



California State University East Bay

FAKE NEWS DETECTION USING NLP AND MACHINE LEARNING

by

Sai Sruthi Bodapati (FS4443)

Supervised by

Professor Jiming Wu

Co-Director of M.S. Business Analytics

OBJECTIVE

The objective is to build a model by using Python Natural Language Processing (NLTK tools) and Machine Learning techniques (Sklearn kit) to perform basic natural language processing and machine learning methods to train the dataset to build a classifier to identify fake news from real ones.

INTRODUCTION

With the advent of social media and the pace at which information travels these days, fake news has become one of the most challenging issues. The reasons behind fake news include media manipulation and propaganda, political and social influence, provocation and social unrest and financial profit. However, people and groups with potentially malicious agendas have been known to initiate fake news in order to influence events and policies around the world. Also, nothing is ever lost in the web. Every article stays and can be a source of information for the people that stumble upon it years after. This can have a negative influence on people who may rely on such information while making important decisions e.g., presidential elections, e-commerce and public behaviors like anti-vaccines, anti-masks etc.

This project tries to build a model using Natural Language Processing (NLP) and Machine Language techniques to classify a piece of news as REAL or FAKE.

CONCEPTS

Natural Language Processing (NLP) covers different approaches on how to process the human language. Supervised NLP algorithms can be used for categorization and classification of texts or parts of text. Machine learning for NLP and text analytics involves a set of statistical techniques in order to identify parts of speech, sentiment, and other aspects of the text.

The **Naïve Bayes Classifier** is a deterministic algorithm that uses the Bayes theorem to classify data.

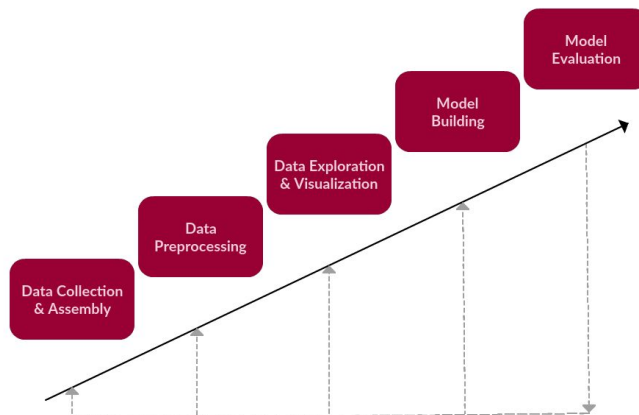
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

It uses probabilities of the elements belonging to each class to form a prediction. The underlying assumption in the Naïve Bayes model is that the probabilities of an attribute belonging to a class is independent of the other attributes of that class. Hence the name 'Naive'. The advantages of using Naive Bayes is that it is simple to compute, and it works well in categorizing data.

APPROACH

The proposed approach is to analyze the data using Python NLTK tools and Sklearn kit to perform basic natural language processing and machine learning methods to train the dataset to build a classifier to identify fake news from real ones.

The input dataset consists of both True and Fake news. We use 80% of the data set to train the model and the remaining 20% to test and validate the model. The dataset needs to be pre-processed and cleaned before it can be fed to the model. Feature extraction is done using NLP techniques. A classification model is built using Naive Bayes algorithm to perform sentiment analysis. Tfidf Vectorizer is used to convert the text to numerical representations and initialize the Naive Bayes Classifier to fit the model. In the end, the accuracy score and confusion matrix is calculated to analyse how well the model works.



SOFTWARE TOOLS

- Python 3.7
- Anaconda IDE
- Python Libraries: numpy, pandas, matplotlib, sklearn, nltk, textblob, itertools

DATASET

The dataset of this project was built with a mix of fake and real news obtained from Kaggle. The Kaggle dataset is in turn collected from various sources using web crawlers. The sources of real news include Bloomberg, Reuters, Yahoo News, and The Guardian among many. The sources for fake news include PolitiFact, ISOT, The Onion, USA Newsflash, Truth-Out, The Controversial Files and so on.

