
leagueoflegends	3500	42
movies	3500	44
Music	3500	76
nba	3500	35
nfl	3500	43
Overwatch	3500	50
soccer	3500	37
trees	3500	36
worldnews	3500	53
wow	3500	50

TABLE 1
Total count of comments in each subreddit and average number of words over each set of comments.

predict instances of cyberbullying and posts written by users with anxiety [2][3].

Most notably, text classification of reddit comments by subreddit has been conducted before by Gutman and Nam (2015), where the best accuracy achieved was 77.3% by using a support vector machine (SVM) model on a processed bag-of-words (BOW). This study was conducted with 7 distinct subreddits: NFL, News, Movies, PCMasterRace, and Relationships. We looked to expand in this area by evaluating a larger corpus with more subreddits. Our subreddits also will include more thematically closely related interests such as hockey, soccer, baseball, and NBA. The expansion of data, and the correlation between subreddits will provide new challenges to the reddit comment classification problem that we will address throughout this paper.

2 Dataset and Setup

The reddit training dataset contains 7000 comments evenly distributed among 20 different subreddits. Each subreddit contains a relatively even set of comments, and is further described by **table.1**:

Comments were tokenized using the spaCY tokenizer and stop list [8]. The Porter stemmer was used, as implemented by the `NLTK` library. The text was extracted from comments containing markdown text. This was achieved by parsing markdown into HTML using the `mistune` library, and the text was extracted from the HTML string with the `beautifulsoup4` library. All text was transformed to lower case before proceeding with the analysis.

2.1 URL Analysis

We prepared a second dataset where additional information about in-comment URLs was analysed. URLs were extracted and excluded from the comment body using a regex filter. Domain names were included in the tokens analysed, as well as sub-domains in a recursive fashion (*e.g.*: the URL `https://a.b.c.com/some/path` would generate the tokens `<DOMAIN PART a.b.c.com>`, `<DOMAIN PART b.c.com>`, `<DOMAIN PART c.com>`, and `<DOMAIN PART com>`).

Links to other subreddits were included in the tokens (*e.g.*: the URL `https://reddit.com/r/canada/some/path` would generate the token `<SUBREDDIT /r/canada>`).

Titles of the webpages linked in comments were also tokenized and processed in the same way as comment bodies and added to the analysis. Links to YouTube videos included the title of the videos similarly tokenized. This was achieved by scraping the pages using the `Selenium` library.

Ultimately, this dataset did poorer in cross-validation than a second on where URL data was not included in a Complement Naive Bayes model. For that reason, all subsequent analyses are on the dataset without URL information.

2.2 Feature Selection

To reduce the dimensionality of the dataset, a linear SVC with L1 regression was trained on the dataset, and a coefficient threshold of 10^{-5}. This was achieved with the sklearn function `LinearSVC` and `SelectFromModel`.

2.3 TF-IDF

Counts were transformed to reflect their importance using the TF-IDF metric. This was achieved using the `sklearn`'s `TfidfTransformer` class.

3 Proposed Approach

In developing a pipeline from text processing to classification, we applied several basic preprocessing steps and experimented with different language processing techniques applied to different classification models. Given the heterogeneity of the

comments within our dataset, we developed models and pipelines that varied in approach for classification. For the preprocessing, we experimented with tokenization, TF-IDF transformation, URL-parsing, and data reduction. For model implementation, our Sci-kit learn methods were decision trees, SVMs, random forests, logistic regression, and a complementary naive Bayes. These models provided a reference point to compare the performance of subsequent methods. Our naive Bayes classifiers gave important points of comparison with respect to the speed and accuracy when compared to our deterministic methods. Alternatively, given demonstrated speed and accuracy of the approach, we implemented a convolutional neural network (CNN) to the dataset as another classification benchmark.

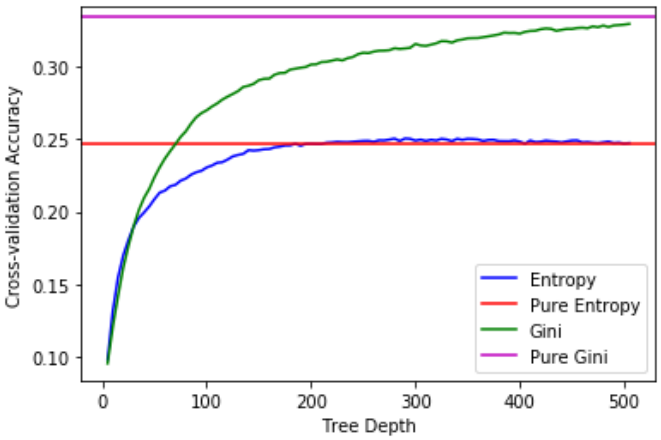

Fig. 1. Comparison of the gini and entropy-based decision tree models, showing accuracy as a function of max depth.

4 Results

4.1 Sci-kit Learn Models

A large variety of different classification models were compared and contrasted in order to determine the ideal model for reddit comment origin classification. stratified 10-fold cross-validation was utilized to determine the validity of the model in classifying the origin, and to select the best model. The models tested were Sci-kit Learn library classifiers: the decision tree classifier, support vector machine, random forest classifier, logistic regression classifier, and a complement naive Bayes classifier. Finally, ensemble voting models were also used to determine the validity of utilising multiple models for origin classification.

