**Project Progress Report**

Machine Learning Solutions to Online Twitter Bot Detection: Implementation, Comparison, and Improvement

**Team #:** 3

**Team members:**

Stephen Ling, Dept. Undergraduate

Wyatt Meng, Dept. Undergraduate

Bryce Chen, Dept. Undergraduate

**Date of submission:** Nov, 18, 2022

**ABSTRACT**

In this Nature Language Processing project, we aim to evaluate the performances of bot detection algorithms based on a dataset with English Tweets and true labels. Text data preprocessing methods tokenization, stop words and punctuation marks removal, and stemming are performed before text data vectorization through Bag-of-words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) in the feature extraction stage. From 10 classic models we realized, cross-validated, and analyzed, Supporting Vector Machine with Stochastic Gradient Descent (SVMwSGD) gives the best result, i.e. high TPR and robustness. We generated visual representations, i.e. Word Cloud and histogram, for showing text content and model performances. Looking ahead, we’ll focus on the implementation of modern deep learning models like Bert and we expect it to outperform in terms of accuracy and simplicity.

**2. Introduction**

Artificial intelligence (AI) techniques pervaded increasingly into the information ecosystem - in the form of virtual agents, chatbots, social bots, etc. (Guzman & Lewis, 2020).

Social bots are agents that deliver information in the media intending to influence online discussions, news attention, and even public opinion through speech. Given the rate of advancement in natural language and deep learning techniques, social bots will become intelligent, so it is crucial to identify them and react accordingly. The significance of this Nature Language Processing classification task spans from achieving bot detection to algorithm optimization.

Nature Language Processing classification tasks generally involve 4 stages: *1. Preprocessing, 2. Vectorization, 3. Model Fitting, 4. Result Evaluation*. In this section, we will discuss commonly used techniques at each stage with a focus on social bots’ detection algorithms.

2.1 Preprocessing

There are applicable techniques for the preprocessing stage. Tokenization is used to break the whole text into small chunks to make it easier to assign meaning (Solangi et al., 2018). Also, Stemming directly converts words into root form to reduce inflectional forms of words in the text. The commonly used models are Affix Removal Stemming, N-gram Stemming, Table Lookup Stemming, etc. In comparison, Lemmatization converts various inflected forms of words into meaningful forms based on the consideration of context (Asghar et al., 2014). Besides, since punctuation marks could not provide any information for analysis, they are removed through techniques named Punctuation Marks Removal (Etaiwi & Naymat, 2017). Finally, some other fancy preprocessing techniques like Part of Speech (POS) tagging are used by researchers to classify words into specific morphological categories (Asghar et al., 2014).

2.2 Vectorization

The vectorization step extracts vectors from the text so that models could use extracted vectors for training and classification. Vectorization could be done by statistical approaches and deep learning. The traditional vectorization methods commonly used are Term Frequency-Inverse Document Frequency (TF-IDF), Chi-square clustering method, and Latent semantic analysis (LSA) (Liang et al., 2017). Moreover, some neural network-based models like Word2Vec and Doc2Vec are also used for vectorization (Singh & Shash, 2019). For the detection of social bots topics, some researchers use Global Vectors (Glove) and Embedding from the language model (ELMO) for vectorization (Heidari, Jones, & Uzuner, 2020). Also, some researchers use Bidirectional Encoder Representations from Transformers (BERT), commonly used for sentiment feature extraction (Heidari & Jones, 2020).

2.3 Model Fitting

Based on the literature, the social network of users, profiles of users, account usage, and Twitter content are commonly used by researchers studying the detection of social bots. In most relevant research, researchers tend to use content, users’ profiles, and account usage to predict whether the account is a social bot (Rodríguez-Ruiz et al., 2020). However, some researchers only focus on the content and after preprocessing the texts, they use a Recurrent Neural Network (RNN) model with word embeddings to detect bots without using other features like users’ profiles or social networks (Wei & Nguyen, 2019). Moreover, some researchers introduce new features like the sentiment of Twitter to analyze whether the content is from humans or bots (Dickerson, Kagan, & Subrahmanian, 2014).

