Predict and Classify the Controversiality of Reddit posts and comments using Machine Learning

SeungHwan Chung
Mathematics & Computer Science department
John Jay College of Criminal Justice (CUNY)
New York, New York
seunghwan.chung@jjay.cuny.edu

Hunter Johnson
Mathematics & Computer Science department (Faculty Mentor)
John Jay College of Criminal Justice (CUNY)
New York, New York
hujohnson@jjay.cuny.edu

I. INTRODUCTION

The paper proposes six different models with six different algorithms, which allows the model to receive text-based tokenized datasets in order to predict and classify the controversiality of Reddit posts and comments. Six different algorithms will provide different accuracy scores and potentially compare the results to the customed deep learning layered model.

II. METHODOLOGY

In this work, I propose a controversial identifier system in which various models will take numerical tokenized Reddit posts and subreddit titles, compare them to the default controversial rate provided by Reddit to predict future Reddit post's controversial rate. The original plan is to compare the various regression models' accuracy scores to be compared by the customized Deep Learning model, which included the stochastic gradient descent optimization method. It will be discussed in the further steps later in the paper.

A. Dataset cleaning and pre-processing

I have received a total of twelve months' worth of Reddit posts that were collected directly through the Reddit website during the year 2016. Since the total data is equivalent to approximately 400 GB, I have used a sample generator that pulled about 10000 data sources, equivalent to 556MB, from the main dataset for the code testing purpose. Due to the equipment quality issue, I was only able to feed the data that was up to 3.2 GB for the research purpose.

To dissect the dataset, the columns in the Reddit dataset contained 18 indexes, and I have used the body and subreddit section as predictors and controversiality as a target. The body contains the comments or the posts that users desire to share their opinions publically. The subreddit represents the user's discussion topic. The controversiality section contains 0 or 1. Category 0 means that the post is not considered controversial posts or comments rated by the moderators. On the other hand, category 1 is considered posts or comments to be controversial.

Once the target and predictors are set, I have used several filters to clean the body section. First, I have created a stripUrl function that removes all the possible URL links inside the body section. Second, I have used replaced elements such as punctuations, line breaks, numbers using str.replace command. Third, I have used WordNetLemmatizer with a customized stop words dictionary to remove specific words such as 'a', 'the, and any subjects like 'I' due to reducing the total feed word counts to the model as well as unnecessary words which may possibly negatively influence the accuracy score.

Once the data has been cleaned, it requires checking whether the datasets are balanced. By checking the cleaned dataset samples, the controversial posts numbers were significantly lower compares to the ones with non-controversial posts. This may result in an imbalanced dataset that will strongly influence the accuracy scores and become unreliable. In order to make it balance, I have used the undersampling method to the non-controversial posts in order to create the balanced dataset for the models.

B. Tokenization process

- Tokenization is a method where it automatically detects the entire group of words stored inside the array from the dataset, splits them by the words, and assign individual word as a token. I have used two different tokenization methods during the tokenization phase, such as pd.categorical method and Tensorflow Keras Tokenizer commands.
- A pd.categorical method is for the subreddit section for quick converting data types to integer 16 types. The body section method is used with Keras tokenizer library, which allows to split massive phrases of comments and posts into a group of words and vectorize a text corpus by turning each text into a vector where numbers are assigned.

C. List of Algorithms

. By finishing tokenization of the entire dataset, I have set up six different models to compare and contrast which algorithms will provide the best accuracy to identify the controversial posts.

1. Logistic Regression

   Logistic Regression becomes my first pick because this algorithm performs the binary classification, with 0, 1 only, which helps whether randomly selected samples will be either controversial 1 or non-controversial 0.

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

The algorithm may change when sigmoid function is included.

2. K-Nearest Neighbors Classifier

K-Nearest Neighbors Classifiers will be my second pick. This classifier algorithm identifies all the similar algorithms in a group, captures all the distances of other close data points, and calculates whether new data from datasets are close enough and considered either controversial or non-controversial.

The major formula for the K neighbors Classifier is Euclidean distance's formula to calculate the distance of one sample data to other close sample data..