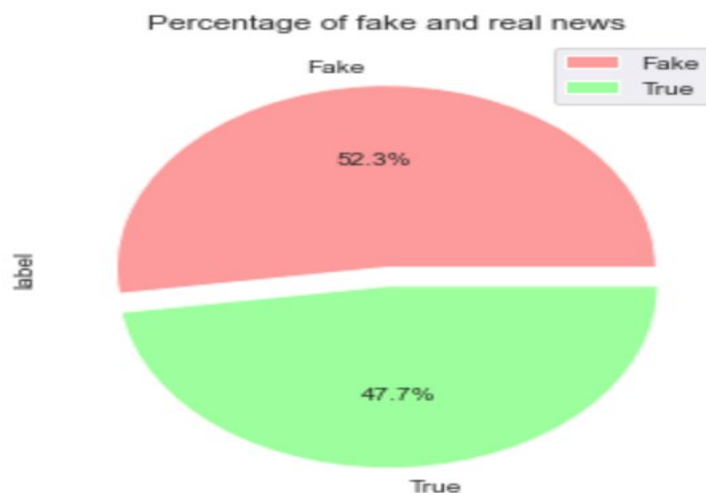
	title	text	subject	date	label
0	Donald Trump Sends Out Embarrassing New Year' ...	Donald Trump just couldn't wish all Americans ...	News	December 31, 2017	Fake
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	Fake
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	Fake
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	Fake
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	Fake

	title	text	subject	date	label
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017	True
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017	True
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017	True
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017	True
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017	True

Each dataset contains 4 columns and a 5th column 'label' is added based on its type, 'True' for real news or 'Fake' for fake news. The collected data was processed using various text preprocessing measures, as explained later and stored in CSV files.

The real and fake data were then merged and shuffled to get a CSV file 'news' containing a consolidated randomized dataset. Then the collected data was processed using various text preprocessing measures, as explained later and stored in CSV files.

This dataset 'news' has a shape of 44898×4. The first column identifies the news, the second and third are the title and text, and the fourth column has labels denoting whether the news is REAL or FAKE. Data is almost balanced when comparing both the datasets 'Fake' and 'Real'. It contained approximately 52% fake news and 48% real news articles.



Balance of fake news articles and True news articles for training and testing data sets

From these records, 80% was used for training the detection model and 20% was reserved for testing the model.

DATA PREPROCESSING

Data preprocessing refers to the various transformations that are applied to the data before feeding it to the model. The goal of text pre-processing is to get a reduced representation of the raw text. This includes removing null values, merging words of the same meaning to a single word etc. The reduced text enables the detection of specific patterns of the raw text. To detect unnecessary items and overrepresentation of words, statistical analysis of their occurrences in datasets was used.

Changing Uppercase to Lowercase

This step is necessary because all the text present in the news articles should have a common representation format. If the two words that possess the same meaning and are used in the same context, but they differ in the format of representation, then the matching algorithm would not be able to match both the words and will treat them as a separate entity. e.g. “Automobile” and “automobile” will be treated as two different words because they differ in their format of representation, which would be incorrect. Thus by reducing “Automobile” to lower case “automobile” the efficiency of the matching algorithm can be increased. The main disadvantage associated with this operation is that many proper nouns are derived from common nouns and can only be distinguished using cases. e.g. “General Motors” and “general motors” the first word refers to the company and it should be treated differently from the second word, but after applying case-folding operation both words will be in the same format and will be treated as the same. Thus case-folding operation leads to loss of information about the proper noun.

Removing Special Characters

Special characters (e.g. \$, ! etc.) don't possess any significance during content-based matching and should be omitted before applying the classification algorithm. Although by removing these special characters, some context-based information can be lost but there is no loss of content-based information. e.g. let's assume a news article contains a statement "cost of a book is \$100" now if the special character i.e. "\$" is removed due to preprocessing. The information about the currency will be lost and the statement after preprocessing "cost of an item is 100" now may be referring to dollar or rupee or any other currency.