Other than commonly used features for the detection of bots' Tweets, the widely used machine learning methods for the detection of social bots on Twitter include both neural network and non-neural network models: Naive Bayes (NBC), Support Vector Machine (SVM), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long-Term Short-Term Memory (LSTM), etc. (Alothali et al., 2018). Also, if there is no human-bot indicator label for the given text, unsupervised learning methods, especially clustering methods like DenStream, StreamKM++, etc., are used (Khan et al., 2016).

2.4 Performance Evaluation

Finally, since the detection of social bots on Twitter is a classification problem, the Confusion Matrix with Precision, Recall, Accuracy, F-measure calculated, ROC, and AUC are commonly used to evaluate the performance of models. Moreover, some other measurement techniques like Random Walk, Counting Credits at vertex, etc. are also used for the purpose of model performance measurement (Alothali et al., 2018).

**3. Methodology**

3.1 Introduction to Data

We contacted the first author of “[Algorithmic Agents in the Hybrid Media System: Social Bots, Selective Amplification, and Partisan News about COVID-19](https://academic.oup.com/hcr/article/48/3/516/6587151)” and were granted permission to use their dataset for this class project (Duan et al., 2022). The dataset contains English tweets about a specific topic, the Covid-19 pandemic. These tweets were collected from Twitter from March 1, 2020, to May 31, 2020 (under the pandemic outbreak) by matching up with a list of 181 keywords about Covid-19. To reduce the computational complexity and time, we randomly partitioned the dataset and used the first 120,000 rows and eight columns of the original one. In general, the quality of this dataset is high, i.e., there are no single missing values nor NAs in the content and handle columns. The feature columns we plan to use will be: 1) `content`: Twitter content sent by users/bots; 2) `handle`: represents the Twitter account's ID or nickname.

The second last column among the dataset, SourceatBot7, is considered the true label, which is a binary classification outcome indicating the Twitter is sent by human or bot.

The previous research on this data has 3 focuses: statistical comparison between humans and bots, the prevalent Twitter topic for bots, and time series analysis. Duan et al. conclude that human Twitter users tended to post slightly more original content and have more comments than bots, and Twitter bots mainly focused on political or societal topics based on the results of the Structural Topic Model (STM) (2022).

3.2 Algorithm, Program, and Platform

This project uses Google Colab and Jupyter Notebook as a platform and working environment. Codes are uploaded to both GitHub repository and Google Drive. Up to the completion of this report, we used Python packages from Numpy & Pandas (data manipulation), scikit-learn (vectorization, machine learning models, confusion matrix), Matplotlib (plotting), Nltk (tokenization, stopwords removal, stemming).

The data preprocessing methods are tokenization, which converts a tweet into a vector of words, removing stop words and punctuations, and stemming, which reduces a word to its word stem.

The feature extraction methods are Bag-of-words, which is a representation of text that describes the occurrence of words within one tweet, and TF-IDF, which describes how common or rare a word is in the entire document set (the closer it is to 0, the more common a word is).

We experimented on 10 machine learning models: Bernoulli Naive Bayes (BNB), Multinomial Naive Bayes (MNB), k-Nearest Neighbors (kNN), Nearest Centroid (NC), Ridge Regression (RC), Perceptron (P), Passive Aggressive Classifier (PAC), Logistic Regression (LR), Support Vector Machine(SVM), Supporting Vector Machine with Stochastic Gradient Descent (SGD). Among 10 machine learning models, the Naive Bayes with Bag-of-words feature extractor is selected as the baseline model. According to <https://journals.sagepub.com/doi/full/10.1177/0165551516677946>

The BNB classifier model has an equivalent to the Bayesian counterpart and has been applied in many text data projects as the baseline model. Bag-of-words model has also been adopted by many studies as the baseline due to its simple intuition and implementation.

3.3 Conduct Experiment

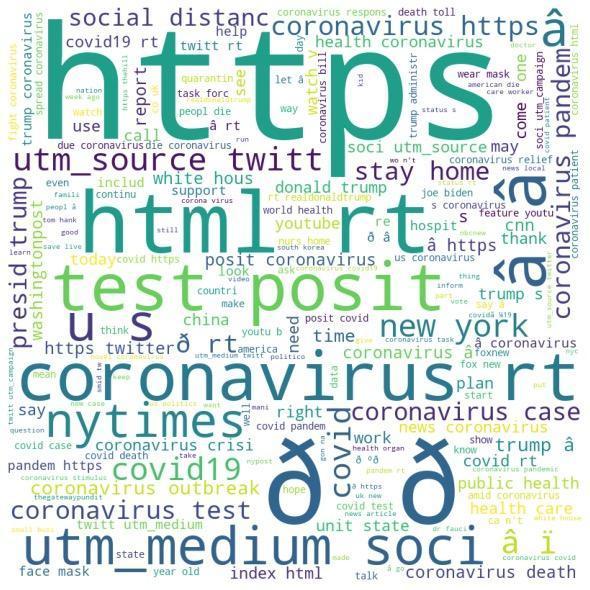
With the given data, the non-English language tweets are removed from the data since we would only deal with the English tweets in this project. After tokenization, each tweet goes through the function that removes the stop words and punctuation marks, and each word is reduced to its word stem by stemming. Then, the Bag-of-Words and TF-IDF extract a feature matrix with shape of 120,000 \* 154,011 (observations \* vector length). Before fitting the classification models, the data is splitted into 80% train and 20% test with random seed = 44. Then, the sparse matrix of training data is piped into 10 machine learning models and related confusion matrices/accuracy scores are generated. Finally, the histogram is plotted to provide a comprehensive comparison between different feature extraction or classification methods.

3.4 Results Evaluation

We anticipate correctly applying AI models like machine learning and neural network models in the context of natural language processing. For evaluation, the confusion matrix will be created, and Precision, Recall, F1, and Accuracy are calculated to evaluate and compare the models’ performance, conditional on the SourceatBot7 column being the ground truth. Successfully answering the following questions would be a successful project: 1) Whether extracting sentiments could contribute to detecting Tweets from Bots? 2) What feature extraction and classification methods would lead to higher prediction accuracy of tweets from bots.

**4. Results**

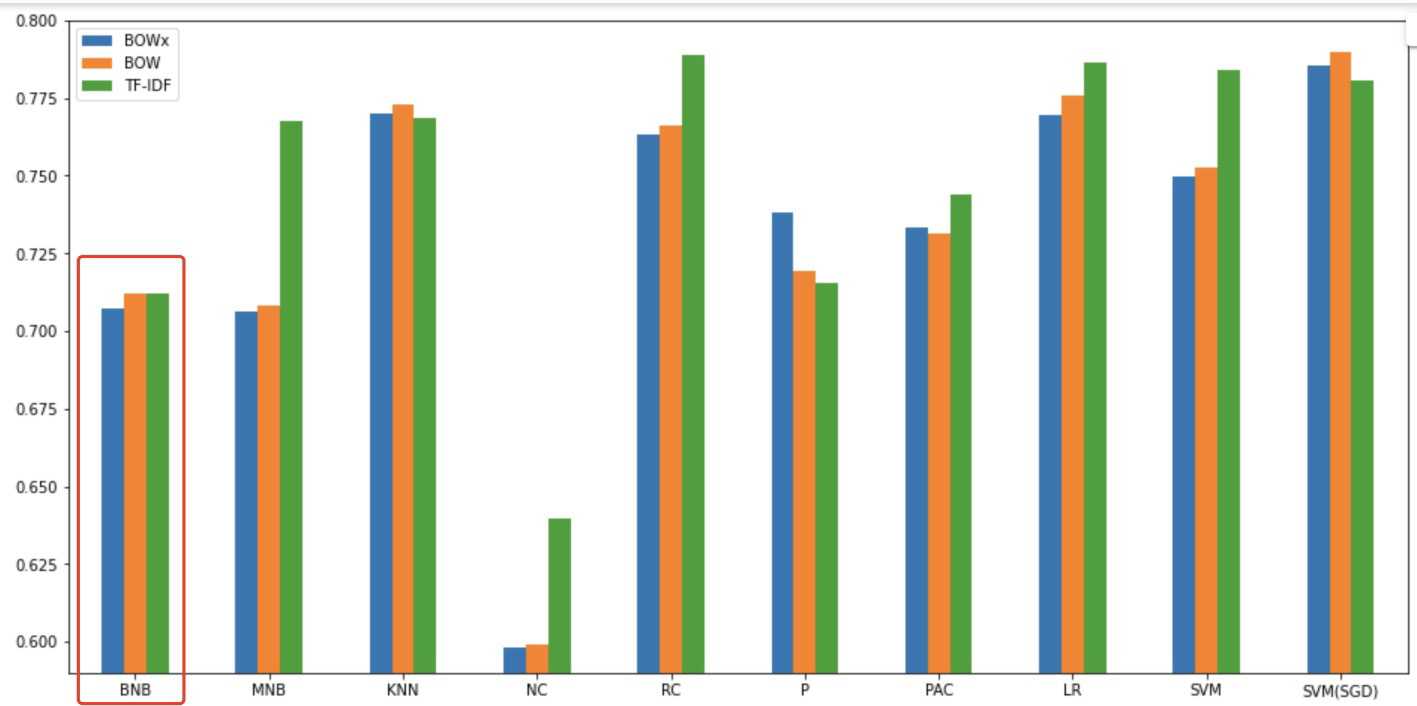
4.1 Word Cloud

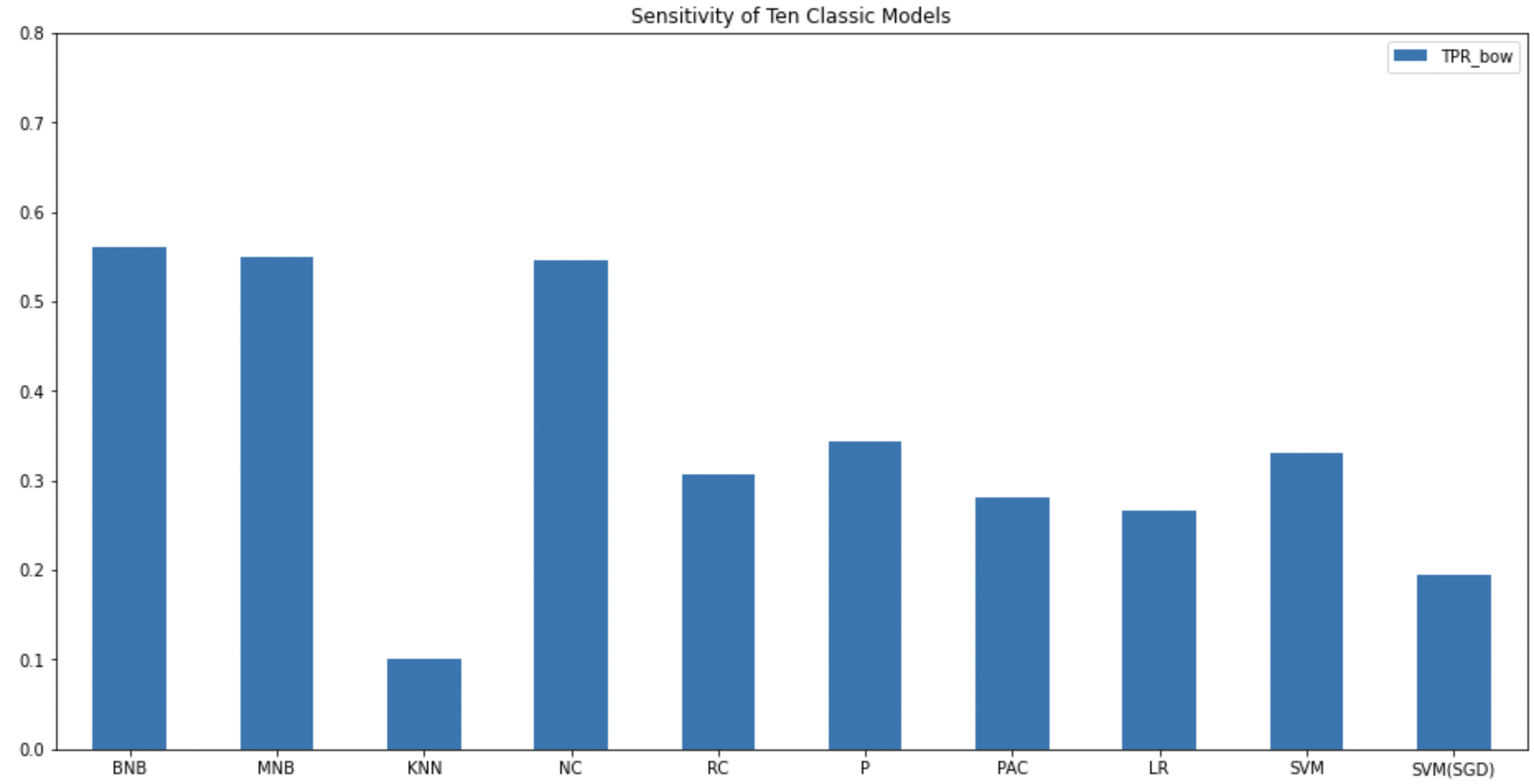
fig 1

We first generate a word cloud for the pre-processed text, fig 1. The font size of words in the image is proportional to the frequency of the words in the dataset. According to the word cloud, Internet and covid-related subject terms appear most frequently in our dataset, i.e. test posit, http, html, and coronavirus.

We implemented 10 model: Bernoulli Naive Bayes (BNB), Multinomial Naive Bayes (MNB), k-Nearest Neighbors (kNN), Nearest Centroid (NC), Ridge Regression (RC), Perceptron (P), Passive Aggressive Classifier (PAC), Logistic Regression (LR), Support Vector Machine(SVM), Supporting Vector Machine with Stochastic Gradient Descent (SGD). We chose bar charts to present the results of 10 models using two vectorization methods, Bag-of-words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF).

Ridge Classification (RC) and Supporting Vector Machine with Stochastic Gradient Descent (SVMSGD) give the best accuracy, over 80%, and Logistic Regression (LR) and SVM follow, achieving about 78%. The Nearest Centroid (NC) is the least powerful one and its accuracies under 2 vectorization methods are relatively low, less than 65%. We implemented two vectorization methods, and the results show that TF-IDF significantly boosts the accuracy of Multinomial Naive Bayes (MNB), RC, and SVM by 10%-5% compared to BoW. While this rule not robust since Perceptron (P) classifier, adding TF-IDF decreased the accuracy comparing to without BoW and with BoW by 2%.





1) calculate some critical values like TP/TF/… for the confusion matrix of the baseline model (BNB).

We calculated confusion matrices and TPR to compare the performances of our algorithms to the baseline BNB one. For our BNB baseline model, the probability of a bot test, conditioned on truly being a bot (TPR) is 0.56, which indicates that there is 56 % that any tweet that has been written by a bot is likely to be classified as bot by the test. The sensitivity graph shows that the BNB, NC, and NBC have the best performance, about 0.55. KNN has the worst performance in predicting the bot.

5. Discussions

5.1 Interpretation of Results

We performed two different vectorization methods and ten bot non-neural network detection algorithms to deal with this text classification problem at this stage. There is a noticeable gap between accuracies and the TPR: the accuracies are much higher than the sensitivities, meaning that some models are not powerful to perform bot detection. Or there is information loss in the vectorization stage.

Looking ahead, we will improve our results in two directions. First, we will introduce more advanced preprocessing methodologies(先加上一两个看到的比较fancy的)。 Second, we will attempt to use neural network models like transformer models, like Bert, to approach this problem. (加上一两句别人做的比较好的结果)。 Based on this, we expect a surge in TPR with technologies discussed above.

We will attempt to give a theoretical grasp or at least intuitions of these methods and results. We will relate to CS539’s contents and explain with clear reasoning.

5.2 Discussion of Objectives

Till the completion of this progress report, we have tested 2 feature extractors mentioned in the proposal, and constructed the baseline model (Naive Bayes) and compared the performance of other non-Neural Network models with the performance of the baseline model. However, even though some models like SVM reach an accuracy of 0.8, in general, the performance of models does not meet our expectations. In the next step, we will introduce Neural Network based feature extraction methods like Bidirectional Encoder Representations from Transformer (BERT) and classification methods like Convolutional Neural Networks (CNNs). Moreover, a more comprehensive comparison with calculation of TP, TN, FP, FN, etc. would also be constructed to compare feature extraction and classification models/methods.

Moreover, we believe the tweets from Bot have a different emotional expression compared with the tweets from humans, so sentiment analysis would also be useful to classify the origin of tweets. Specifically, we would use a pre-trained sentiment analyzer in the nltk package to generate a sentiment rate, which could serve as a new feature for classification.

6. References