COVID-19 Fake News Classifier

Problem Statement:

Can we use NLP tools to decipher differences in language between fake news and real news on COVID-19. The objective of this project is to see whether classification models can accurately predict the origin of text from headlines that are either fake or real. I seek to understand what kind of language is being used in news sources the public should trust and sources we can dismiss.

Executive Summary

This project uses APIs specifically NewsApi, natural language processing techniques, and classification models to understand what kind of terms are being used in fake and real news on COVID-19. The website poynter.org is a database that unites fact-checkers in more than 70 countries and include articles published in 70 countries and 40 languages. NewsApi.org is an up to the minute software interface program which connects users to all relevant news sources.

Data Acquisition and Cleaning

Using News API I specified my search to just the terms "COVID-19" and "Coronavirus" my sourcers were bbc-news, abc-news, the associated press, bloomberg, cnn, financial post, and cbs news. For acquiring the fake news data a colleague of mind scraped poynter.org and passed along the fale in csv form.

In the cleaning department, since the poynter.org only had headline textual data I had to remove from the NewsAPI data the author, the source (which will discuss later), the description, the url, the published time, and the content. You'll only find one CSV in my data file - that is because I cleaned, trimmed, merged and saved in one fell swoop as oppose to working with multiple CSV files.

Classification Modeling and SHAP Values

Using a gridseraches and pipelines I was able to iterate over multiple parameters to distinguish which performed best. The models used were:

Logistic Regression

Decision Tree

Random Forrest

XGBoost

The last portion of this project, and most critical will be the creation of a SHAP value chart, which will communicate back to anyone who sees this just what terms have the highest average impact on our model's output.

Conclusion

All of the models created with the exception of the XGBoost package were overfit to the training data. In the future I will continue to play around with hyperparameters even though the parameters set out a wide range of possibilities. I believe the crux of the issue with this project is simply not having enough data. One reason to believe this is each best performing model actually kept out normal english stop-words.

Next Steps

The number one thing I want to explore the next I run through this problem is creating a new set of stop words that might better fit this data. The second thing I want to explore would be to run the same models on stemmed and lemmed data, which is avaiailable to me as I did that in my pre-processing stage.