Date Format Normalization

Since dates that are mentioned in the news may serve as the most important factor that can further improve the efficiency of the classification model. By using this information we can check whether the news is a derivation of the original news or fabrication of it. E.g. Date can occur in various formats in the news such as dd/mm/yy, mm/dd/yyyy, dd/mm/yyyy. All these forms should be converted to a standard form such that two dates that are mentioned in news and refer to the same day, then these dates should be matched with each other.

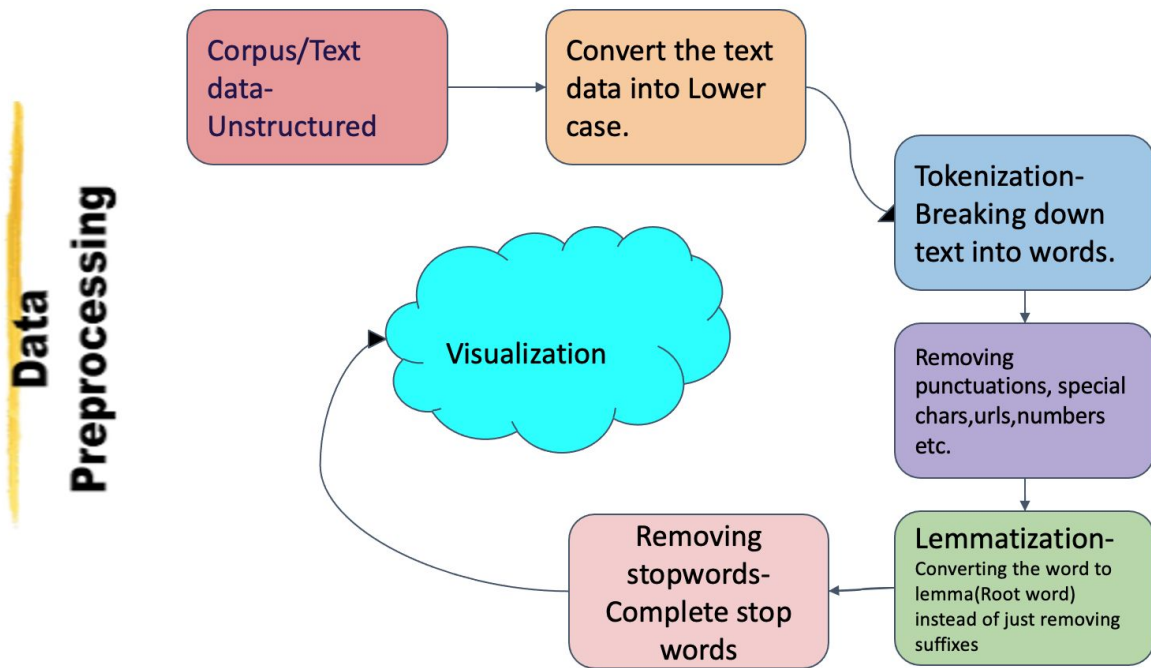
Stop Word Removal

Stop words are the most common words in a language (e.g. *a, an, the, was, in*, etc.). These words don't possess any discriminating power and need to be discarded before constructing the bag of words model, because stop words take up more space and increase the processing time, such words do not possess much relevance. The word having higher relevance possesses high local frequency (number of time word occurs in the given document that is to be classified) and low

document frequency (number of documents in the corpus containing that word). This paper focuses on removing the stop word by calculating its relevance using the term frequency-inverse document frequency (tf-idf).

Lemmatization

For grammatical reasons, people use different forms of a word (such as run, ran, running) or derivationally related words e.g. photograph, photography, photographic. Lemmatization is a process to group all the inflectional forms of a word so that all of them can be analyzed using a single item. This technique uses the vocabulary and the morphological analysis of the word to group all inflected forms. If two sentences only differ in the use of the inflectional forms of the word then these sentences should match with each other but if lemmatization is not performed then the classification model will mismatch such types of sentences. E.g. Jill runs, Jill is running, Jill ran. Although all these sentences are describing the action John does and so all the sentences should match syntactically. After lemmatization all these sentences map to “Jill run”, because of all the inflectional form of the word i.e., run, ran, running will now map to a common base run. The main disadvantage of lemmatization is the loss of timing information as different inflectional forms of word convey different timing information such as ran refers to the past, running refers to the present. But contextual information is not much relevant as compared to the content-based information for this problem domain.



