Mini-Project 2: Reddit Comments Classification

Shashank Murugesh
McGill University
shashank.murugesh@mail.mcgill.ca
260786369

Anthony Ho
McGill University
anthony.ho@mail.mcgill.ca
260501840

Julien Courbebaisse
McGill University
julien.courbebaisse@mail.mcgill.ca
260614548

***Abstract*—The task at hand for this project was to build a classifier for reddit comments from 20 subreddits, ie 20 classes. Different models were tested such as Logistic Regression, Multinomial Naive Bayes, and Support Vector Machines from the sklearn library. The Bernouilli Naive Bayes Model was also implemented from scratch but did not offer the best accuracy.**

Some findings were in the importance of task-specific text preprocessing combined with tfidf vectorization. The training data for this project was composed of 70000 reddit comments with their respective id's and corresponding subreddits. The test data, with no corresponding subreddits, was made up of 30000 reddit comments.

This project was also part of a Kaggle Competition on which our team's best accuracy came out to 58.55%. This was achieved using a Multiple Layer Perceptron model with a single hidden layer.

I. Introduction

The first part of our work was to pre-process raw comments into a matrix of adjusted word frequencies with the tfidf vectorizer. This way of processing the data obtained better accuracy than simple count vectorization as it makes up for words that are naturally used more frequently. The data was also cleaned prior to tfidf vectorization to remove potential noise.

The non-exhaustive list of models tested for this project is: SVM, Multinomial Naive Bayes, Bernoulli Naive Bayes, Logistic Regression, Decision Tree, Gardient Boosting, Random Forest, and Multi-Layer Perceptron. The Multinomial Naive Bayes, SVM, and MLP performed best. The Bernouilli Naive Bayes Implementation reached an accuracy of 54.6%. The Multinomial NB model also scored better than the MLP during 5-fold validation but the MLP obtained better accuracy on the test data.

Some methods used during testing of models were Grid Search, which allowed us to search the hyperperameter space of each model. K-fold validation was also used to validate models and avoid overfitting on a given validation set.

II. Related work

Text Classification is a well know topic in Machine Learning. It presents elements from Natural Language Processing and Classification. Some well known models are Support Vector Machines which are convenient for text classification as they offer high dimensional input space [1]. For text classification as presented in this project Multi-Layer Perceptrons models are also known to perform well[2]. The TFIDF and Word2Vec methods are also know to provide best features for model and perform well for text classification[3].

III. Dataset and setup

The given training dataset contains three fields: id, comments and subreddits. The comments field has user text and subreddits contains twenty different categories. Each comment is associated with a specific subreddit category. We train our model on comments and subreddit dataset.

The user comments contains both useful information and noise data. Hence, its important to remove noise data from comments before training our model. This step is called preprocessing.

In preprocessing, we perform two main steps. First, we remove special characters, punctuation, numbers, whitespaces, url links, stopwords, high frequency words and rarely occurring words which do not provide any context to our model. Second, we extract vocabulary and corresponding Term Frequency(TF) and Inverse Document Frequency(IDF) is computed from comments which is to be used in text classification.

$$TFIDF \text{ score for term } i \text{ in comment } j \ = TF(i,j) * IDF(i) \tag{1}$$

$$TF(i,j) = \frac{\text{Term } i \text{ frequency in comment } j}{\text{Total words in comment } j} \tag{2}$$

$$IDF(i) = \log_2 \left(\frac{\text{Total comments}}{\text{comments with term } i} \right) \tag{3}$$

After performing TFIDF, we select the best features based on variable ranking technique. In our project we compute chi-square test. This test measures dependence between stochastic variables that helps in removing the features that are the most likely to be independent of class and therefore irrelevant for classification[4].

IV. Proposed Approach

In order to classify comments data belonging to subreddit category. We trained our dataset on different supervised learning classification models. This model learns from the given input comments and uses this learning to classify new comment.
Below are the different models which we implemented:

A. Logistic Regression

Logistic Regression is one of the simplest classification algorithm. It uses log odds ratio as the dependent variable and predicts the probability of occurrence of a event using sigmoid function.

B. Bernoulli Naive Bayes

Multi-class Bernoulli Naive Bayes model used in this project is implemented using the two-class Bernoulli Naive Bayes algorithm in class with the following modifications.

$$\hat{P}(c) = \frac{\text{\# of samples of class } c}{\text{Total \# of samples}} \tag{4}$$

$$\hat{P}(w|c) = \frac{\text{\# of samples where } w \text{ in class } c + 1}{\text{Total \# of samples in class } c + 2} \tag{5}$$

$$\hat{P}(w|c') = \frac{\text{\# of samples where } w \text{ in class not equal to } c + 1}{\text{Total \# of samples not in class } c + 2} \tag{6}$$

