Gigi Gabeau
20sum1metcs521sc2
Burnstein

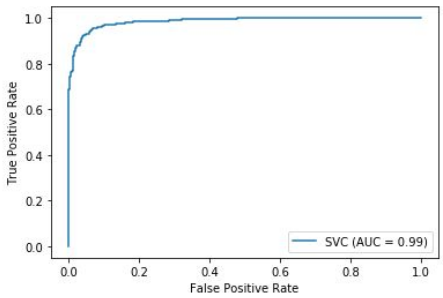
Discussion Writeup

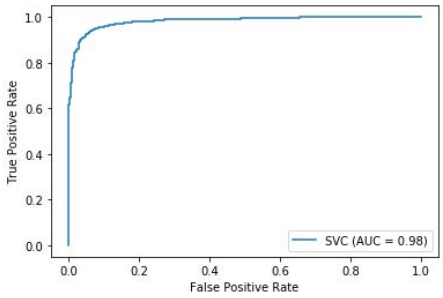
Oftentimes, people find themselves reading an article that seems to be factual, informative and eye opening but actually is biased, inaccurate, or simply just fake news. How do you know if an article is legit? This project aims to solve that problem or at least take the first step in creating a solution by making a program that takes in the url of an article and lets the user know if that article is real or fake.

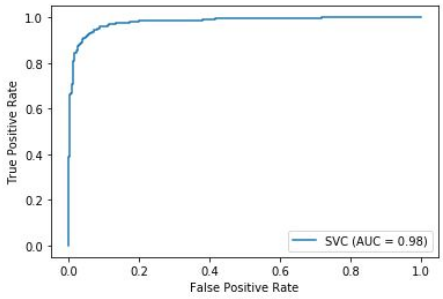
There are five files included in this project. The **ReadMe** details how to reproduce the results using the **Fake News Final Project.py** file. The **download script** opens and reads the **news.csv** file which contains a dataset of news to train and test. Finally this **write up** will walk through the Fake News Final Project.py file explaining the methods and techniques used as well as how data was analyzed to produce its results.

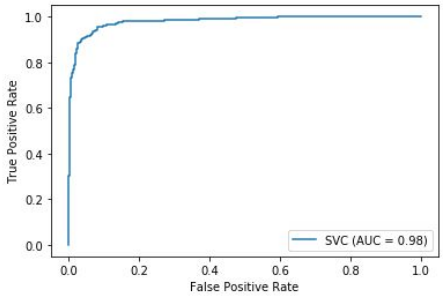
In the Set Up section of the program, many different packages are being imported, notable ones being pandas, matplotlib, sklearn and beautiful soup. Pandas' primary use was to create data frames, matplotlib to create graphs, sklearn to run the training and tests and beautiful soup for web scraping. The TfidfVectorizer splits up the data into five random groups then tests 20% of the data against the 80% trained data.

The Proof of Text Model Concept and Proof of Title Model Concept compares the results from training and testing based on just the title name versus the text of the article. 2 visualisations are produced for each category and are depicted below. Figure 1 and Figure 2 compare the True Positive Rate (TPR) against the False Positive Rate (FPR) for data training based on text and titles. The sweet spot on the graph is the top left corner as that is where the TPR equals 1 and FPR equals 0 (100% accuracy). By looking at the two graphs side by side you can see that the graph based on text yields higher rates of accuracy than the graph based on titles. Intuitively this also makes sense. Because the text of an article has more words than the title, there is more data to analyze, thus the program is able to make a more accurate read when analyzing based on text as opposed to just the titles. The numbers match this analysis as well- the average positive rate for the graph based on title was 90.4% as opposed to the graph based on text which averaged at 98.2%.









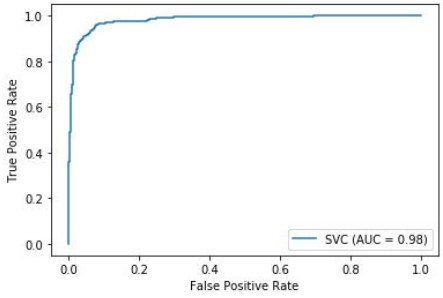
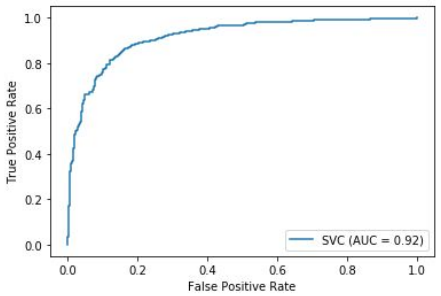
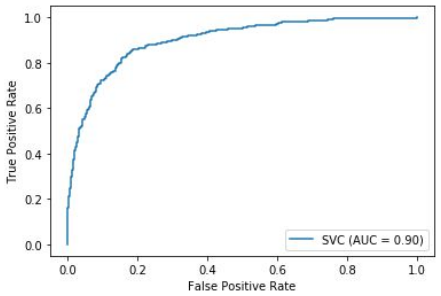
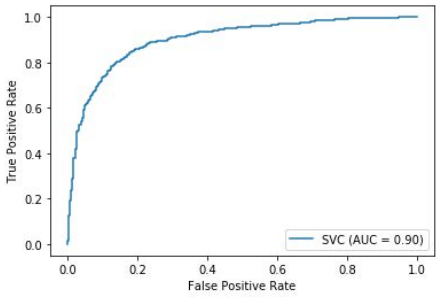
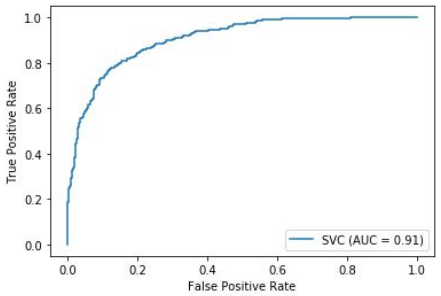


