Machine Learning for Engineers (ECSE 511)
Mini-Project #2

Group 18

Chonghui Zhang	260955721
Jiarui Xie	260961962
Siyuan Sun	260964824

Abstract

Text classification is the process of categorizing text into organized groups. Since it is time-consuming for human-being to categorize the large subset of a textual dataset, the investigation of machine learning techniques to categorize the textual dataset is desired. The objective of this project is to build classification models that determine the categories of the input comments from a forum named Reddit. Different machine learning methods including Bernoulli Naïve Bayes classifiers and artificial neural network (ANN) classifiers are implemented, tested and analyzed on the training datasets. And the hyperparameters of the ANN model are optimized using a grid search. The final ANN model is fully connected with an activation function of relu and 45 neurons in the only hidden layer. Two types of Bernoulli Naïve Bayes classifiers are investigated and compared in terms of accuracy and runtime. Five feature selection methods are compared to find the best technique and optimized number of features for this problem. Bernoulli Naïve Bayes classifier achieves 84.9% of accuracy while the ANN model achieves 88.3% of accuracy. The work was carried out by using python on Google Colab.

1. Introduction

Text mining has been increasingly earning researchers' and entrepreneurs' attention due to its commercial potentials [1,2]. Algorithms such as spam filtering and chatbots have been deployed in many popular social media and applications. Machine learning technologies for text mining, often referred to as natural language processing (NLP), are proven to be powerful and reliable approaches to text analytics [3].

Text classification is a type of text analytics that classifies text documents. The objective of this project is to build classification models that determine the categories of the input comments from a forum named Reddit. Two types of NLP techniques are implemented: Bernoulli Naïve Bayes classifier and ANN. For the Bernoulli Naïve Bayes classifiers, Laplace smoothing is used to deal with words that do not appear in a category and reduce overfitting. Two approaches are adopted and compared in terms of accuracy and runtime. One is building a single classifier that produces 8 outputs. The other is constructing eight 1-vs-all binary classifiers. For ANN, a structure of a fully connected neural network with one hidden layer is chosen. A grid search is performed over the hidden layer size, regularization penalty and activation function to obtain the optimized hyperparameters. Finally, an ANN with a hidden layer size of 45, a regularization parameter of 0.026 and an activation function of 'relu' scored an accuracy of 88.3% and outperformed the Naïve Bayes classifier that has an accuracy of 84.9%, evaluated by 6-fold cross-validation. This report also presents the comparison between feature selection methods of text analysis, including TFIDF, information gain (IG), mutual information (MI), chi-squared stats (CHI2) and likelihood ratio (LLR).

The paper is organized as follows. Section 2 describes the preprocessing steps for preparing the feature vectors in the training dataset. Section 3 describes the proposed approach implemented in the project. Section 4 summarizes the results by using different models. Section 5 includes a discussion of the key takeaways from the project. Section 6 concludes the statement of contributions of each team member.

2. Datasets

The dataset was gathered from posts and comments from 8 subreddits from the website Reddit. And two datasets were used in this project, including the training and test datasets. In the training set, 11382 comments from 8 different subreddits are given, then the text corpus was preprocessed with removing stop words, lemmatization and vectorization, etc. The test dataset consists of 2898 comments from the 8 categories. The test dataset was used to evaluate the performance of the model built for this project.

The process of data preprocessing in this project starts from lemmatization, which converts the word to its base form [4]. For lemmatization, correct POS (Part-of-speech) tag was found to be mapped into the right input so that the function WordNetLemmatizer transferred it to the second argument for the function Lemmatize [4]. Then the words in the dataset were lemmatized with the appropriate POS tag. The next step for data preprocessing was vectorization by using the function CounterVectorizer, which encoded the word as integers for use as input to the algorithm. ConterVectorizer was an effective way to convert text to word count vector. Also, the Natural Language Toolkit (NLTK) library was used in the project since it is a useful tool for textual data processing for classification, tokenization, stemming, etc. [5]. Regularization was also used to prevent overfitting in the model.

3. Proposed Approach

Text classification in the project is to categorize textual datasets into different classes. The proposed approaches consist of feature selection, training/validation dataset split, algorithm selection, regularization and hyperparameter tuning. Bernoulli Naïve Bayes classifier and ANN are implemented for this project.

