COMP 551 Project 2
Reddit Comments Classification

Yifei Zhao, Weige Qian, Yichao Yang

Abstract

In this project, we investigate several classification algorithms on the Reddit comments. Among all the algorithms we tried, Multinomial / Bernoulli Naive Bayes, Logistic Regression and Linear Support Vector Machine have great accuracy (more than 50%). Before applying these classification algorithms, we tried stop-word, lemmatization, count vectorization and TF-IDF vectorization as the preprocessing work to transform the content of comments into numerical data that algorithm could analyze. By the comparison of the different test results, it is shown that the combination of TF-IDF vectorization and Multinomial Naive Bayes has the best achievement on the accuracy of comments classification. Stopword and lemmatization did not have obvious improvement in accuracy. But they improve classification efficiency by reducing the number of useless features.

We also implemented our own model ensemble method and compared the performence with the voting method imported from ensemble package, it turns out that our own method showed a better result.

text preprocessing | Reddit comment classification | feature engineering | sci-kit learb | K Fold Validation | TF-IDF | Naive Bayes | ensemble

1 Introduction.

1.1 Dataset information.

Tons of comments posted from Reddit could be classified and predicted their subreddit by analyzing words and sentence in the comments. In this project, we predict these subreddits based on the word it contains, the term frequency and TF-IDF. Dataset we used is a csv file containing the original comments from Reddit. we also used the method in sci-kit and NLTK package to fetch the useful data from the set of contents which the algorithm can analyze.

1.2 Task.

The whole task contains several parts. First, we need to develop a Bernoulli Naive Bayes classifier without the external library. Before applying the word, we need to extract the dataset that the classifier could analyze. About this point, we are free to use any text preprocessing tools, including tokenizing words, vectorizer, tf-idf. In feature engineering task, we even tried PCA method to extract the most important and valuable features, but unfortunately it raise error 'computation cannot be performed with standard 32-bit LAPACK'. This may because the dataset is too large when handling eigenvalue matrix. So This can also be viewed as a topic for future invesigation, which we discuss in 'Conclusion' section Next part, besides the Bernoulli Naive Bayes, we also need to implement the experiment with two more classifiers with the help of the sci-kit library. After all the models are developed, we need to implement a validation pipeline to evaluate the performance of all the models we developed.

1.3 Important findings.

From this project, we have some important findings. First, the text preprocessing tools, stopword and lemmatization, have limited effect on the accuracy improvement of classification. It seems that more original features would behave better than simplified features in this experiment. We think this may because the train samples are still not large enough, so we cannot see a considerable improvement. But since the result did not degrade. We think they are still worthy to be used during text preprocessing since the efficiency is improved with no doubt.

Then, before we run the tests, we think the SVM will bring us the highest accuracy. But, by comparing the result from the different classifiers, we found the combination of the TF-IDF vectorizer and multinomial would result in the highest accuracy among all the tests which is out of our expectation.

1.4 Ethical concerns.

The ethical issues surrounding machine learning is not simply about the algorithms themselves but more about the way the data is used. As algorithms expand their ability to organize society, politics, institutions, and behavior, sociologists have become concerned with the ways in which unanticipated output and manipulation of data can impact the physical world.

In the case of *Reddit Comments Classification*, we use comments from Reddit to classify those comments based on different machine learning algorithms. However, without permission from users, we should be careful about the comments in that it may cause invasion of personal privacy to some extend.

Additionally, algorithmic bias raised many social issues in recent years, and privacy violations is heated topic among general public, sociologists, government, celebrities and so on.

There's still much more about ethical concerns regarding machine learning, such as job automation, AGI alignment, AI rights and so on, what we need to do is to find appropriate legislation for AI in these fields, and be responsible for algorithms.

2 Related work.

Professor Zhenzhong Li(4) illustrates the importance of precision,recall and f1-score in evaluation of model in his article. In our project, we write a select algorithm to decide the final

output depend on this value when different models have different results.
Pushpalata Pujari and Jyoti Bala Gupta(1) shown us the steps in feature selection. it contains screening,ranking and selection. we use similar method like lemmatization to reduce the number of features in our algorithm.