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(qi - pi)^2}$$

3. Decision Tree Classifier

Decision Tree Classifier will be my third choice. This classifier model will split the sample dataset input into a single word and store it into a tree structure. This classifier algorithm will determine the decision based on our target and test if any single words are related to either controversial or non-controversial words.

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

The formula provided above will calculate the one attribute of entropy using a frequency table. Once it finds the value of one attribute, the entropy will use the formula provided below to find the two attributes using the frequency table to find the similarity score.

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

4. Support Vector Classification (SVC)

Support Vector Classification (SVC) considered being the best algorithm to be used when a supervised learning method is conducted. The SVC creates a hyperplane that segregates the datasets into two different classes, which in this case will be either controversial or non-controversial. If the support vectors (point of the sample data) are located further from the hyperplane, those vectors are considered to be accurately placed into a class. If the vectors are close to the hyperplane, the vectors are close to being inaccurate.

5. Naïve bayes – Bernoulli

The Naive Bayes algorithm provides the fastest accuracy score compared to other algorithms. While other algorithms require all the data samples to be inputted first, then classifies the data correctly. On the other hand, the Naive Bayes algorithm predicts the classification of input data using conditional probability. The Bernoulli Naive Bayes receives the binary formed vectors in binary form, calculated by conditional probability, and classifies either 0 as non-controversial or 1 as controversial.

$$p(x) = P[X = x] = \begin{cases} q = 1 - p & x = 0 \\ p & x = 1 \end{cases}$$

The above formula is representing Bernoulli distribution

6. Naïve bayes - Multinomial

As Naive Bayes explained in sub-section 5 from Bernoulli, Multinomial counts the frequent words from the input sample dataset and classified based on how frequent words are used based on the target.

III. Results

The result section will cover each algorithms' performance, confusion matrix for the highest accuracy scored algorithm, and a classification report for the highest accuracy scored algorithm.

TABLE I. ACCURACY OF EACH ALGORITHM PERFORMANCE

ALGORITHM	ACCURACY
LOGISTIC REGRESSION	0.689
K NEIGHBORS CLASSIFER	0.555
DECISION TREE CLASSIFIER	0.621
SVM	0.540
BERNOULLI NAÏVE BAYES CLASSIFIER	0.683
MULTINOMIAL NAÏVE BAYES CLASSIFIER	0.672

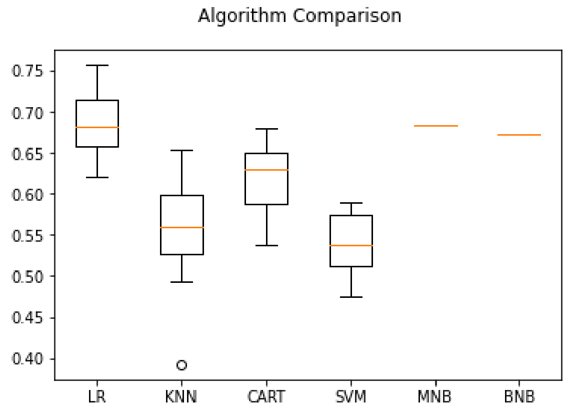

Fig. 1. Algorithm Comparison Box plot

TABLE II. CONFUSION MATRIX

		True Labels	
		Positive	Negative
Predicted Labels	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN

This matrix provides various evaluation methods:

- precision = $\frac{TP}{TP+FP}$

  The precision provides the accuracy of positive predicted cases.

- Recall = $\frac{TP}{TP+FN}$

  The recall provides the accuracy of positive cases that were predicted.

- Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}$

  The accuracy provides the ratio of the correct prediction. The higher percentile of accuracy represents the accuracy of the prediction.

