Priyanka Racharla

Fake News Detection in English and Spanish

History of Fake News

Fake News is defined as “false stories that appear to be news, spread on the internet or using other media, usually created to influence political views or as a joke” (Cambridge Dictionary). Fake news has become increasingly more prevalent as the years have gone by, but it has existed for years. Some past examples include: Nazi propaganda about antisemitism and the false stories about African American’s crimes in the early 1800s. In the late 1890s, fake news become so publicized in newspapers that a term was coined for it: “Yellow Journalism”.

The motivations behind fake news vary from lighthearted jokes to nefarious purposes like damaging someone’s reputation. Yellow Journalism first started as an attempt by newspaper companies to get people’s attention. The Sun gained its popularity from a false article about how there were aliens living on the Moon in 1835. The tactic of writing exaggerated, sensationalized news articles led to the popularity and profitability of newspapers in the 1800s. Even today, this tactic is used for some tabloid sites and on YouTube channels that make news video reports. Youtube channels will use false titles for the sake of getting more views and from those views, they are able to make a profit on ads. Other motivations behind fake news include damaging someone’s reputation. In May 2020, the popular singer, Doja Cat faced false allegations of participating in White supremacist chats and stripping on camera. False claims and photo evidence of Doja Cat being in video chats was posted on Twitter. Within 24 hours, the hashtag #DojaCatIsOverParty was trending. Fake news can also take a more positive form. An example is The Onion, a popular satirical news company that sheds light on problematic issues in society via fake news.

Introduction

Regardless of the motivations for fake news, it has become increasingly easy for misinformation to spread. Nowadays, established newspapers are not the sources of fake news, but small no-name tabloids and even individuals on social media spread fake news. The rise of the internet has created an information overload making it hard to distinguish between actual and fake news.

Given the information overload and the impacts of false information, a fake news detector would be helpful. Currently, there exist models to detect fake news, but they are mostly in the English language. Support for fake news detection in other languages like Spanish is minimal. Spanish is 4th most popular language in 2022 so it would be useful to be able to detect fake news in Spanish. Thus, I decided to build models to detect fake news in Spanish and English as accurately as I could. I also compared the models to see if they would yield similar levels of accuracy for predictions. The models for both languages were trained on data that underwent similar processing steps and then the models were tested twice. Each model was tested on data originally in the same language and on data that was translated into the model's respective language. Both English and Spanish fake news detection was done using the following models: Naïve Bayes, Logistic Regression, Stochastic Gradient Descent, Random Forest, and Decision Tree. The best performing model in any language was Logistic Regression. Unfortunately, Spanish trained models generally performed poorer compared to English trained models. Spanish models were more accurate only on the original Spanish data when compared to English models on the same data translated into English.

Related Works

In " Fake News Detection Using Machine Learning Approaches" by Z Khanam and "Fake news detection in social media" by Kelly Stahl, the process of creating a fake news detector in English is explained. Both papers used Natural language processing to different degrees on text articles. Khanam and Stahl both did sentiment analysis. Sentiment analysis is the analysis of emotions and polarity (whether the text is positive, negative, or neutral) in a text. But, Khanam did even more natural language processing such as tokenization, part of speech tagging, and named entity recognition. Named entity recognition locates and classifies entities like people, organizations, or places. I did not use sentiment analysis or named entity recognition, but I did use tokenization, stop word removal, part of speech tagging and lemmatization. I wanted models in both English and Spanish to be similarly constructed so they could be compared more easily. After the data had undergone natural language processing, it needed to be translated into Vectors, which computers can more easily deal with. I chose to use only a CountVectorizer, while Khanam's paper used CountVectorizer and TfidVectorizer. As for the models trained on the data, I picked ones that I was familiar with. Khanam and Stahl both used Naïve Bayes, which

I also used. Khanam also used Random Forest and Decision Tree for models, which I used as well. The only models I did not use were Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). Khanam and Stahl used SVM, while Khanam only used KNN. I did use two different models: Logistic Regression, Stochastic Gradient Descent on data. When evaluating model performance, I used a confusion matrix and bar chart for accuracy in addition to a precision vs recall line chart. Khanam's papers evaluated his results with a confusion matrix and bar chart too.