3 Dataset.

3.1 Size Information.

The original training data file is a CSV file that contains 70000 comments and their subreddit.The test data file contains 30000 comments.
After the preprocessing, the dataset usually has 74265 features. In our following training and testing process, each single word appearing in the training dataset would be represented as a feature. Instead of simply splitting the training/testing dataset by ratio 8:2, we choose k-fold (k=5) cross-validation to evaluate the performence for each model we used.

3.2 Setup and Pre-processing.

As we mentioned before, the original data is a set of comments. To analyze the dataset, we need to transform the form of data into numerical form. The main methods we used are stopword, lemmatization, count vectorization, and TF-IDF vectorization.
The main purpose of stopword and lemmatization is to filter the unnecessary word. Theoretically, this process could remove the bad feature which could confuse the classifier during classification. Besides that, filtration could also make the classification process faster.
Count vectorization encoding the vocabularies and record the term frequency on every comment and express it as a sparse matrix form. TF-IDF vectorization uses a more complicated algorithm that could add a weight on different vocabulary frequency. This method also outputs a sparse matrix.

4 Proposed Approach.

4.0 Brief Introduction.

In this project, we implemented Naive Bayes classifier and model selection method without importing sci-kit package. In addition, we compared the performences of various models, including Multinomial Bayes, Linear SVC, SGD Classifier, Decision tree, Logistic Regression, LDA, QDA, Neural Network.

4.1 Models implementations and performences. As is shown in lecture, accoring to the basic algorithm, we developed the Bernoulli Naive Bayes algorithm based on the log-odds which is similar to Linear Discriminant Analysis(LDA). The decision boundary is to see whether the probability of a sample in a class is or not greater than the probability out of the class, then compare the log-odds of different classes and classify the sample to the class which has the highest likelihood.

$$log\frac{P(y=1|x)}{P(y=0|x)} = log\frac{P(y=1)}{P(y=0)} + \sum_{j=1}^{m} log\frac{P(x_j|y=1)}{P(x_j|y=0)}$$

$$= log\frac{P(y=1)}{P(y=0)} + \sum_{j=1}^{m}(\omega_{j,0}(1-x_j)+\omega_{j,1}x_j)$$

$$= log\frac{P(y=1)}{P(y=0)} + \sum_{j=1}^{m}\omega_{j,0} + \sum_{j=1}^{m}(\omega_{j,1}-\omega_{j,0})x_j$$

Also with laplace smooth to fix the data, we implemented our own Naive Bayes model. As shown in Figure.1, compared with Naive Bayes imported from Sci-kit package, Naive Bayes implemented by ourselves turns out a higher accuracy but lower efficiency.

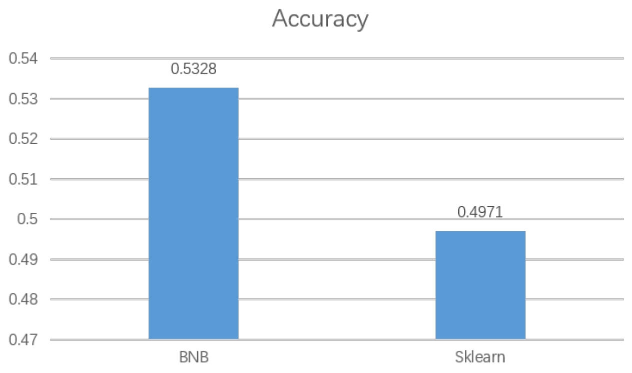

Fig. 1. Comparison between self-derived BNB and BNB imported from Sklearn

During our exploration, three models performed best: Multinomial Bayes, Linear SVC and SGD Classification, whose accuracies lied stably between 55.5% and 57.5%.Moreover, the running time for these three models are largely shorter than the running time of the others. (Data details are shown in the 'Result' section). Neural Network, LDA, QDA spend much more time to train and have lower accuracy than expected.