TABLE III. LOGISTIC REGRESSION CONFUSION MATRIX

Labels	Precision	Recall
0	0.72	0.62
1	0.68	0.77
Accuracy	0.700	

IV. CONCLUSION & MISTAKES & FUTURE WORK

The paper proposes various methods to conduct properly classify and predict the controversiality of Reddit posts and comments using various methods. The methodology section explains the list of different algorithms that can be used to classify the randomly selected datasets into two classes as known as controversial or non-controversial, using different algorithms. Based on the result, as shown in table 1, Logistic Regression provided the highest accuracy among six different algorithms, which resulted in 70 percent accuracy. The second best algorithm is the Naive Bayes algorithm with the Bernoulli classifier, which resulted in 68 percent accuracy. The third best algorithm is the Naive Bayes with Multinomial classifier, which resulted in 67 percent accuracy. Based on the Confusion Matrix on Logistic Regression, as shown in table 3, label 0, as known as non-controversial, provided 72 percent precision and 62 percent recall. Additionally, label 1, as known as controversial, provided 68 percent precision and 77 percent recall. Therefore, Logistic Regression seems very promising when classifying and predicting the controversiality of Reddit posts or comments.

For Mistakes and Future work, this project has so many possibilities that can be tweaked into the different prediction analyses. The development of six different models with various algorithms wasn't based on solid knowledge. The duration of the research, the knowledge regarding machine learning and deep learning, and the quality of equipment were limited. As a theoretical approach, all of the models' accuracies can be significantly boosted in several ways:

1. Better hyperparameters of the algorithms.
2. Organization of tokenized vectors (dataset quality).
3. Customized deep learning layers.

All the algorithms that were provided in this research were used default hyperparameters. If additional customed hyperparameters such as stochastic gradient descent optimizers or customed momentums, learning rates, alpha, or other features were included, the accuracy might possibly increase. While researching the methods to increase the accuracy score, a clean dataset provides the high quality of input data towards the models, which leads to the high quality of the trained model with better accuracy. If the dataset has been well cleaned during the preprocessing phase, the accuracy score could have increased. Lastly, Deep learning customed layers could have generated better accuracy with proper layers to dissect the dataset. The highest accuracy score only reached 70 percent because the default classification algorithms weren't detailed enough to classify the datasets properly. The original purpose of this research was to compare and contrast the default classifier algorithms with a deep layer model with three different stochastic gradient descent optimizers such as SGD, SGD with Nesterov, and Adam. Due to the limited knowledge with deep learning layers, the plan for the deep learning architecture could have been constructed with max pooling, ReLU, and LSTM layers. Max pooling layer provides downscaling the size of the dataset, which allows the model to dissect the dataset much easier. To increase the efficiency with the data transportation speed, Long short-term memory (LSTM) will provide efficient memory management, and lastly, ReLU or Logistic Sigmoid layers with optimizers will provide training for the model. This is just an example that may construct the small prototype of a neural network, deep learning purpose, and higher accuracy compared to other default classification algorithms.

In future work, this current classifying and predicting models based on text-based tokenized datasets can be applicable to various industries. Although it only predicts whether the user's posts or comments are considered to be controversial or non-controversial, multiple features can be attached to the current model just like a Lego block, such as identifying the user's political preferences. It can also be developed to predict and prevent future fake news or comments that can raise chaotic situations in the internet community.

REFERENCES

[1] Analytics Vidhya Team, *Siimple Guide to Logistic Regression in R and Python*, Analytics Vidhya, Nov. 1, 2015. Accessed on: May. 25, 2021. Available: https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression-in-r/

[2] A. Agarwal, *Support Vector Machine – Formulation and Derivation*, Sep. 24, 2019. Accessed on May 26, 2021. Available: https://libraryguides.vu.edu.au/ieeereferencing/webbaseddocument

[3] E. Szczerbicki, *Management of Complexity and Information Flow*. Newcastle, AustraliaL Elsevier Science Ltd, 2001, pp. 247-263

[4] I. Jose, KNN (K-Nearest Neighbors) #1, towardsdatascience, Nov 8, 2018. Accessed on May. 25, 2021. Available: https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d

[5] S. Narkhede, *understanding Confusion Matrix*, May. 9, 2018. Accessed on May 26, 2021. Available: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[6] C.L. Wu, S.L Shin, *Machine Learning based classification for Sentimental analysis of IMDb reviews*