Data

The English fake news dataset was found on Kaggle.com and has almost 8000 rows of data. The link to the English fake news dataset on Kaggle: https://www.kaggle.com/datasets/nopdev/real-and-fake-news-dataset There was not much information on the source of the data, but the Kaggle usability rating is 5.3. I also independently verified around 75% of article sources to see if the dataset was correct in classifying fake news or true news. The Spanish fake news dataset was sourced from the IberLEF 2021 conference, a conference on natural language processing in Iberian languages. The link to the Spanish fake news dataset: https://github.com/jpposadas/FakeNewsCorpusSpanish Although there was a verified source for this dataset, it was rather small, consisting of only 677 rows. For both datasets, there was some initial processing such as dropping irrelevant column information, renaming column labels, and mapping values in Category field to 'Fake' and 'Real" to standardize the datasets. The Category field tells whether news is fake or real. I also dropped rows with missing relevant information in Text field and Category field. Text field contains the contents of articles. After all this preprocessing was done, I split each dataset into training data and testing data. Roughly 80% of the original dataset was used for training the models and other 20% was used for testing them. I saved the training and testing data for each language in .csv files to make access easier. Once the data was split, I took each language's testing data and translated it into the respective other language. Translation was done using Google Translate function in Google Spreadsheet.

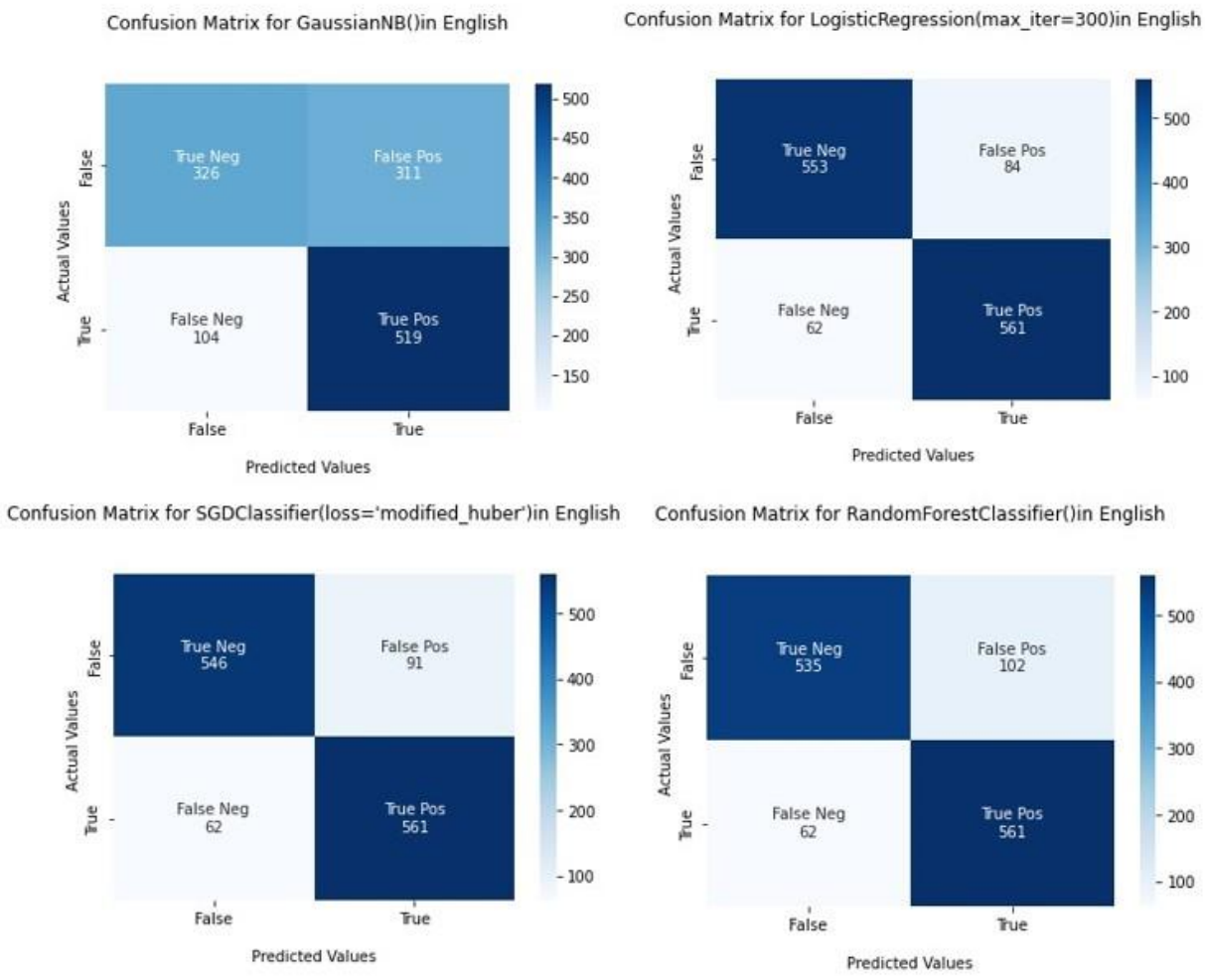
Methods

Once the data was preprocessed, the text data (article content) was further processed with help of natural language processing and afterwards, the text was transformed into vectors and fed