DATA VISUALIZATION

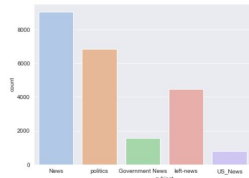
Subject statistics

Bar charts were generated with subject classifications for the true and fake news datasets. Below pictures show the subject categories with the number of fake and true articles in them. We can observe the high number of fake news in categories like politics, middle-east etc., which indicate the areas which are highly susceptible for false information.

Fake News Subject Breakdown

Fake News Subjects:

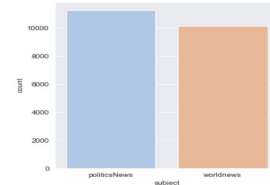
News: 9050
 politics: 6841
 left-news: 4459
 Government News: 1570
 US_News: 783
 Middle-east: 778



Real News Subject Breakdown

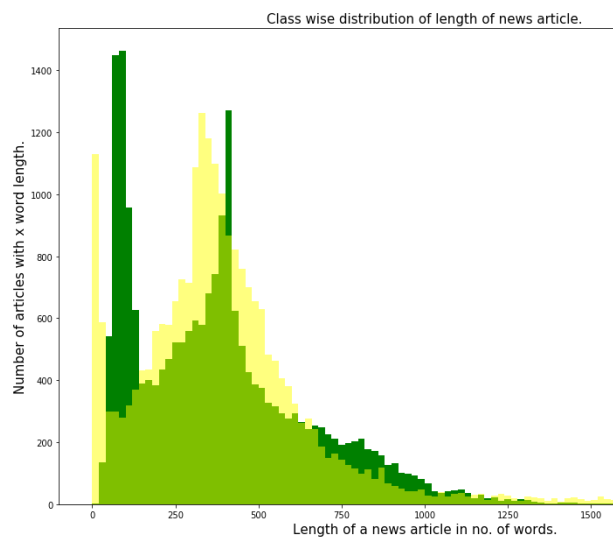
True News Subjects:

politics news: 11272
 world news: 10145

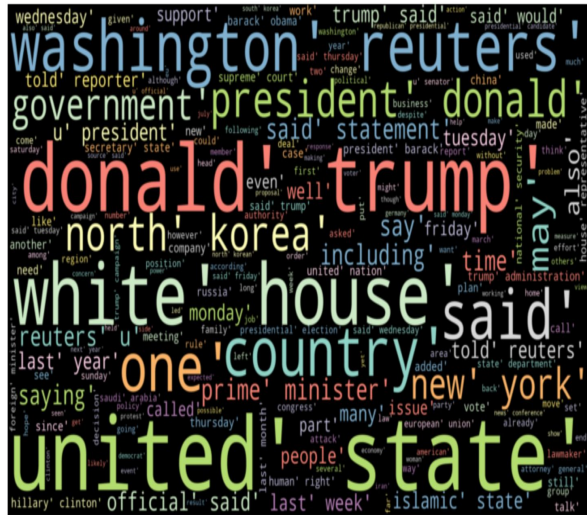


Article Length Statistics

Histogram bins the news articles together with the same word length. Rarely, the real news article contains more than 1400 words. On the contrary, there are comparatively much greater numbers of fake articles with word length more than 1400 words. Note that there are more than 1100 fake news articles with no text. This means the article contains only title, with no content.



A Word Cloud was generated for both Fake and True news representing the frequency of each word after the stop words are removed. we can observe some words appearing more often in the false articles, which include ‘hillary’, ‘republican’, ‘clinton’ etc., but don’t appear frequently in the True-labeled articles. Similarly, true articles include words like ‘thursday’, ‘year’, ‘reuters’ etc which indicate that the true articles include the dates and sources. These textual words can provide important signals for distinguishing the true articles from the false ones.