4.2 Designed features.

In feature engineering part, generally every single words appeared in training set represents a single feature. After tokenizing each input comment into separate words, we choose counter vectorizer and tf-idf method to transform our string type data into numerical content which can be analyzed by classifiers. Then for different models we do further constructions to modify the features. In addition, we implemented 'sentiment' method to extract sentiment values from every given comment, which returns two data: 'polarity' and 'subjectivity'. We tried to add these data as new features but it did not improve the result much.

4.3 Model selection and ensemble.

In order to further improve the accuracy, we choose the best three models(Multinomial Bayes,Linear SVC and SGD Classifier) and ensemble them to fix the final prediction.
We implemented our own model selection method. According to the classification report of each three models, we observed 'precision' 'recall' 'f1-score' for each subreddit predicted by the model. Then we choose the best prediction

(with largest liklihood to predict subreddit correctly) for each testing comment by comparing 'precision' value of models. We also tried to compare 'recall' and 'f1-score', but they had less improvement. By doing so, improved result is about 0.6% higher than the highest result among 3 models. Running time is about 0.062s. In addition, we import ensemble package and tried voting method to ensemble our 3 models. This also improved our result, but the improvement is still a bit lower than previous one and its running time is about 80s. After comparing two model selection methods, we choose the one we implemented for final prediction. More data and details are shown in 'Result' section

4.4 Other strategies.

We tried to split training/validation and test by the ratio 6:4,7:3,8:2,9:1 and 9.5:0.5, final results for ratio 6:4 is a bit lower and not stable, which may fluctuate between 56% and 58%. While the results of 9:1 and 9.5:0.5 are even more unstable, although for our training set they could acheive a higher accuracy, which we consider as the 'overfit' condition. So we choose the ratio 7:3 and 8:2 finally.
As for regularization strategies, we tried 'l1' 'l2' and 'None' when processing dataset.'l2' repularization shows the best performence, which is as expected. To optimize the result, we use grid search to seek the best value for hyperparameters of each model. For example, alpha=.22 for Multinomial Bayes, C=0.2 for Linear SVC,alpha=0.00002 for SGD Classification. More data and details are shown below in 'Result' section.

5 Results.

5.1 Multinomial Naive Bayes.

Mutinomial Naive Bayes showed a stable and high-accurate performence among all models we tested. While the accuracy may change slightly when we modified one of the hyperparameters: smoothing parameter alpha. Here is the behaviour of multinomial Naive Bayes performance against alpha, shown in Figure.2.
When we used the grid search to find the best smoothing value, we found that the accuracy of the classifier keeps increasing as adding the alpha until it reaches the peak value (about 0.5725) when a = 0.2. After that, the accuracy starts to fall as the alpha increase.

5.2 SGD Classification.

From the trials of many different models, we found that linear model behaved better in this experiment. With the property that SGD has an efficient approach to discriminative learning of linear classifiers under convex loss functions, we chose this model and make more exploration. The result is indeed as expected: SGD Classification behaved similar with LinearSVC model, it sometimes may have a slightly higher accurate. We modified its hyperparameter alpha: Constant that multiplies the regularization term to further investigate its behaviour.
As shown in Figure.3, when we use the grid search to find the best alpha value, we found that the accuracy of the classifier keeps increasing as adding the alpha until it reaches the peak

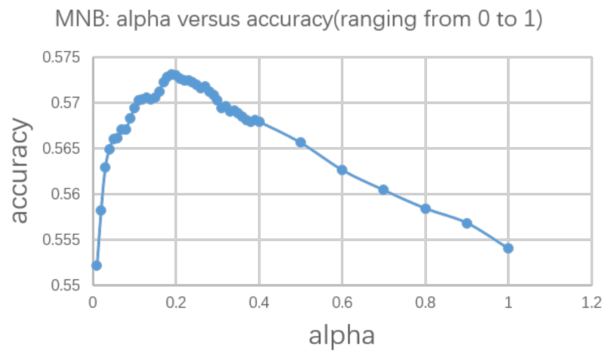

Fig. 2. The influence of alpha on MNB performance

value (about 0.5575) when $a = 2e^{-5}$. After that, the accuracy starts to fall slightly as the alpha increase.