into models. Before natural language processing, the text data was lowercased, stripped of URL's and any non-alphanumeric characters. I used the Natural Language Toolkit (NLTK) in Python for natural language processing in English and parts of Spanish. NLTK performed tokenization, stop word removal, part of speech tagging, and lemmatization. Tokenization is the process of dividing text into smaller pieces called 'tokens'. Tokens can be characters, subwords, words, and even sentences. For the data, I used words as tokens. An example sentence: "My name is Joe" is tokenized into ['My', 'name', 'is', 'Joe']. Stop word removal is the removal of common words that do not directly impart any meaning to a text. Stop words include: "the", "is", "a", "an". These words are removed from text so that distinct words can be focused upon. Part of speech tagging is the recognition of a word's part of speech based on the context the word is used in. Lemmatization is finding the root word, 'lemmas', from a word. Lemmatization relies on part of speech to find lemmas. For the Spanish text, NLTK can be used for tokenization and removing stop words. Stop words in Spanish include: "el", "la", "es". Lemmatization and part of speech tagging is done using the Simplemma library because NLTK does not support lemmatization in Spanish. After all this natural language processing, the text is stored in .csv files for easier access in the future. Now the text consists only of relevant words to the article's meaning and is simple enough to transform into vectors. A numerical vector is much easier for a computer to perform computations on than a list of strings. The CountVectorizer method from the SciKit library is used to transform the text into a vector based on the frequency each word occurs in each article.

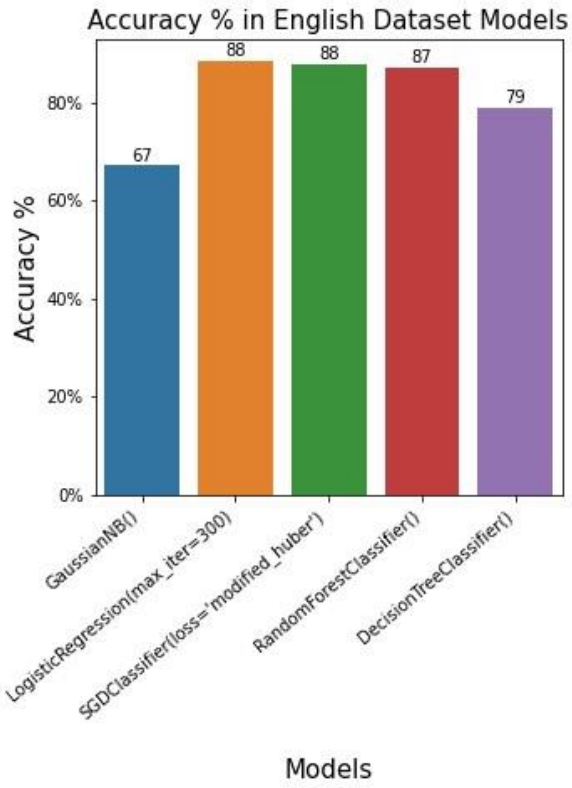
Experiments

Vectorized text data is then fed into models. The following models were trained in English and Spanish: Naïve Bayes, Logistic Regression, Stochastic Gradient Descent, Random Forest, and Decision Tree. For each dataset tested, a confusion matrix was made for each model. The confusion matrix displays the number of true positives, false negatives, false positives, and true negatives. My confusion matrix is also color coded based on the number of values in a box, so it shows visually how well a model is performing. The matrix also can show whether the model has more issues with wrongly categorizing fake news (false negatives) or wrongly categorizing true news (false positives).