True news word cloud

To add onto our list of features, we used the package TextBlob to do sentiment analysis on the titles and text. TextBlob provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. It always gives you two scores that range from 0 to 1.

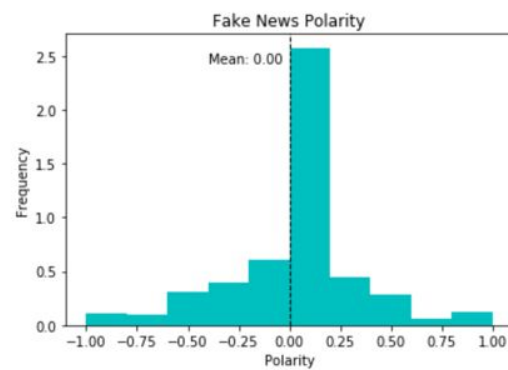
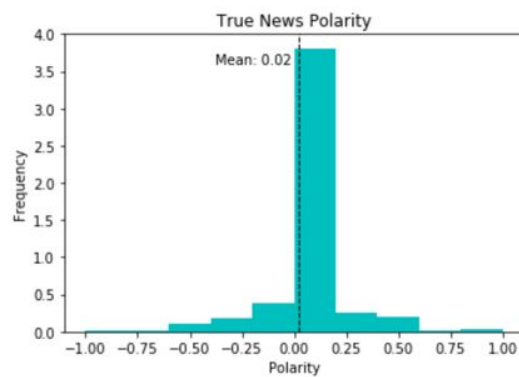
Polarity: It describes positive and negative emotions in the text, where 0 is most negative and 1 is most positive.

Subjective: Sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information. This ranges from 0 where it is most objective to 1 where it is most subjective.

And the data we explored was as below

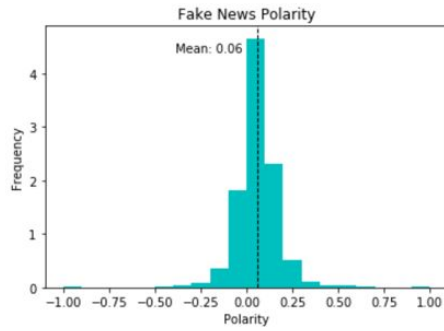
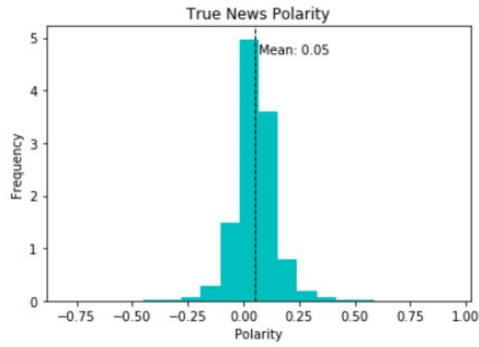
Sentiment analysis : 'Polarity'(Title) :

- Mean polarity of title/headlines is nearly identical & neutral regardless of fake/real news.
- However, spread is higher for fake news (more extreme scores).

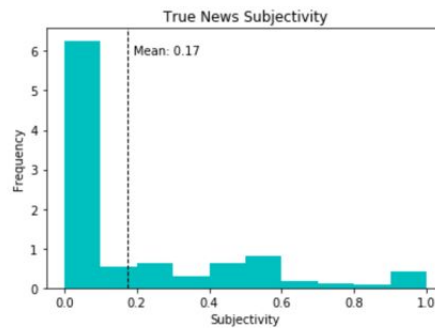
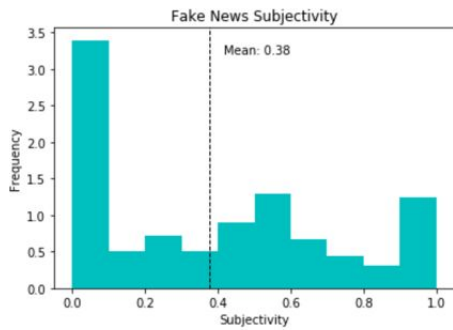


Sentiment analysis : 'Polarity'(Text) :

- Mean polarity of article text is nearly identical regardless of fake or real label and relatively neutral.
- Spread is similarly identical. Is subjectivity a better predictor?

