

CS5830 Project 5: Naive Bayes

Caden Maxwell, Calvin Bell, Karl Poulson

Introduction

Our analysis delves into the domain of fake news detection, a critical area given the proliferation of misinformation in today's digital age. Leveraging a dataset sourced from Kaggle, a data science and machine learning competition platform, we utilized a Multinomial Naive Bayes model to predict whether a sample of news articles were real or fake. Through our analysis, we achieved promising results with relatively high statistical scores, which validates the potential for our model to play an important role in the ongoing battle against fake news.

Github: [Naive Bayes](#)

Slides: [Google Slides](#)

Dataset

The dataset used in this analysis comprises 72,134 news articles, including 35,028 instances of real news and 37,106 instances of fake news. It was procured using information from four separate information sources, including another Kaggle dataset, and McIntire, Reuters, and BuzzFeed Political data. This diverse dataset represents a much more generalized sample than any of those data alone, which helps to ensure the robustness of our model and prevent overfitting. The dataset itself is structured around three key features: the title of the article, the text composing the article, and an authenticity label, where a label of 0 denotes an instance of fake news and 1 indicating real news.

Dataset: [Kaggle](#)

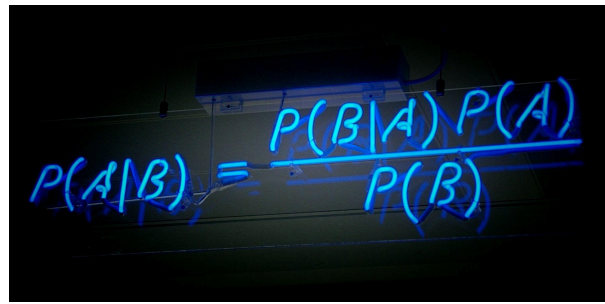
Analysis Technique

In our analysis, we split our data into training and testing sets using the `train_test_split` function to fairly assess our model's performance. To convert text features, like titles, into numbers for our model, we used the `CountVectorizer` from `scikit-learn`. This tool helped us turn words into numerical values, making it possible for our model to learn. We also filtered out common English words that don't help with classification using the `stopwords` parameter.

Our study focused on how well our Multinomial Naive Bayes method could spot real and fake news articles. This method is based on the assumption of conditional independence between features given the class. Bayes Theorem assumes each feature, in this case a word, is

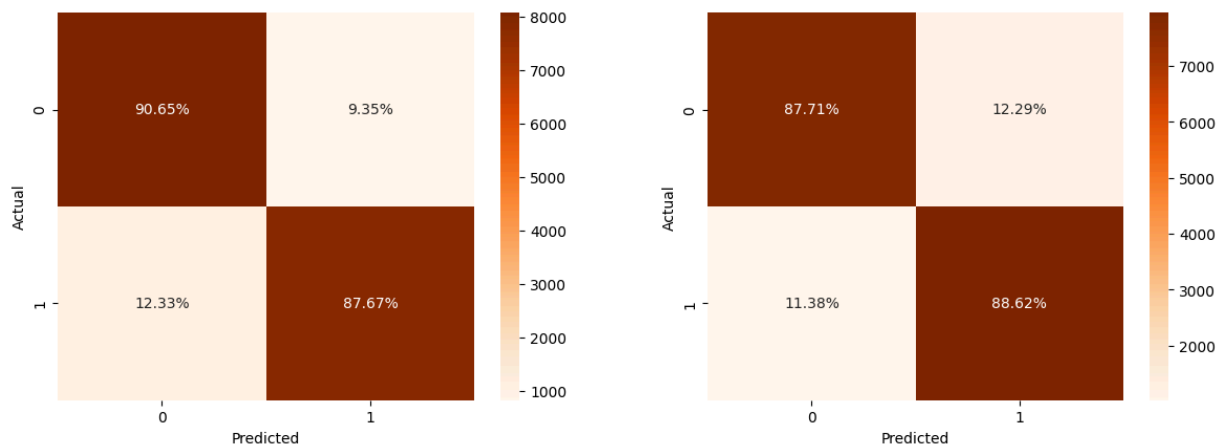
independent. This assumption can cause some shortfalls in our model's robustness. However, our model is able to train quickly without the use of a GPU.

We looked at key measures like accuracy, precision, recall, and F1 scores to see how our model performed. We fit our model on the text and headline separately and combined the results with a logical OR to achieve our best score.


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

Results

We found the analysis to be very suitable for the data. When using both the title and the text, we got an F1 Score of 0.9034. This shows an overall strong performance of the model with the dataset. We ran a k-fold and standard test-split test and received the same scores within 2 significant figures. We believe that this is because our dataset is diverse and large.



Here's the comparison between using the text on the left and the titles on the right. Both perform similarly, though titles show slightly higher precision, likely because they're designed to grab attention with catchy language. This makes it easier for the model to identify patterns. Overall, our models fit the data well, boasting a high F1 score, which means they might be good for gauging article truthfulness.

Technical

Countvectorizer: fantastic little function! lowercased our words, tokenized, and removed stop words then turned it into a vector. Dataset was very clean and usable. We filtered through all of the words and cleaned them up by making them all lower case and removing common filler words and punctuation.

We found our results by creating a Multinomial Naive Bayes model provided by Sklearn. This model was used for our prediction process. We used Train_test_split to divide our training and our testing data.

We decided to do a logical or to combine the title and text predictions. Next time, we could try to test if the string of the title and the text should be combined prior to creating the model.

Conclusion

In conclusion, we're pleased with what we've learned from this dataset and plan to keep working with it to uncover more insights.

