Classifying Articles:

Project Overview:

To build a ML classifier for a JSON file containing articles to be recommended on Udacity Nanodegree programs.

Clustering the engineering Articles.

The Dataset:

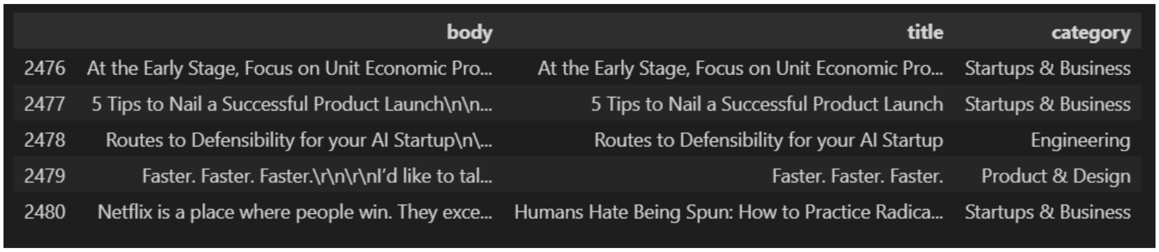
CSV file containing 2481 rows, each row resembling a different article.

Columns:

Body: Contains the body of the article.

Title: Contains the title.

Category: Contains the category.



	body	title	category
2476	At the Early Stage, Focus on Unit Economic Pro...	At the Early Stage, Focus on Unit Economic Pro...	Startups & Business
2477	5 Tips to Nail a Successful Product Launch\n\n...	5 Tips to Nail a Successful Product Launch	Startups & Business
2478	Routes to Defensibility for your AI Startup\n\...	Routes to Defensibility for your AI Startup	Engineering
2479	Faster. Faster. Faster.\r\n\r\nI'd like to tal...	Faster. Faster. Faster.	Product & Design
2480	Netflix is a place where people win. They exce...	Humans Hate Being Spun: How to Practice Radica...	Startups & Business

Data Wrangling:

Converting the title and body columns to lower case.

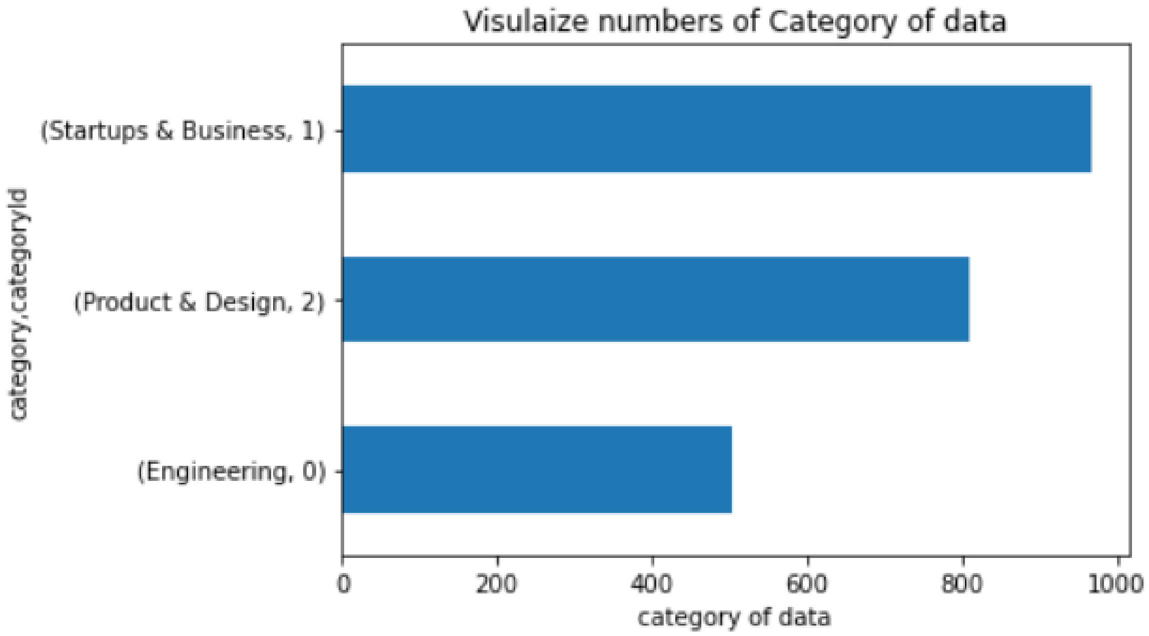
Found over 100 duplicated rows in the dataset and removed them.