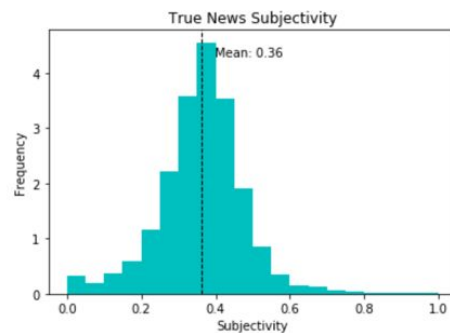
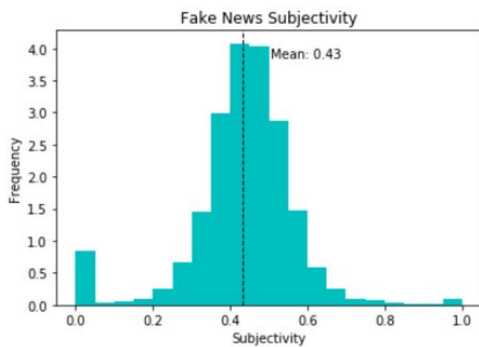


Sentiment Analysis : 'Subjectivity'(Title) :



- Greater spread of subjectivity scores for fake news and differences in mean (more subjective).
- True news tend to be more objective based on title/headlines.

Sentiment Analysis : 'Subjectivity'(Text) :

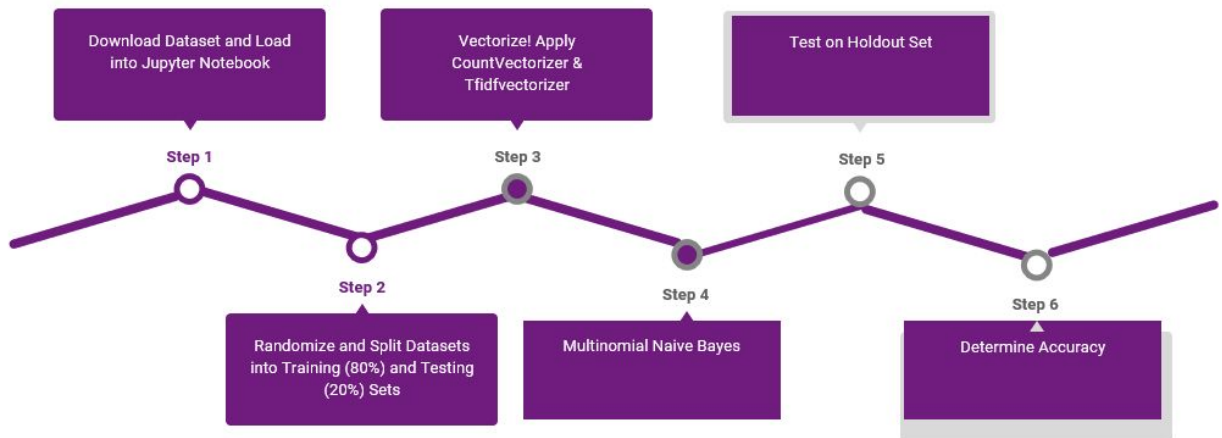


- Looking at article text, (though means similar), more fake news articles have scores > 0.4 (more subjective). Slight left-skew.
- More real news articles have scores < 0.4 (more objective). Slight right-skew.