Figure 1









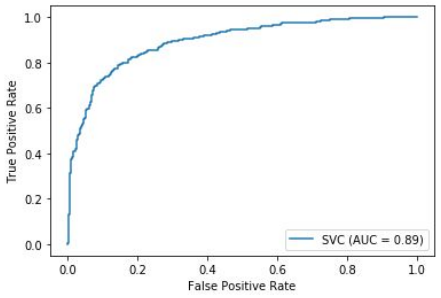


Figure 2

Figure 3 and Figure 4 compare Percent Accuracy of testing based on if the article is real or fake. The graphs for text and title were similar in that precision (avoiding false 'REALs') was higher when the article was real whereas recall (avoiding false 'FAKEs') was higher when an article was fake. This is great to see because it shows that regardless of if you use text or title, while text is generally better, both cases yield a pretty positive result. In other words, this program returns highly accurate results.

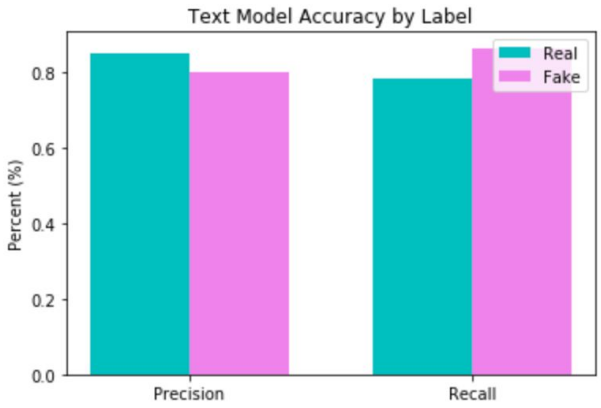


Figure 3

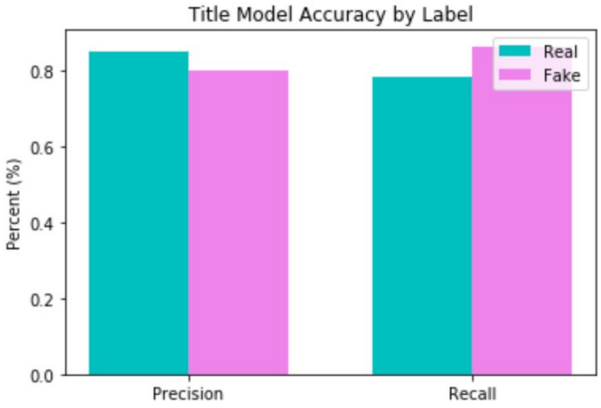


Figure 4

The Creating Models sections simply creates models using all of the data from the news.csv file to be used against the user's inputted article for testing. Two models are created, one for testing by text and the other for testing by title.

Next is the Web Scraping and Formatting section which goes through the code of an article url given by the user and searches for the text and title of the article by removing irrelevant tags and details.

Lastly, the Test Article sections allow you to input article urls to test by text or title. When testing different news sources, it was discovered that some sites block web scraping from occurring so if the situation arises where the program is unable to scrape through a site, the option of copying and pasting the relevant information is available. When these sections are run, the program prints either "This article is fake. Find a better source," or "This article is real. You can trust this source."

This project has taught me a lot of things including how to use matplotlib and sklearn to create graphs, how to load a document programmatically using requests and more. What was most fascinating about this project was that I didn't have to do much work in terms of how to analyze categorical data because sklearn can do that on its own as long as it has enough information. I am excited to see how much further one can take this project. The next stepI would be interested in taking is evaluating bias among real news. While an article may be from a legitimate and real news source, it doesn't necessarily mean that the information relayed can be taken as is. Creating a program that could report on the biases of real news would

definitely be an interesting tool given the state of our country in terms of politics and ethics.