The first model that was tested was the decision tree classifier. Two internal parameters of the model were varied to determine the best decision tree model for the classification problem. Due to a lack of a post-pruning implementation within the decision tree classifier library within Sci-kit Learn, the max-depth was varied instead to find the best classifying tree. This was done with both the entropy-based decision calculation, as well as the gini-based decision calculation. It was found that the gini-based classifier outperforms the entropy-based classifier, and that the accuracy of the models increased with max-depth, eventually hitting a plateau. The cross-validation accuracy is shown as a function of max depth for the two decision making processes in **Fig.1**. The max accuracy of the two models were found to be 25.04% and 32.94% for the entropy-based decision and the gini-based decision, respectively.

Secondly, the support vector machine model with a linear kernel was tested. Class probabilities were utilized in order to allow for the implementation of the SVM model into a soft-voting ensemble classifier. The accuracy of this model was found to be 51.88%.

The third model tested was the multinomial logistic regression classifier. Four different solver algorithms were tested in order to determine the solver that provided the best accuracy, which utilised an L2 regularization method. These four solvers were the newton-cg solver, lbfgs solver, sag solver, and the saga solver. It was found that all but the lbfgs solvers had equal accuracy, while the lbfgs solver had slightly reduced accuracy. Regression model iteration count was varied to assess the impact on the classifier performance, with the results summarized in **Fig.2a**. Furthermore, logistic regression models using the saga solver were tested using elasticnet regularization. **Fig.2b** shows the cross validation accuracy of the models as a function of the elasticnet L1-L2 regularization ratio, where the greatest accuracy results from a ratio of zero, corresponding to pure L2 regularization. The L2 regularization factor was then varied, and the resulting accuracies plotted in **Fig.2c**. Thus it was determined that the best logistic regression model was the model that utilised L2 regularization with a regularization factor of 1.75, and a cross-validation accuracy of 53.7%.

After, a random forest classifier was implemented. The number of estimators was varied between 1 and 21, and the results are summarized in **Fig.3**. The results show that the best accuracy of the random forest classifier was 45.48%, resulting from a model that utilised 21 estimators. The trendline

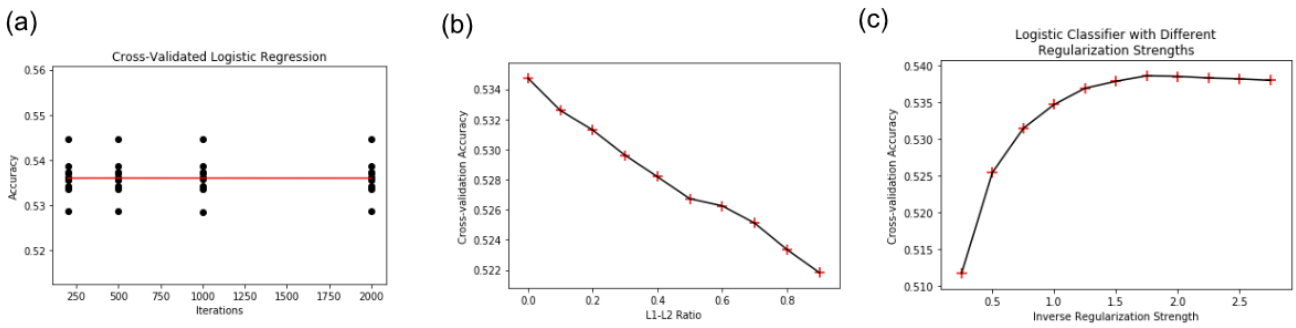

Fig. 2. Logistic regression model performance on comment origin classification. (a) Different iteration values. (b) L1-L2 ratio of the elasticnet regularization and its effect on accuracy. (c) Accuracy as a function of the L2 regularization parameter.

indicates that the model accuracy would continue to increase with more estimators, but would hit an asymptote below 50% accuracy.

Finally, the complement naive Bayes model was tested for comment origin classification accuracy. The additive smoothing parameter was varied in the model in order to determine the best smoothing to provide the greatest cross-validation accuracy. The results of the parameter variation on accuracy is summarized in **Fig.4**. It was found that the best cross-validation accuracy arose from using a smoothing parameter of 1.375, producing an accuracy of 57.25%.

Furthermore, ensemble methods were incorporated in order to test a soft-voting system utilising various combinations of the tested Sci-kit Learn models. However, a soft-voting ensemble method was unable to hit the performance level obtained from the complement naive Bayes model by itself. A final summary of maximum cross-validation classification accuracies of all tested models are shown in **Fig.5**, showing that the highest accuracy was achieved using the complementary naive Bayes model, while the worst method was the decision tree classifier.

Thus, due to its accuracy, the complementary naive Bayes model was used in the relevant Kaggle competition in order to test its predictive ability compared to the results obtained from models developed by other groups. It was found that the test set classification accuracy used in the competition was 56.20%. The test-set accuracy of the model slightly under performed when compared to the TA baseline of 57.333%, and placed 97th in the competition.