The text processing and feature selection were mainly achieved with the module, CounterVectorizer. At first, all the words from both the training and test sets are combined to one corpus for feature selection as the generalized features representing comments in Reddit are desired. Only choosing features from one dataset would arguably result in overfitting. Second, the words from the corpus were segregated and transformed to lemma forms. Meanwhile, the stop words and any lemma appeared in more than 80% of the sentences were discarded. Then, top features ordered by the feature selection techniques across the corpus were kept and vectorized. The vectorizer was then built and able to transform a text corpus into vectors of selected features. Five feature selection techniques will be used to train many Naïve Bayes classifiers with different numbers of features. The performance of each technique will be compared based on their precision, recall and F1 score. The technique with the best scores is chosen. From the training file, the dataset was divided into two portions: 80% was distributed to the training set and 20% went to the validation set. The validation set aids us in the initial algorithm selection.

The first model is a Bernoulli Naïve Bayes classifier. It is a generative classification machine learning algorithm that utilizes Bayes' theorem to estimate the likelihood of a data entry belonging to a category based on the posteriors computed from the training data [6]. Two methods can be considered as there exist 8 classes in this problem. One is building a single classifier that computes the likelihoods of the entry belonging to each of the eight class. The other is building 8 classifiers that each calculates the probability of the entry whether belonging to one class or not. Both approaches were utilized, and the results were compared to find the better one. Laplace smoothing was applied to both methods to address the features that do not appear in a class. The second model is a fully connected ANN with one hidden layer. ANN with at least one hidden layer yields a non-linear classifier, which could approximate any measurable function [7].

An ANN model has three types of layers: input layer, hidden layer and output layer [8]. In this model, the input layers size is 5000 and the output layer size is 8, as there are 5000 features and 8 classes, respectively. L2 regularization was applied to the ANN model to reduce overfitting. Some hyperparameters are shown in table 1. The other crucial hyperparameters including the hidden layer size, regularization parameter and activation function were determined by performing a grid search. A range of 10 to 300 neurons in the hidden layer and a range of 0.008 to 0.03 L2 regression penalty were searched with a spacing of 5 and 0.01, respectively. Also, two types of activation layers, 'relu' and 'tanh', were searched.

Table 1: Attributes and hyperparameters of the ANN classification model.

Attribute	Value	Attribute	Value
Hidden layer size	[10 - 300] neurons	batch size	entire dataset
L2 penalty (α)	[0.008 - 0.03]	learning rate	0.001
activation	relu, tanh	max iteration	500
optimizer	adam	Numerical stability (ε)	1E-08

Ultimately, 6-folds cross validation was used to choose the best model from the proposed models. The data was split into 6 subsets. The three proposed models were trained six times. Each time a different subset was used as the validation set, and the remaining 5 subsets were the training set. Then the model with the best predictive performance will be selected.

4. Results

The comparison between different feature selection methodologies are shown in Figure 1. For most cases, the more the number of features, the better the performance is. Also, LLR has the best performance among all feature selection techniques. Therefore, LLR will be utilized in our feature selection process to choose the first 5000 features for the construction of the final models.

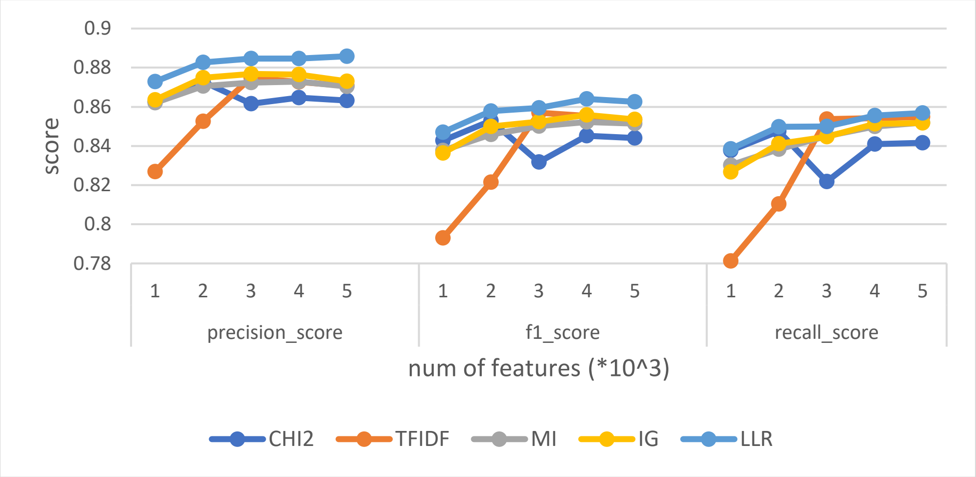


Figure 1: Comparison among the feature selection methods vs. different number of features. The scores are from Naïve Bayes classifiers built with these feature selection methods.