Found 49 rows with no text inside and removed them.

Checked for anomalies in the articles, removed 100 articles consisting of those with less than 200 words (most are scrapping errors).

I've removed stop words (words like that are filler with no effect, such as "the" and such) and did lemmatization (converting words to their lemma, for example, best and better gets converted to good), this helps reduce the number of features (words) that the algorithm needs to test.

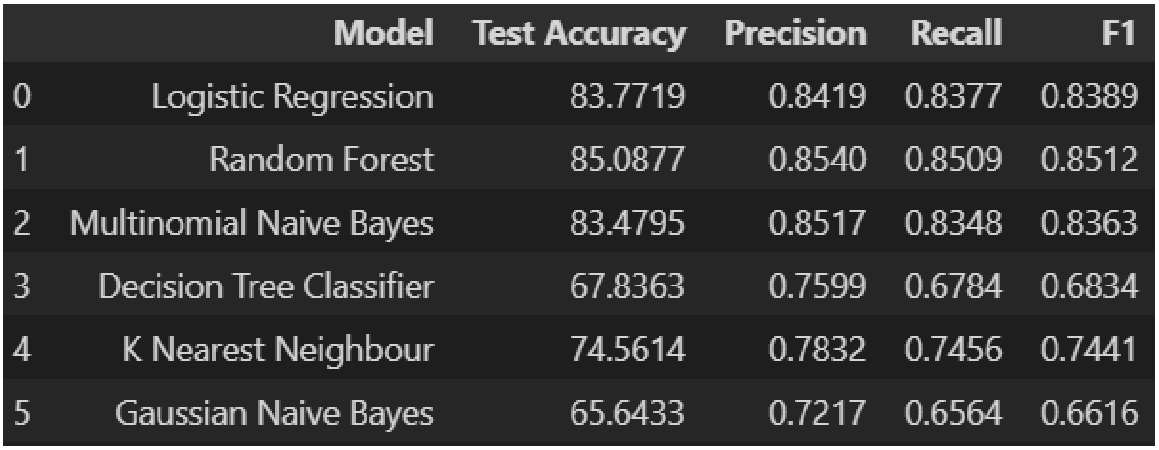
Distribution of Categories:



Modeling:

I’ve tested six different models and used the one with the highest accuracy as stated in the project requirements

Models:



	Model	Test Accuracy	Precision	Recall	F1
0	Logistic Regression	83.7719	0.8419	0.8377	0.8389
1	Random Forest	85.0877	0.8540	0.8509	0.8512
2	Multinomial Naive Bayes	83.4795	0.8517	0.8348	0.8363
3	Decision Tree Classifier	67.8363	0.7599	0.6784	0.6834
4	K Nearest Neighbour	74.5614	0.7832	0.7456	0.7441
5	Gaussian Naive Bayes	65.6433	0.7217	0.6564	0.6616