MODEL

Steps followed

- Initialize a TfidfVectorizer
- Fit & transform train set, transform test set
- Initializing a CountVectorizer
- Fit & transform train set, transform test set
- Get the feature names of `tfidf_vectorizer`
- Get the feature names of `count_vectorizer`
- Build a naive bayes classification model and fit training sets
- Predict and calculate accuracy
- Build confusion matrix
- Build a naive bayes classification model and fit training sets
- Predict and calculate accuracy



After playing with the data and exploring it, we got a better understanding of what would be the suitable model to train the dataset. Initially we divided the dataset into training data(80% of data) and test data(20% of data) .Training data is used while training the model and to check how well our model performed we used test data. The whole dataset is randomized and used .For training the dataset, we used **TfidfVectorizer** module from `sklearn.feature_extraction.text`, The result is a sparse matrix recording the number of times each word appears and weights the word counts by a measure of how often they appear in the documents. We used 'scikit_learn's `train_test_split()`' for splitting the `text_count` (contains our X) and `dataset['Sentiment']` (contains Y).

Once we had trained data and test data is available, we started the process with the below steps.

Defining the model: As mentioned earlier to define the model we used Naive Bayes classifier. Naive Bayes classifiers have worked quite well in many real-world situations, especially for document classification and spam filtering. They mostly require a small amount of training data to estimate the necessary parameters.

Here we're classifying text into one of two groups/categories — positive and negative sentiment. Multinomial Naive Bayes allows us to represent the features of the model as frequencies of their occurrences (how often some word is present in our review). In other words, it tells us that the probability distributions we're using are multinomial. Also Tried Random Forest (multiple decision trees), but it was too slow so confined to the MultiNomial Naive Bayes model.

Compiling the model: sklearn's modules and classes were used , where they are imported as precompiled classes for compiling the model. The training time is 0.115 seconds.



Fitting the model: Generated the model-fitting the dataset in the MultinomialNB. Once the model is fit and accuracy calculated was 93.55% which is very decent and thus we can conclude the model worked great.

Evaluating the model:

Once the model is fit we need to evaluate it. So we used a confusion matrix here to evaluate the model, Confusion matrix is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done to test how accurate the model is. Using the confusion_matrix, we can plot a heatmap using seaborn to see how much records in the testing data the classifier has been successfully predicted.

RESULTS

After we built the model, it was validated against the remaining 20% dataset. The results obtained are as below.

Multinomial Naive Bayes		
	 CountVectorizer 	TfidfVectorizer
Accuracy	0.952561 or 95.26%	0.935523 or 93.55%
Precision	0.952579	0.935517
Recall	0.952561	0.935523
F1	0.952566	0.935514

Accuracy is used when the True Positives and True Negatives are more important while F1-score is used when the False Negatives and False Positives are crucial. Precision refers to the

percentage of your results which are relevant, recall refers to the percentage of total relevant results correctly classified by your algorithm.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

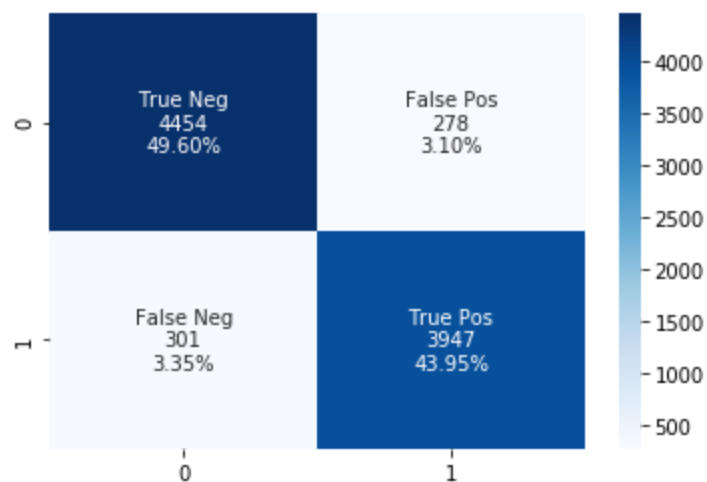
$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$\text{F1} = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

As you can see, the model accuracy is 93.55% which is very decent. The Precision and Recall values are also close to 1, which means that the model is performing well.

With basic preprocessing steps and using a relatively simplistic model (Naive Bayes), the model achieved 94% accuracy on the test set and high precision and recall. Further, the model performance does not drop significantly between the train and test set, so it looks like we have avoided overfitting.

Model is also evaluated using the Confusion matrix and the results were as below.

