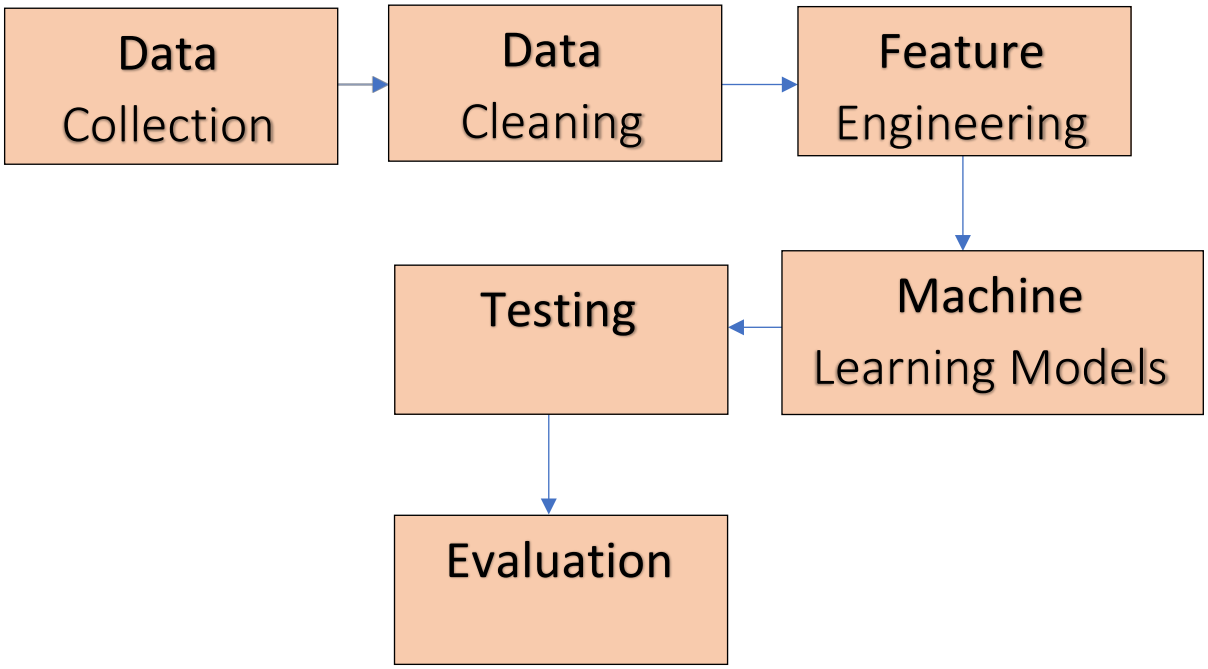
Text Classification using Spark:

Data Pipeline:



Data Collection

- Our code pulls URL data from the New York Times API for articles relevant to the following topics: “Mueller”, “Cohen”, “Trump”, “Democrat”, “Republican”, “Ronaldo”, “Superbowl”, “Yankees”, “LeBron”, “Knicks”, “Billion”, “Walmart”, “Bitcoin”, “Warren Buffet”, “Amazon”, “Google I/O”, “NASA”, “A.I.”, and “iPhone”.
- We created Python functions were used to scrape the New York Times website using Beautiful Soup to extract the articles.
- The articles were saved in a CSV file, along with their respective category (Business, Technology, Sports, and Politics).

Data Cleaning

- The data was cleaned in the PySpark code.
- Our code removes stop words such as her, his, their. These words do not contribute to the analysis.
- The code removes punctuation and numbers as well, to add interpretability.
- The code converts all words to lower case to ensure 'TRUMP' and 'trump' are represented equally.

Feature Engineering

- We chose to extract Word Count as a feature to help us determine the category of the articles.
- Only the Top 30 frequently occurring words of each category were used to assist classifying this data.

Machine Learning Models

- We chose three Machine Learning models to analyze the feature-extracted data.
- Two, covered in class, are Naïve Bayes and Logistic Regression.
- We added Random Forest, too, due to our experience with this method from STA545.

Testing

```
 Train Accuracy:

Logistic Regression:
 0.6434101018999994
Random Forest:
 0.44463865891762794
Naive Bayes:
 0.4739655196936644

 Test Accuracy:
Logistic Regression Accuracy:
 0.6768135639639883
Random Forest Accuracy:
 0.47672774303558824
Naive Bayes Accuracy:
 0.4547989126898396

 Final Accuracy Results:

Logistic Regression Acc:
 0.67882345545
Random Forest Accuracy:
 0.5114951014266667
Naive Bayes Accuracy:
 0.47848484871
```

- We used the Machine Learning methods to perform tests and their accuracy was gauged.
- The results that were obtained are reported to the right.

Evaluation

- From the results, the best method for classifying this data was logistic regression with 0.68 accuracy.
- This value was not as high as we would have liked, further analysis and more Spark knowledge would be useful to boost this value.
- Random Forests performed with 0.51 accuracy, which is significant since a random guess at the category would be 0.25.
- Naïve Bayes performed worst with 0.48, not too far behind Random Forests classification.