During the training of Naive Bayes model, the probability of each subreddit class $\hat{P}(c)$ and the probabilities of each vocabulary w in subreddit class c (i.e.,$\hat{P}(w|c)$) and not in a subreddit class c (i.e.,$\hat{P}(w|c')$) are computed using the formula above. The probabilities obtained will be used in calculating the posterior log probability of test sample X during class prediction. Laplace smoothing is also used in calculating the probabilities for the vocabulary which allowed the assignment of non zero probability to vocabulary which do not occur in the sample.

$$\hat{P}(c|x) = \log\left(\frac{P(c)}{1-P(c)}\right) + X * \log\left(\frac{P(w|c)}{P(w|c')}\right) + (1-X) * \log\left(\frac{1-P(w|c)}{1-P(w|c')}\right) \tag{7}$$

During classification, the probability of comment x belongs to a subreddit class will be computed using the formula below. The subreddit class with the highest $\hat{P}(c|x)$ will be the prediction for the test sample x.

C. Multinomial Naive Bayes

Multinomial Naive Bayes model estimates the conditional probability of a particular word w given a class c.
The most probable class c given comment x is given by

$$P(c|x) = \log(P(c)) + \sum_{w \in V} \log(P(w|c) \tag{8}$$

where w is word in vocabulary V for each class c.
$P(c)$ and $P(w|c)$ are estimated as follows:

$$\hat{P}(c) = \frac{\text{\# of samples of class } c}{\text{Total \# of samples}} \tag{9}$$

$$\hat{P}(w|c) = \frac{\text{Total \# of words } w \text{ in class } c + 1}{\text{Total \# of samples in class } c + |V|} \tag{10}$$

During prediction, the subreddits are classified based on highest $P(c|x)$.

D. SVM

SVM is one of the most popular and widely used classification algorithm. SVM constructs a hyperplane in multidimensional space to separate different classes. It learns optimal hyperperamaters which helps in prediction of new data point.

E. Multi-Layer Perceptron

Multi-Layer Perceptrons belong to the class of feedforward artificial neural networks. Hidden nodes in this network have non-linear activation functions. This helps them separate data which is not linearly separable and hence achieve more accuracy. MLP's work well on huge datasets and the model was prone to overfitting on this medium-sized dataset.

V. Results

We performed grid search in order to obtained best hyperparameters for our model. After which, five fold cross validation was performed to evaluate the performance of our models. Table V shows mean accuracy across five fold of different models. Multinomial Naive Bayes model has the highest accuracy score of 59.84% , while Logistic Regression has an accuracy score of 54.50%.

Model	Accuracy
Logistic Regression	0.54504
Bernoulli Naive Bayes	0.54601
Linear SVM	0.58514
MLPClassifier	0.58564
Multinomial Naive Bayes	0.59844

TABLE I
Compression of different models with there mean accuracy

Fig.1 shows the accuracy of five folds on various models such as Logistic Regression, Linear SVM, Multinomial Navie Bayes, MLP Classifier.

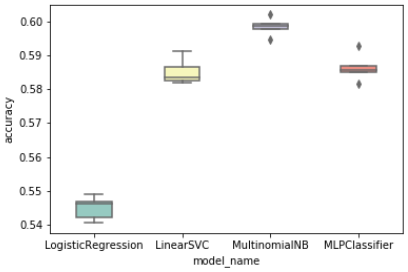

Fig. 1. Comparison of accuracy across various models

Multiple Layer Perceptron model with a single hidden layer gave us the best test accuracy score of 58.55%.Thus, this model was used in Kaggle competition.

VI. Conclusion

Some key takeaways from this project are 1) Models performed better when we vomit most frequently occurring words and rare occurring words. 2) Variable ranking helped in feature reduction without compromising the performance. 3)

Although Multinomial Naive Bayes has accuracy of 59.84% on five fold cross validation data it didnt perform well on test set. On the other hand, MLP with accuracy of 58.56% on validation dataset helped us to take our test accuracy to 58.55% in kaggle competition. Further scope of this project would be to try various models including Ensemble methods, LSTM, bert algorithm and deep neural network. Also, we can explore various techniques such as word2vec, variable ranking and Best-Subset selection for feature extraction.

VII. Statement of Contributions

All groups members contributed equally to this project. Table II lists contribution from different group members.

Shashank Murugesh	Anthony Ho
Dataset processing and Parameter tuning Implemented Logistic Regression, Linear SVM, Multinomial NB Running tests Report	Dataset processing Implemented Bernoulli NB Running tests Report

Julien Courbebaisse
Dataset processing Implemented MLP, Parameter tuning Running tests Report

TABLE II
Statement of contributions

References

[1] M. B. N. G. Mohamed Goudjil, Mouloud Koudil, "A novel active learning method using svm for text classification," *International Journal of Automation and Computing*.

[2] A. N. Sukhjit Singh Sehra, "A review paper on algorithms used for text classification," *International Journal of Application or Innovation in Engineering Management*.

[3] Z. Meilin, "Research on text classification method based on multi-type classifier fusion," *Advances in Computer Science Research*.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "sklearn feature selection chi2," *https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html*.