**About Our Model:**

We created an unsupervised machine learning model using sklearn. We used it to split our data (train\_test\_split) and to create a TF-IDF vectorizer for our dataset. The purpose of that was to convert text to feature vectors since our dataset was composed of articles. The TF-IDF is a statistical measure that determines the relevancy of a word in a document. We also initialized a PassiveAggressiveClassifier from sklearn. This algorithm is ideal for large amounts of data, like our news articles, and making predictions. This model came out to be 92.74% accurate. The training score was 84.06%. After training the model, we tested it using the scraped articles we got from Facebook and appended each result to the dataframe as its own column in order to create our visualizations. Then we saved the model using “Pickle” so we could load the trained model onto Flask.

After loading the model onto our flask app, we utilized “Newspaper”, a python library for article scraping. The summarized article content loaded from “Newspaper”, is then run through the model to predict. The prediction is rendered through the template and could be seen on the website as “This article is “FAKE” or “REAL””.

**Data Visualization:**

**Alicia**

Using the results from testing our model, I made a couple of bar graphs and scatter plots to summarize our finding. We web-scraped from six news channels via Facebook posts - BBC, Buzzfeed Politics, Conservative Post, CNN, Daily Mail and Fox News. Surprisingly, out of 1982 news articles we tested, 1477 came out as Fake news and 505 came out Real news. The articles that we managed to scrape from Facebook are ranging from 274 to 373 across the six major news channels. As I further analyzed the number of fake and real posts among these news channels, I found that Daily Mail came in top in spreading fake news, followed by Conservative Post then BBC. The total number of posts extracted from Daily Mail was 373, the number of posts for Conservative Post was 353, and the number of posts extracted from BBC was 274. Daily Mail has about 87% fake news content, followed by Conservative post with 84% fake news content and lastly with BBC coming up at 94% fake. This is utterly shocking as I would have assumed the BBC would contribute the least amount of fake news content. This could also due to our prediction model accuracy. We managed to get 90-92% accuracy and this could possibly contributed to the reason why BBC has such a high percentage of fake news content. In contrast, we extracted 320 posts from Fox news, 312 from Buzzfeed and 350 from CNN. The top channel with the least amount of fake news content was Fox News, followed by Buzzfeed then CNN. With our tested results, it was showing approximately 58% real news for Fox news, followed by Buzzfeed with 35% then 34% for CNN. This clearly shows that Fox news is the most trust-worthy news site among all the six news channels. When I further analyzed the content of the data to explore trends, I found that most of these news articles were political news and mostly fall into the category of the Election and government law.

**Medha**

I used the data csvs from the tested model to try to analyse a little bit into the content of the articles that we were studying. In order to do so, I created weighted word clouds to observe the most commonly used words in the articles. I categorized the data as two different csvs of fake data and real data and then used gensim and matplotlib to draw out word clouds for both of the datasets. The words are weighted in such a way that the more commonly occurring words are displayed in a bigger font size than the rest. I observed that in both the word clouds the words, “Biden”, “President” and “Trump” were the highlights. This makes me believe that there is a lot of news available in the topics of Politics and the presidency and the fact that it is repetitive even in the fake news word clouds confirms my suspicions that we do consume a lot of misinformation about the political situation in the USA.

I wanted to do a study of grouped words to determine the most probable accompanying words with certain primary words. For this, I employed t-SNE which is derived from PCA plots. The t-SNE had used sklearn modules and gensim models to convert the words into vector and then used matplotlib to plot the curves. Once the plot was complete, the words were put into groups and I could study the words that commonly accompanied other words and the probability of those words appearing alongside that specific primary word.