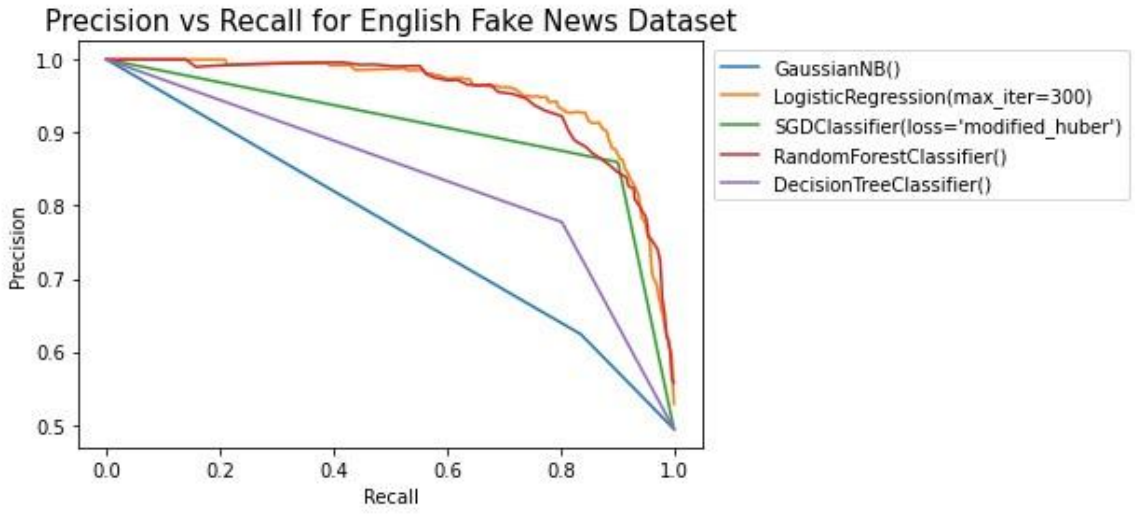


[Confusion Matrices for testing English dataset on English trained models]

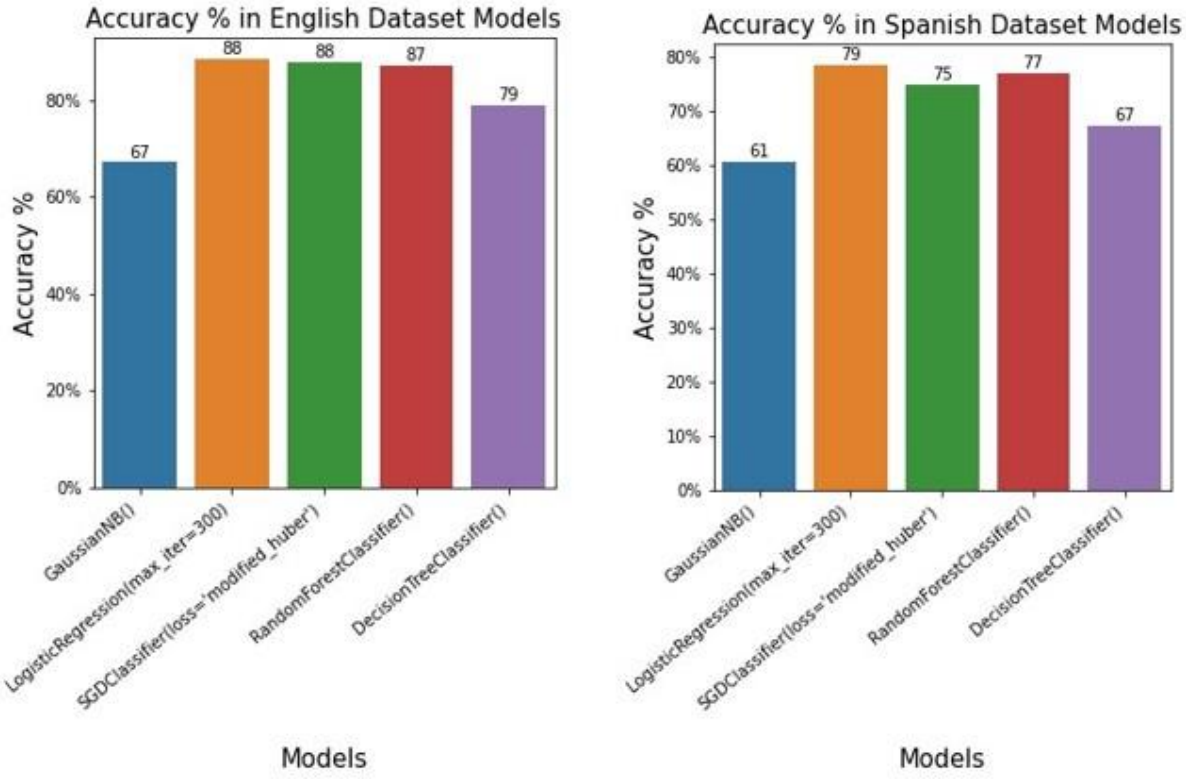
The accuracy percentage of a model is just a statistic and holds more value when compared between models for a dataset.