The grid search module in Python trains models with combinations of hyperparameters given the ranges. A 6-fold cross-validation was performed on each model. Some of the average accuracies of the models built with the combinations of hyperparameters are shown in Table 2. The grid search results indicated that the model with an activation function of relu, hidden layer size of 45 and L2 penalty of 0.026 performed the best in terms of the predictive accuracy.

Table 2: Grid search average accuracy results of ANN model over activation function, L2 penalty and hidden layer size.

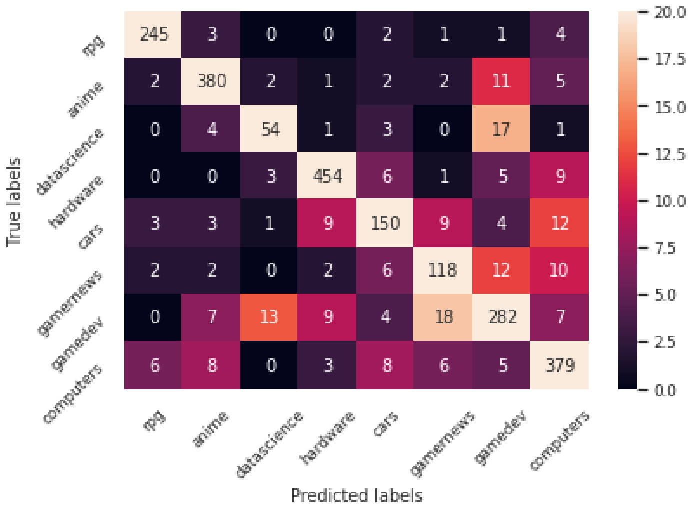
	α / layer size	35	40	45	50	55		α / layer size	35	40	45	50	55
tanh	0.022	0.878	0.878	0.881	0.878	0.879	relu	0.022	0.880	0.879	0.882	0.879	0.880
	0.024	0.877	0.878	0.880	0.878	0.878		0.024	0.880	0.880	0.882	0.879	0.880
	0.026	0.878	0.878	0.881	0.878	0.878		0.026	0.879	0.880	0.883	0.879	0.880
	0.028	0.877	0.877	0.880	0.877	0.878		0.028	0.879	0.879	0.882	0.879	0.879

To compare among the two Naïve Bayes models and the ANN model, the average predictive accuracy on the validation sets, average training time to build the model, and the average predictive runtime to generate predictions of the validation sets were considered. Table 3 shows the accuracy, training runtime and predictive runtime based on 6-fold validation. It could be seen that the ANN model offered the best predictive performance of 87.3% on average. Although it has the longest training time (14.5s on average), its predictive runtime was essentially shorter than other methods. Therefore, the ANN model was chosen as the final model for this problem.

Table 3: Accuracies and runtime of three models: (a) a single Naïve Bayes classifier for 8 classes; (b) 8 one-vs-all Naïve Bayes classifiers; and (c) ANN model selected by grid search.

	(a) Naïve Bayes Option 1			(b) Naïve Bayes Option 2			(c) ANN		
Model	Accuracy	Training time (s)	Predictive time(s)	Accuracy	Training time (s)	Predictive time(s)	Accuracy	Training time (s)	Predictive time (s)
First	86.0%	0.459	118.6	84.2%	0.452	3.92	86.1%	17.8	0.036
Second	84.4%	0.488	118	84.7%	0.454	3.87	87.5%	11.2	0.04
Third	84.2%	0.477	120.4	82.6%	0.445	3.88	87.1%	15.1	0.035
Fourth	85.0%	0.498	117.5	85.1%	0.449	3.89	87.8%	17.7	0.037
Fifth	85.4%	0.487	116.5	84.6%	0.449	3.88	88.1%	13.2	0.036
Sixth	84.5%	0.485	115.1	87.5%	0.446	3.86	87.3%	11.7	0.037
Average	84.9%	0.482	117.683	84.8%	0.449	3.883	87.3%	14.450	0.037

Confusion map is a good technique to inspect the predictive performance of a model. As shown in Figure 2, the horizontal axis is the prediction, and the vertical axis is the actual labels of those predictions. The number of correctly identified labels are in the diagonal. The numbers off the diagonal are the numbers of labels misclassified.



	rpg	anime	datascience	hardware	cars	gamernews	gamedev	computers
rpg	245	3	0	0	2	1	1	4
anime	2	380	2	1	2	2	11	5
datascience	0	4	54	1	3	0	17	1
hardware	0	0	3	454	6	1	5	9
cars	3	3	1	9	150	9	4	12
gamernews	2	2	0	2	6	118	12	10
gamedev	0	7	13	9	4	18	282	7
computers	6	8	0	3	8	6	5	379

Figure 2: Confusion map of the ANN model of one validation set.