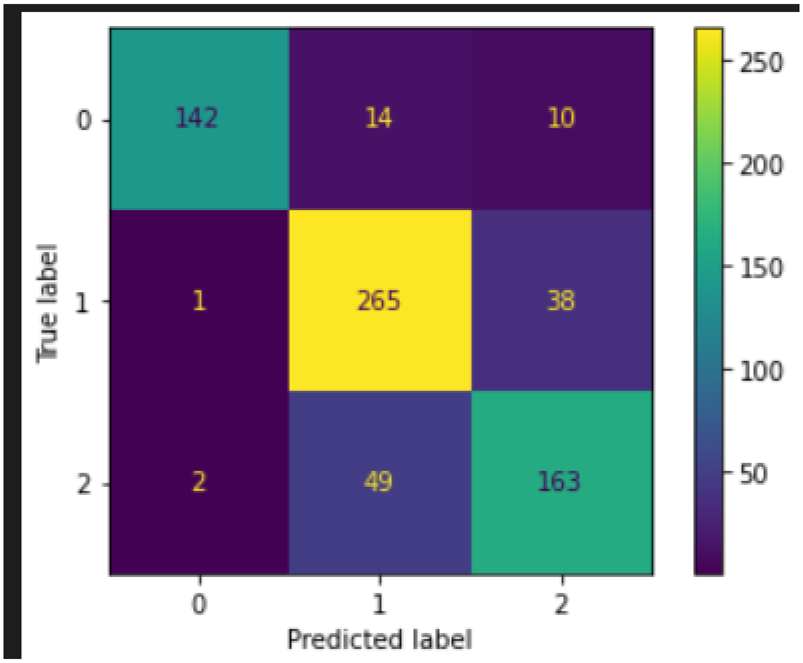
Random Forest was the one with the highest accuracy during testing, which makes sense given that random forest works well even with unbalanced features (articles words in this case).

Optimizing the Model:

I tried to increase the number of estimators (how many times the algorithm tries different “branches” or decisions), but it took 10 times as long and gave worse accuracy score during testing, so I’ll keep using the defaults.

Results and Conclusion:

Confusion matrix (which categories were mistakenly classified as what):



We can see from the confusion matrix that the most wrong prediction is between category 1 (Business) and 2 (Product), which makes sense given that they both have many articles and probably a lot of overlap in terms used.

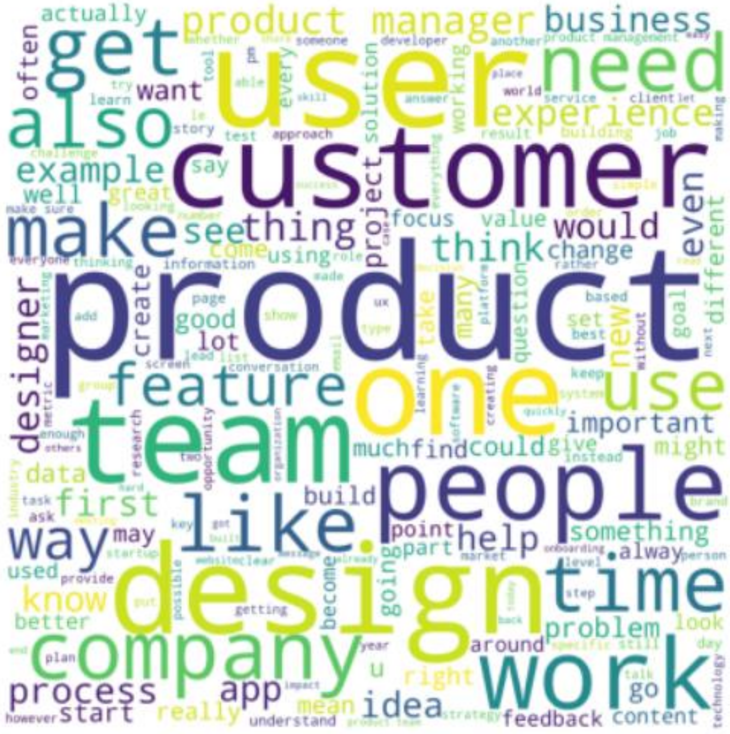
Which is what we see when we start visualizing most common words.

We can see that there is a lot of overlap between the three categories, like "Team", "User", and "Product".

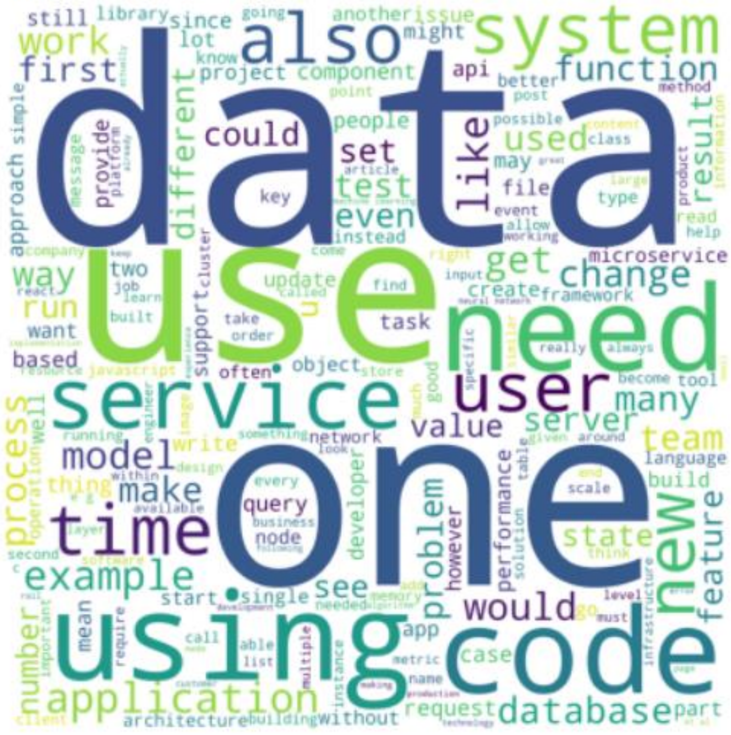
Which makes sense given that engineering makes products for businesses (the three categories we have).

WordCloud Visual:

Words common in product articles:



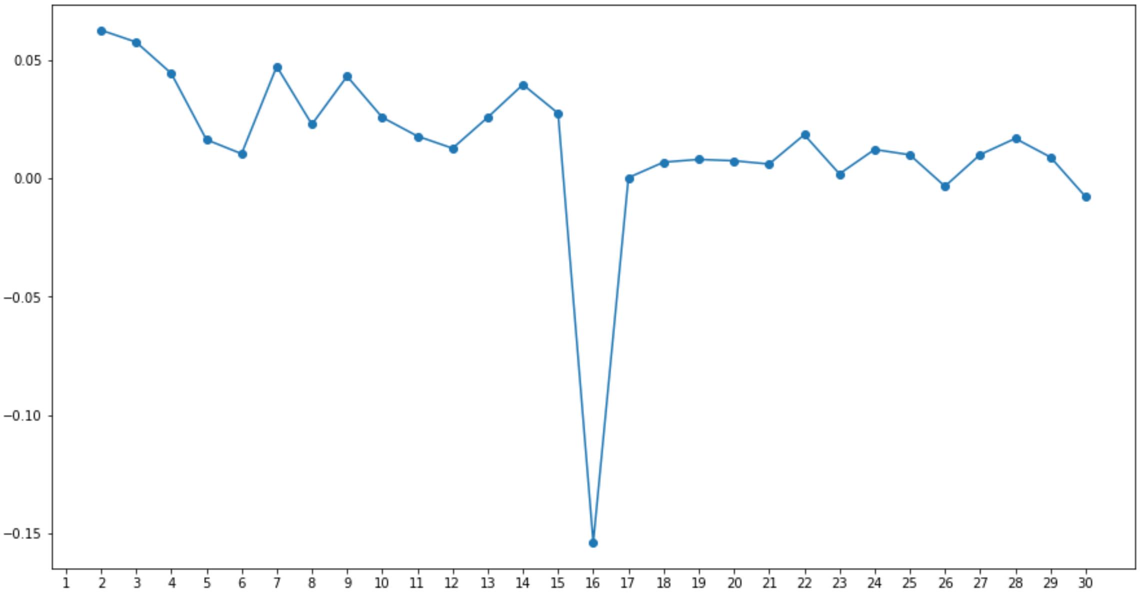
Words common in engineering articles:



Clustering:

I did clustering for the Engineering category based on title.

Calculating the Silhouette score (a score to determine the optimal number of clusters), it is said to be two clusters, but kept them to four given that’s what’s mentioned in the project



The count of articles in each cluster is (42, 393, 36, 33), we can see that most (around 80%) fall inside the second clusters, making the two clusters reasonable.