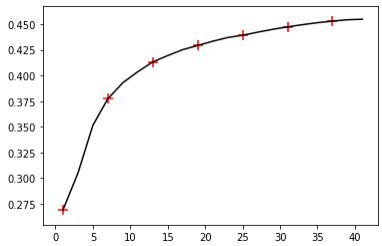

Fig. 3. Performance of the random forest ensemble method in predicting subreddit comment origin as a function of the number of estimators.

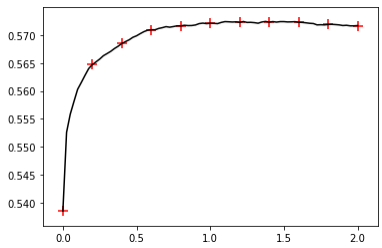

Fig. 4. Performance of the complement naive Bayes classifier in predicting subreddit comment origin as a function of the smoothing parameter.

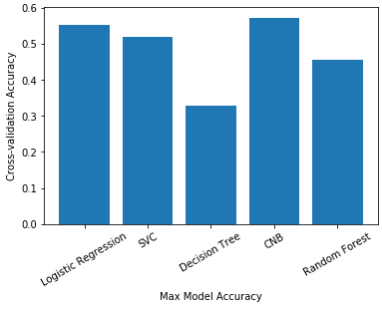

Fig. 5. Performance of the different classifier models at predicting comment subreddit origin

4.2 Internal Naive Bayes Implementation

A self-developed naive Bayes model classifier was produced and compared to the accuracy of the Sci-kit Learn models. It was found that this model was able to achieve 20% accuracy when trained on 10000 comments and tested on 50 held out comments. Vocabulary was reduced by L1 normalization (13289 words).

4.3 Convolutional Neural Network Model

A convolutional neural network was trained on the dataset. A word embedding layer, a one-dimensional convolutional layer, and two dense layers were applied. Dropout and GaussianNoise layers were also included to combat over-fitting. Ultimately, the model did overfit, but achieved a maximum of 48% on a held-out test set after 7 epochs.

5 Discussion and Conclusion

It was found that the model that produced the best cross-validation accuracy on the reddit comment origin classification task was the complementary naive bayes, with a classification accuracy of 57.25%. The test-set classification accuracy of this model was 56.20%. Given that the model underperformed when compared to the test-set accuracies of the top groups within the competition, it can be concluded that, while the model works well for the multi-class classification task, there are still many improvements that can be made in order to improve the performance. As well, it can also be concluded that there exists a large amount of potential preprocessing steps, and that testing all of the possibilities involves significant computational time in order to create the best possible classification model.

One of the possible directions for trying to improve the classification accuracy of the task involves better data preprocessing. It is possible that there are other natural language processing features that could have improved the model accuracy. These include semantic language processing, in order to try and obtain the meaning of the comment, rather than simply looking at the words and characters. Furthermore, it is possible that sentiment analysis as a model feature could have also produced better accuracy results, as it is possible that certain subreddit comments tend to carry more negative sentiments, while others are more cheerful and happy. Finally, a last preprocessing step could have been to implement a readability scale, as subreddits aimed at an older or more educated user base might have more complex sentence structure when compared to some other subreddits, and as a result could have helped improve classification accuracy.

Another potential direction for improving classification is through utilising more intricate models. One potential model to try could be a multi-layer perceptron network to see if a more complex model might provide better classification results. As well, it is possible that certain deep learning networks might produce significant improvements in the natural language processing and classification of the comment data, and as a result a potential future step could include the implementation of transformer deep learning models to try and significantly improve the accuracy of the origin predictions.

6 Statement of Contributions

All three group members provided equal contributions to the project. Joseph Szymborski contributed to the data preprocessing steps necessary for the classification task, while Daniel Borisov tested the various Sci-kit Learn models and their cross-validation performance. Furthermore, Joseph Szymborski implemented the internal naive Bayes algorithm. Similarly, equal contributions were made towards the writeup, with Matt D'Iorio contributing to the Introduction, and Proposed Approach, Joseph Szymborski contributing to the Dataset and the Deep Learning results, and Daniel Borisov contributing to the Sci-kit Learn Results and Discussion. A joint effort was put into the Abstract and proof-reading.

References

[1] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., and Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J, Passos, A., Cournapeau, D., and Brucher, M., Perrot, M., and Duchesnay, E. (2011) JMLR 12. 2825-2830.
[2] Chen, T. and Guestrin, C. 2016. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 785-794.
[3] Imran, M., Mitra, P., Castillo, C. (2016). arXiv.
[4] Bin Abdur Rakib, T. Soon, L.-K. 2018. Intelligent Information and Database Systems 180-189.
[5] Hanwen Shen, J. and Rudziez, F. 2017 Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology 58–65.
[6] Gutman, J. and Nam, R. (2015). Technical report, New York University.

[7] Bird, S., Loper, E., and Klein, E. (2009), Natural Language Processing with Python. O'Reilly Media Inc.
[8] Honnibal, M., Ines, M. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing