And Now, The Fake News:

Comparing The Efficacy of Four Classifiers in Determining Whether or Not A Given Article Is Real or Fake News

Zach Quinn

Bellevue University

**Introduction**

**The Growing Threat of Misinformation**

Identifying and deploying a comprehensive, reliable and consistent machine learning approach for the explicit purpose of fake news detection has become increasingly necessary as a means of deterring and combating the spread of misinformation. Though what is and is not objectively true may seem intuitive, individuals, organizations and entire populations are routinely deceived due to their inability to discern fact from fiction. According to Pew Research, 64% of Americans, nearly two thirds of the country, admit that they believe the prevalence of misinformation has skewed their perception of verifiably accurate information (Barthel, Mitchell, Holcomb, 2016). The consequences of consuming information from unreliable sources and falling prey to misinformation can be wide-ranging, destructive and even deadly. One of the most famous purveyors of fake news, political consulting firm Cambridge Analytica, exploited the inability of individuals and machines to consistently identify fake news content by spending millions of dollars on incendiary multimedia campaigns designed to sway elections, exacerbate existing ethnic tensions and even fuel physical violence (Wiley, 2019).

Cambridge Analytica’s actions demonstrated that fake news does not just represent an abstract threat. Instead, misinformation dissemination is now a tool of psychological warfare, the field’s equivalent of the atomic bomb. Political consultancy firms are not the only entities experimenting with how information access impacts individuals’ psychological states. In 2014 Facebook revealed that its developers had changed the content of news stories displayed to over half a million randomly selected users to chart their reactions to certain kinds of media (Wiley, 2019). Fake news detection is especially pressing in the context of the current pandemic. Research suggests that users are more likely to spread misinformation in terms of crises and there is a significant correlation with the spread of false information during global health crises such as COVID-19, Ebola, H1N1 and stretching as far back as the flu of 1918 (Fleming, 2020).

**The Challenge of Classification**

The volume and subtlety of fake news items poses a daunting challenge for even the most hyper-tuned machine learning algorithms. First, for a classifier to be truly comprehensive, it would need to be able to consistently discern falsity from truth across many of the world’s most widely spoken languages. However, the variety and scope of human language means that while some languages may be phonetically correct, they are not always syntactically identical. While English contains gendered pronouns that account for singularity and plurality, other languages like Turkish, do not account for gender in pronoun usage. Additionally, Chinese and Russian do not use the Latin alphabet, meaning they have an entirely different array of symbols and meanings than English letters.

Other languages, like Korean, can construct sentences multiple ways and remain syntactically correct, meaning that any classification effort applied would be largely insignificant (Seo et al., 2019). One of the most famous examples within the data science community of a classification project gone wrong was Twitter user Lupin’s attempt to classify a sentence that, due to his native German’s grammar, yielded an output of “The economics of economics (including economics, economics, economics, economics, economics, economics) is a part of economics” (Lupin, 2017). The tweet from Lupin emphasizes that news detection presents a multi-faceted natural language processing challenge that requires a dimensional, nuanced approach, especially when applying word vectors between one or more spoken or written languages.

**The Project**

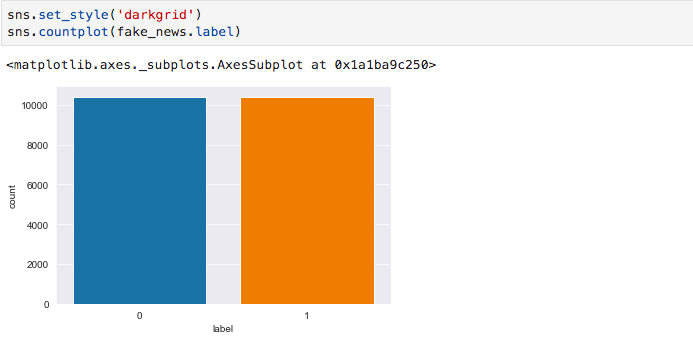
**Focus and Hypothesis**

One of the best defenses against misinformation is a classification algorithm, particularly those that rely upon natural language processing. The algorithms within this category include Naïve Bayes, support vector machines, and logistic regression. Building and deploying these classifiers is not super labor intensive, they do not take as long to execute as deep learning neural network architectures and can be applicable to a variety of text data including news stories and even Tweets. On the other hand, since these methodologies are being used as text classifiers there is a heavy focus on word association and not overall sentence structure. Also, they cannot detect tonal differences like humor and sarcasm.

The goal of this project is to assess parameters of textual data, determine an appropriate method to split and vectorize individual words and features from a string input, and construct an ML model, which uses natural language processing to make a binary determination of whether or not an item is real or fake based on the number of times it appears in the input document. The research question to answer is: What kind of natural language processing classifier is ideal for fake news detection? However, given the scope of the classification problem, after running an initial test using a passive aggressive classifier, it became apparent that more insight would be gained from fitting the model to other classifiers, constructing a pipeline, and generating and comparing the resultant classification reports as a comprehensive metric. Consequently, the focus of the research shifted from simply building and assessing an NLP classifier, to assessing the NLP classifier best suited for the task of fake news detection. The hypothesis of this experiment is: Given its ubiquity and range of application for classification problems, Naïve Bayes will be the best classifier to employ for determining whether or not the information contained in a news story is objectively true or false. The output would be charted in a classification report and a correlation matrix that contains the binary labels 0 and 1 for true and false respectively.

**Data**

The data examined in this project originated on open source data set repository Kaggle. After converting the CSV file to a data frame, it became apparent there was five columns: An index ID, title, author, text and binary label. Since the data was already relatively clean, the index did not have to be reset and the columns did not have to be renamed. The title column included a brief headline for the story as it appeared online, the text contained concise digitally published news stories of about 200 words and the binary label was an integer value of 0 or 1. The data contained 20,800 values prior to cleaning.



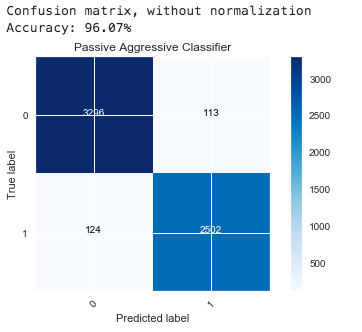
Additionally, before removing NA/NaN values, the distribution of true to false stories was relatively equitable and nearly evenly distributed. However, upon cleaning, the values dropped to 18,285. Notably, the distribution of the data shifted, with the clean data set yielding more true stories than false stories.



Since the project is a classifier trained by comparative analysis, the focus was on the features in the text column rather than title column. The target variable was the label column, which determined whether or not an item was real or fake. Since there was no consistency in authorship, the author column was largely insignificant to this analysis.

**Methodology**

To build the model the split parameter chosen was .33, constituting a slightly higher split than a conventional 75 to 25% split of testing and training data. The resultant training and test sets were transformed and stored in vecotrizers, specifically a “bag-of-words” count vectorizer and a term frequency-inverse document frequency (TF-IDF) vectorizer. Stop words were chosen to ensure the model does not use insignificant word values such as articles like ‘a, an, the.’ The stop words chosen were the default English parameters. Before feeding the model into a Naïve Bayesian classifier, a passive aggressive classifier was fit and executed to gain insight into initial testing metrics. In total, the test was repeated with four classifiers, Passive Aggressive Classifier, Naïve Bayes, Support Vector Machine and logistic regression. Consequently, the passive aggressive classifier yielded a nearly perfect accuracy rate without normalization.

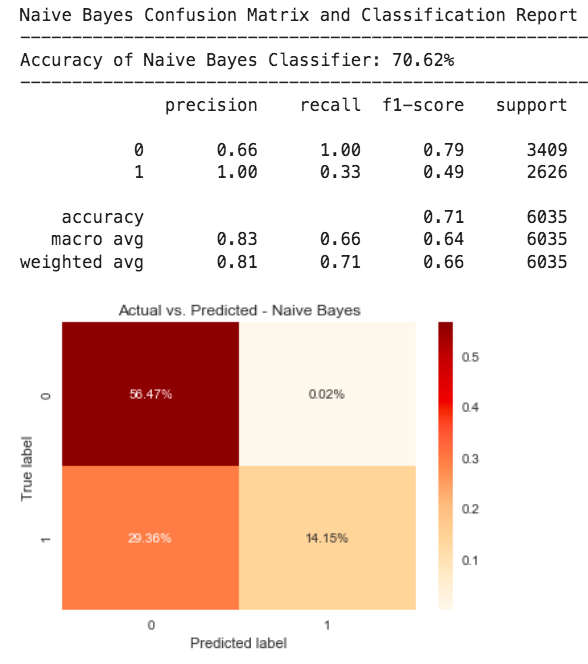


However, it must be noted that accuracy can be a misleading metric that does not necessarily offer depth insight into the performance of a model.

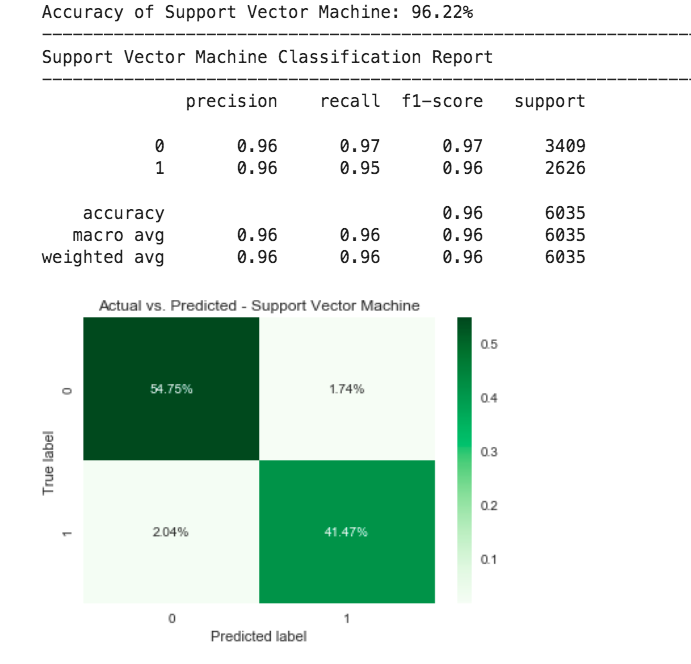
After conducting research, it became clear that accuracy is not necessarily the most reliable metric to assess the performance of a model. Accuracy is a particularly arbitrary metric. Therefore, for the next tests, classification pipelines were constructed using SK Learn’s pipeline function. To evaluate these tests, a confusion matrix was created and executed to provide an array of values assessed. Additionally, a classification report was generated containing precision, recall, F-1 score, support, accuracy a macro average and a weighted average to provide depth insight into the performance of the model. The statistical tests focused narrowly on recall and precision because both measurements provide specific insights about how elements among the class are correctly identified. The F-1 score and support were the descriptive test statistics because F-1 score conveys to analysts the harmonic mean between precision and recall and support only indicates how well distributed the overall data set is.

**Results**

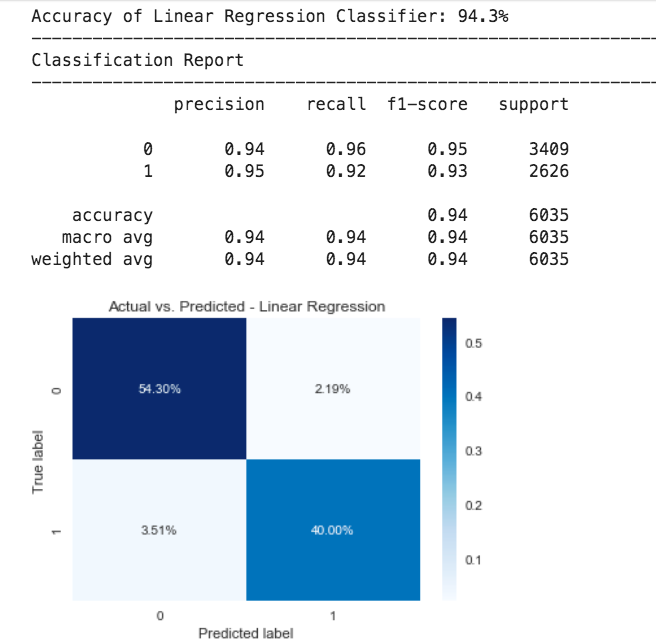
In the process of my experiment, I actually ended up proving my null hypothesis because Naïve-Bayes turned out to perform the worst across the board. Despite receiving stellar scores of 1.0 in recall and precision, Naïve Bayes proved to be only seventy percent accurate.



Conversely, the support vector machine with a linear kernel pipeline performed the best out of all four assessed models with an accuracy rate of 96% and precision, recall and F-1 scores in the high 90s.



The linear regression classifier also performed consistently well, yielding an overall accuracy rate of 94.3% and strong precision, recall and F-1 scores as depicted in the confusion matrix embedded below.



Ultimately, I proved my null hypothesis although with an important caveat. Though, at first glance, it may appear that Naive Bayes received the lowest accuracy score, as I mentioned above, accuracy is not the best metric to assess a model's performance. It must be noted that even though Naive Bayes was only 70% accurate, its recall and precision scores were excellent, meaning that it could correctly classify values in either class. However, the most consistently high-performing model was the linear kernel Support Vector Machine that scored in the high 90s across all three categories: Precision, recall and f1-score. It is also worth noting that the performance of these classifiers could still vary greatly depending upon the stop words chosen, features filtered and hyper parameters entered. There is also little guarantee these methods would be effective on non-English language classification problems. However, this experiment determined that a Support Vector Machine is the most effective classifier for addressing an English language fake news detection problem.

**Discussion and Conclusion**

Though the project appears to have run relatively smoothly, it is worth noting a few considerations that may be helpful to any individual who attempts future replication of this experiment. First, after receiving notes on my presentation, I realized that I unintentionally data snooped, or used my test set more than once in the process of model evaluation, meaning that the model has already ‘seen’ the test set before and, therefore, cannot provide an accurate representation on how well it would perform on a blind trial of a test data set. Since I already conducted the experiment and wrote my paper before receiving my notes, I had not intended to revise the project itself at this point. Additionally, I made the decision to not change the original experiment because, as a student, it is important to be transparent and self-aware of miscalculations such as data snooping that could result in a biased assessment of a model. I am researching the concept of data snooping and taking supplemental e-courses between Bellevue terms so I am able to better rationalize, construct and evaluate models in the future.

Since I used a data set on Kaggle, a site that incentivizes users to provide clean data sets, I did not have to perform extensive cleaning on my data frame. Next, since the data was so clean I could get away with not tuning my hyper parameters to the extent of specifying particular stop words and filtering for meaningless strings like date lines and subheadings. In building my TF-IDF vectorizer I, again, used a default max\_iteration value of 50. Being a largely text-based problem, I was fairly limited on the visualizations I was able to generate aside from frequency bar plots and confusion matrices.

Finally, and I have to stress this, I can only assume whoever put this together was one hundred percent correct in their initial classification of a story as true or false. As I explained in the beginning of this paper, fake news is getting harder to spot. It is entirely possible there are misclassified articles within this data set that would invalidate these results. However, since there are over 20,000 200-word stories, I have to trust that the creator possesses a journalistic background that allows him or her to make such decisions.

**References**

Barthel, M., Mitchell, A. & Holcomb, J. (2016). Many Americans Believe Fake News Is Sowing Confusion. *Pew Research Center: Journalism and Media*.

Fleming, N. (2020). Coronavirus misinformation and how scientists can help to fight it. *Nature Research Journal*. Retrieved 2 August 2020 from: <https://www.nature.com/articles/d41586-020-01834-3>

Lupin. (2017). ‘God bless the German language.’ *Twitter*. [Tweet].

Wiley, C. (2019). *Mindf\*ck. Cambridge Analytica and the Plot to Break America*. Random House*:* New York.