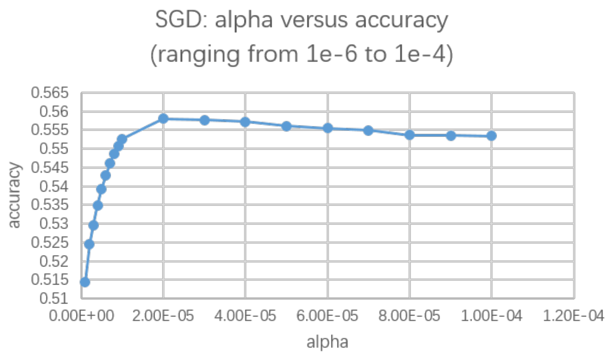

Fig. 3. The influence of alpha on SGD performance

5.3 LinearSVC.

LinearSVC is also a good choice for this experiment. As discussed in lecture, SVM is good at dealing with large number of data and multi-classes. At first we tried SVC with kernel='linear', but the performence is not as well as LinearSVC. Linearsvc is implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples (5). In order to take advantage of such flexibility, we changed its hyperparameter C: Penalty parameter of the error term and see its performance. As shown in Figure.4, we tried the value of C range from 0.001 to 10. Through the figure, we could find that the accuracy reaches the peak point(about 0.57) when C = 0.2.

5.4 Runtime accuracy comparison.

the figures show us the running time and accuracy of the different classification algorithms. In this part, we could ignore the ensemble. Most of the classifiers are based on TF-IDF vectorization (Bernoulli Naive Bayes use the count vectorization due to its algorithm needs) which give us the highest accuracy. As shown in Figure.5 and Figure.6, it is obvious that Multinomial has the best score both on time and accuracy (2.55451s and 0.57083).

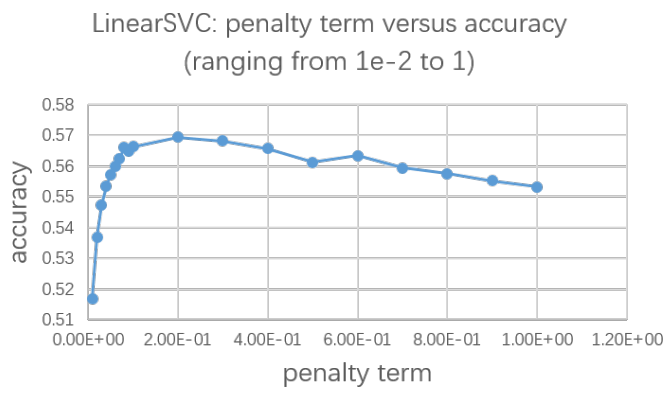

Fig. 4. The influence of penalty term on LinearSVC performance

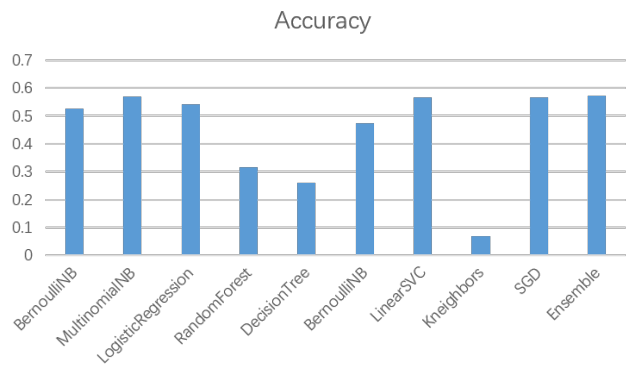

Fig. 5. Performance of different models (accuracy)