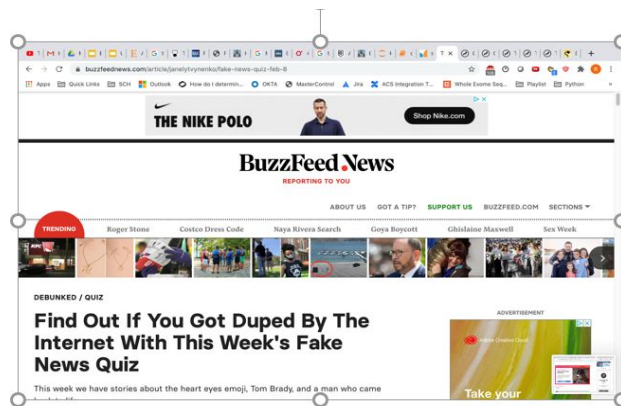


	pred:fake	pred:real
true:fake	4454	278
true:real	301	3947

From the plot, we can see that the result is pretty good.

ONLINE TOOL TESTING

Used the model on a online True/Fake detection quiz to calculate how efficient the model performed.



The model was able to correctly guess 5 out of 7 questions in the quiz. The model was confused by the sensationalist (but true) titles like “A man who was revived from a drug overdose stole and crashed a police cruiser.” and “A woman found a man she didn't know in her apartment

closet, wearing her clothes.”. However, the model was able to guess correctly when full articles/text were inputted (not just headlines). This could be solved by broadening the training data.

CONCLUSION

Introduced the naive Bayes model for classification and applied it to the text categorization task of sentiment analysis. Naive Bayes with binarized features seems to work better for many text classification tasks and from this project below points were concluded.

1. What are some trends that were observed in the fake articles vs real ones?

- Fake news articles tend to be longer!
- Fake news articles contain more names/nouns, while real news articles contain more dates and locations.

2. What is the polarity and subjectivity of the articles labeled as fake vs real?

- Less differences in median or mean polarity, though more spread in the data for fake news.
- Fake news articles have higher subjectivity scores.

3. Can we use machine learning to correctly classify whether an article is fake vs real?

- Yes! However, there are a wide variety of news articles beyond the subjects analyzed in this dataset. Need to broaden training dataset and explore relationships between words.
- Also, examine **subjectivity** as a predictor in the next iteration.

4. Overall, what implications can NLP and ML have on our society and how we ingest information?

- From Multiple Sources (example: News Agencies, Google, Facebook, Twitter...) information can be overloaded and we will need automated solutions.

“When all else fails, **Google the headline** of the article and see if there are results indicating it’s a fake news article ;)”

REFERENCES

1. Lina L. Dhande and Dr. Prof. Girish K. Patnaik, “Analyzing Sentiment of Movie Review Data using Naive Bayes Neural Classifier”, IJETTCS, Volume 3, Issue 4 July-August 2014, ISSN 2278-6856.
2. A Gentle Introduction to Scikit-Learn: A Python Machine Learning Library-by Jason Brownlee on April 16, 2014 in Python Machine Learning.
3. 6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R
SUNIL RAY, SEPTEMBER 11, 2017 LOGIN TO BOOKMARK THIS ARTICLE
4. Wang, W. Y.(2017). “liar, liar pants on fire”: A new benchmark dataset for fake news detection.
5. Movie review sentiment analysis with Naive Bayes | Machine Learning from Scratch (Part V)
Learn how to process text data.
6. K. M. Leung, “Naive Bayesian classifier,” [Online]Available.
7. Sentiment Analysis- <https://streamsql.io/blog/sentiment-analysis>
8. TextBlob: Simplified Text Processing Release v0.16.0. (Changelog).
9. Natural Language Processing with Python--- Analyzing Text with the Natural Language Toolkit Steven Bird, Ewan Klein, and Edward Loper
10. Natural Language Processing (NLP) with Python — Tutorial by Author(s): Pratik Shukla, Roberto Iriondo
11. Other websites like stackoverflow, Github , Medium etc.