5. Discussion and Conclusion

From Table 3, the two options of Naïve Bayes models possess the same level of accuracy, with option 1 having slightly higher stability. The training runtime for both Naïve Bayes models is short and close to each other. However, the predictive runtime of option 1 is substantially longer than option 2. It is because the implementation of option 2 can be fully vectorized, without any explicit for loop. While, option 1 would require an explicit for loop to build one classifier for eight classes together, which largely expands the occupation of the memory space and increases the runtime exponentially. The ANN model has 30 times the training runtime compared with the Naïve Bayes classifiers. This is because the structure of ANN increases the number of operations to calculate the numerous weights, activation function, backpropagation, etc. Meanwhile, the construction of the Naïve Bayes classifiers only requires the calculation of the posterior probabilities (5000+5000*8 operations). In the confusion map (Figure 2), there are many subreddits that are misclassified amongst anime, gamedev and datascience, which means that they might exhibit the same patterns based on the current features. There are many other types of subreddits misidentified as computers, especially for cars, hardware and gamenews. Also, data science has a low accuracy of 67.5% in the validation set. Thus, further investigation and improvement should be done on feature selection to add or remove features so that datascience and gamenews become more different from the other categories, and anime, gamedev and datascience can be better distinguished.

To build classification models that determine the categories of the input comments from a forum named Reddit, the team started from data preprocessing, since the cleanliness of dataset has a significant effect on the performance of the model built in this project. And the training dataset was divided into two portions: 80% was distributed to the training set and 20% went to the validation set, which aids us in the initial algorithm selection. For the algorithm selection, two models were used in the project. An ANN with a hidden layer size of 45, a regularization parameter of 0.026 and an activation function of 'relu' scored an accuracy of 88.3% and outperformed the Naïve Bayes classifier that has an accuracy of 84.9%, evaluated by 6-fold cross-validation. Therefore, the ANN model was chosen as the final model for this project. Confusion map was used to inspect the predictive performance of the ANN model as shown in Figure 2. Also, five feature selection methods were compared at the beginning of the project to find the best technique and optimized number of features in this project. The number of features have a significant effect on the features built based on the performance of TF-IDF, and the feature selection method based on the log likelihood ratio has the best performance compared to others.

6. Statement of Contributions

Jiarui Xie was responsible for code, result and discussion.

Chonghui Zhang was responsible for the comparison of feature selection methods.

Siyuan Sun was responsible for abstract, dataset and proposed approach.

References

[1] "What is text classification?," MonkeyLearn, [Online]. Available: https://monkeylearn.com/what-is-text-classification/. [Accessed 7 November 2020].

[2] D. Milward, "What is Text Mining, Text Analytics and Natural Language Processing?," Linguamatics, 14 August 2020. [Online]. Available: https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing. [Accessed 3 November 2020].

[3] M. J. Garbade, "A Simple Introduction to Natural Language Processing," 15 October 2018. [Online]. Available: https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32. [Accessed 5 November 2020].

[4] S. Prabhakaran, "Lemmatization Approaches with Examples in Python," [Online]. Available: https://www.machinelearningplus.com/nlp/lemmatization-examples-python/.

[5] "Natural Language Toolkit," NLTK 3.5 documentation, [Online]. Available: https://www.nltk.org/. [Accessed 5 November 2020].

[6] P. Joshi, Artificial intelligence with Python: build real-world artificial intelligence applications with Python to intelligently interact with the world around you. Birmingham, UK: Packt Publishing Ltd., 2017.

[7] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” Neural Networks, vol. 2, no. 5, pp. 359–366, 1989.

[8] A. Saluja, J. Xie, and K. Fayazbakhsh, “A closed-loop in-process warping detection system for fused filament fabrication using convolutional neural networks,” Journal of Manufacturing Processes, vol. 58, pp. 407–415, Aug. 2020.