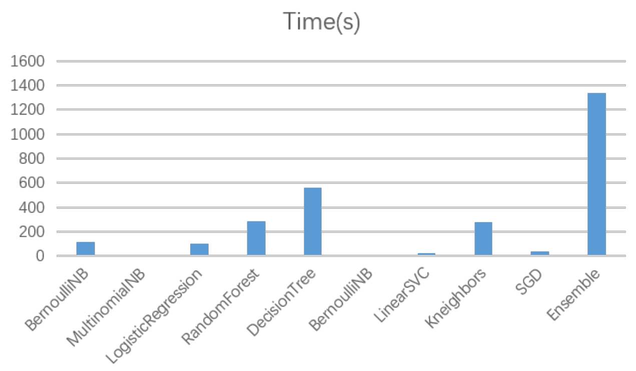

Fig. 6. Performance of different models (runtime)

5.5 Ensemble comparison.

To get higher accuracy, we use the voting method in the Ensemble package. From the last bar in figures. We could find that the ensemble could improve the accuracy at an extremely high time cost. As shown in Figure7, by optimizing the algorithm (we only use the 3 best classifiers in the ensemble), through the graph, we could find both efficiency and accuracy have been improved.

5.6 Leaderboard Accuracy.

Our final test accuracy on kaggle leaderboard is 58.733%, rank 17.

6 Discussion and Conclusion.

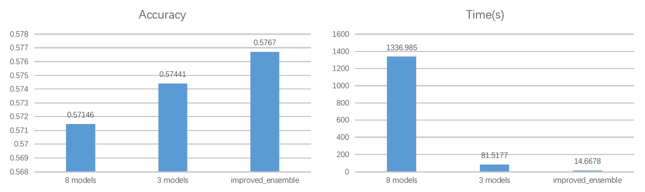

Fig. 7. Performance of different ways of ensemble

6.1 Discussion.

In this project, we test the accuracy of several classifiers, Multinimial / Bernoulli Naive Bayes, logistic regression and linear support vector machine on comments classification. We also explore how the feature of dataset construct from the original content of the comments by vectorization.
Depend on the classifier we choose, the feature engineering methods are different. For example, if we want to implement the Bernoulli naive Bayes, we only need to check if the term presents in this content and calculate the probability. we could use count vectorization, and change the value above 0 to 1. Or if we want to use logistic regression, we could use the TF-IDF vectorization to calculate the weights. This could give us a higher accuracy.
Besides the model and feature engineering, the text preprocessing is also very important, this could help us reduce the unnecessary feature at first and improve our efficiency.

6.2 Conclusion.

In this project,we met several problems. First,we compare the bernoulli , we only use a stacking voting ensemble method to improve accuracy. Stacking is used in several classification algorithms. In the future, we could construct several models with the same algorithm, and using the boosting to focus on the false part from the previous model to improve the accuracy.
The algorithm in sci-kit learn has an absolute advantage on time costing, improvement of the algorithm efficiency is important.
Then, as we mentioned before, we meet an exception with an error message of 'computation cannot be performed with standard 32-bit LAPACK' when we implement the PCA algorithm because the dataset is too large. In the future, we could find a way to compile the code in 64-bit lapack and further evaluate the influence of PCA method.

Reference.

1. Pushpalata Pujari, Jyoti Bala Gupta [*Improving Classification Accuracy by Using Feature Selection and Ensemble Model.*] ISSN: 2231-2307, Volume-2, Issue-2, May 2012
2. Chia-Hua Ho, Chih-Jen Lin. [*Large-scale Linear Support Vector Regression*]. Journal of Machine Learning Research 13 (2012) 3323-3348 Submitted 4/12; Revised 10/12; Published 11/12
3. Ozge Lule, Baris Ozkuslar,Esref Ozturk. [*Classification of User Comments into Social Media Groups with Opposing Views*].
4. Z. Li, W. Shang and M. Yan. [*News text classification model based on topic model*]. 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, 2016, pp. 1-5. doi: 10.1109/ICIS.2016.7550929
5. sklearn.svm.LinearSVC `https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html`

8 Statement of contributions.

Yichao Yang: Task 2,Task 3

Yifei Zhao: Task1,Task 2,Task 3
Weige Qian: Task 1, report writing