The graph shows that Logistic Regression and Stochastic Gradient Descent are most accurate performing model on English data, but Logistic Regression is slightly more accurate actually. Another graph to visually represent performance of models is the Precision vs Recall line chart. Precision is the percentage positive instances out of total predicted positive instances, while Recall is the percentage of positive instances out of total actual positive instances. In the line chart, the higher the area under the line, the better the model performs.

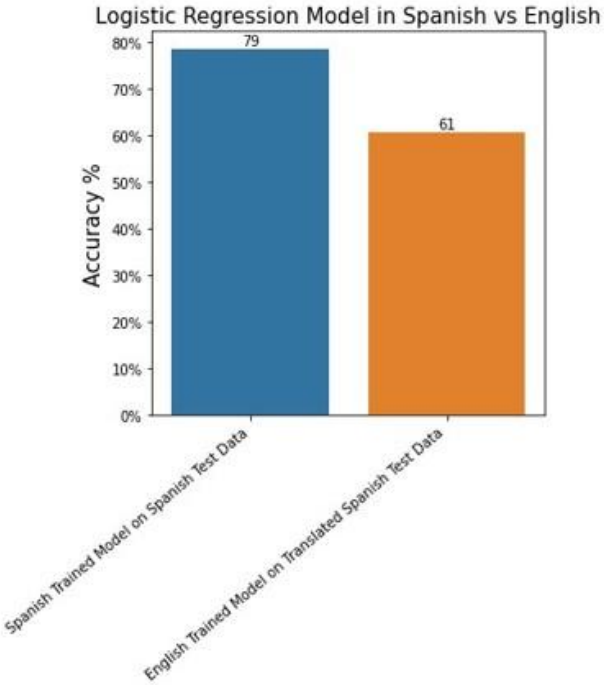


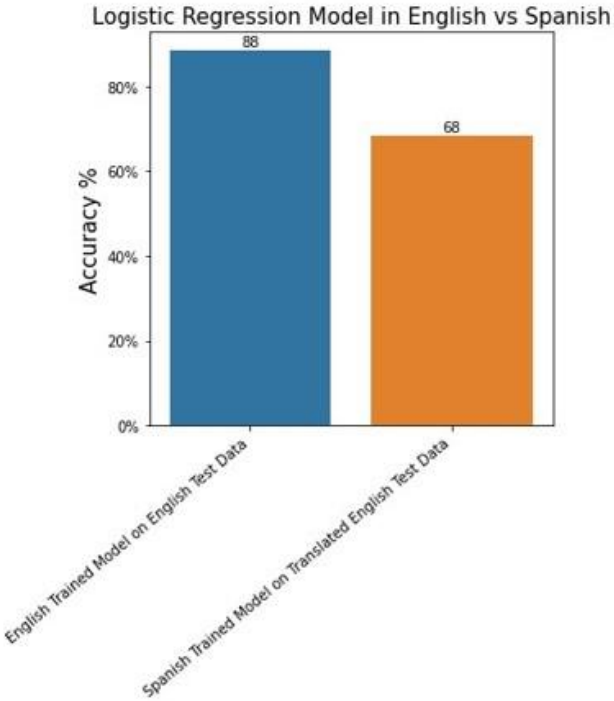
The graph revalidates that the Logistic Regression model performs the best on English data. In Spanish data, Logistic Regression also performed the best. When comparing model performance in English and Spanish, models in English performed an average of 11.6% better than Spanish ones. The only time English models performed worse was on Spanish data translated into English when compared to the Spanish model on the original Spanish data.



Since Logistic Regression was the best performing model for any data, I compared both Logistic Regression models in English and Spanish on two sets of data: 1) the test data portion of the English dataset and 2) the test data portion of the Spanish dataset. The English test data was used to test the English model and the data was translated into Spanish to test the Spanish model. The Spanish test data underwent a similar process. Overall, English and Spanish models performed

best on data that was originally in the same language. Both language models performed about 10-20% poorer on translated language data.





Conclusion

In conclusion, English trained models outperformed Spanish trained models by 11.6%. The models performed the best on data that was originally in the same language. Spanish trained models did so poorly when compared to English trained models possibly because of the small amount of Spanish data to train the model on. The English dataset had about ten times as much data as the Spanish dataset.

In the future, I would like to train the Spanish models on more data and try to achieve a similar accuracy level to English models. Furthermore, I would like to build an interactive application where you could see whether news was fake or real. You would be able to choose a model and feed in the contents of a newspaper article. The application would then tell you whether it was fake or real news along with the percent accuracy for the model.