AMAZON ELECTRONICS REVIEW PREDICTION

Mythri Partha
Springboard Data Science Career Track

THE PROBLEM AND GOAL

- Amazon is the most popular online shopping platform with their own technology.
- They also have a great way to see if the product is good or not by using reviews.
- I want to create a model that can predict the star rating/whether a rating is positive or negative
- Goal: Find words that are important in predicting electronic reviews, create a model that can detect text patterns and see if a review is positive or negative

DATA ACQUISITION

- The data was sourced from data.world and Datafiniti (a database with several business, product, and property listings from all over the internet)
- The data was in 3 separate csv files with similar categories
- Data was found: https://data.world/datafiniti/consumer-reviews-of-amazon-products

DATA WRANGLING/CLEANING

- There were several columns with thousands of missing values, so I removed those columns altogether (e.g. whether the person bought the product, date updated, date seen, review ID, etc.).
- I then noticed that the categories of the products were not well defined so I used the text from the names of the product to categorize them more specifically. (e.g. Tablets, Alexa, Fire TV, chargers, etc.)
- I noticed all the products were electronics and they were all made by Amazon, so I removed the Manufacturer and brand columns as they are all the same.

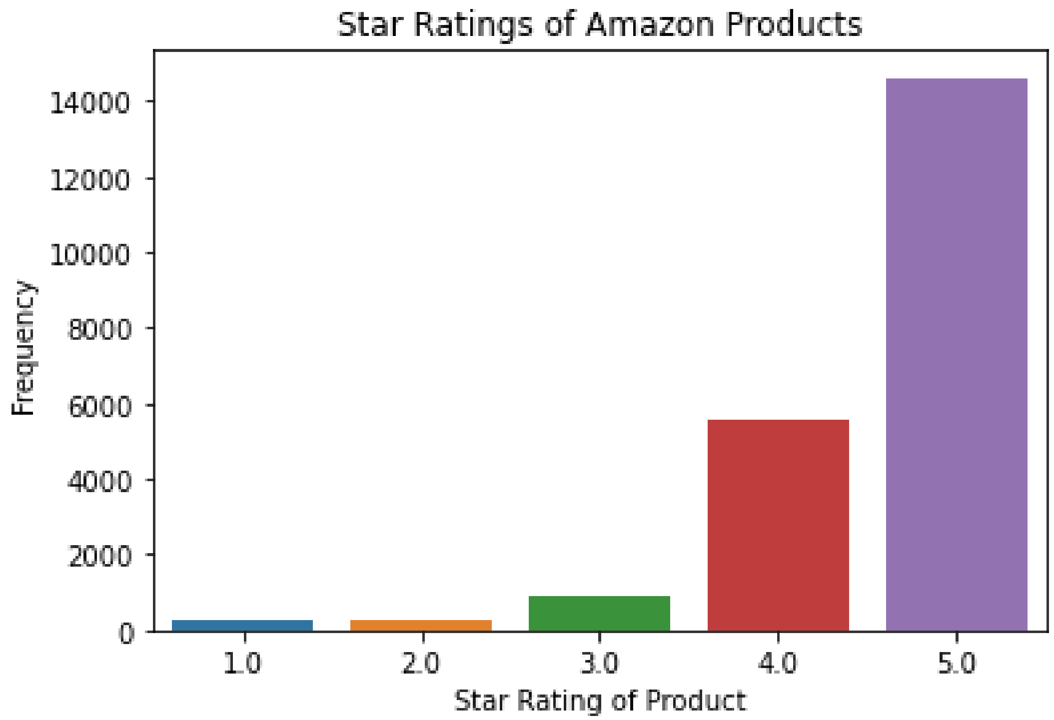
DATA WRANGLING PART 2

- I removed columns from all three dataframes where they did not match each other and reduced the data from 24 or 27 columns to about 12 columns
- After removing rows with empty data, we were left with 21625 rows and 12 columns (I further reduced the number of columns after EDA was completed)

EXPLORATORY DATA ANALYSIS

- I mainly observed patterns when it came to ratings and created visuals to observe how well the data was balanced and what words were most common in reviews.
- I looked at ratings in each category, which star ratings were most frequent in the dataset, which products were reviewed most, the star rating of the product versus the number of people who found the review helpful, ratings of the actual electronic products (not the cases), and I saw the words that occurred most frequently in 5 star reviews using a wordcloud.

VISUALS



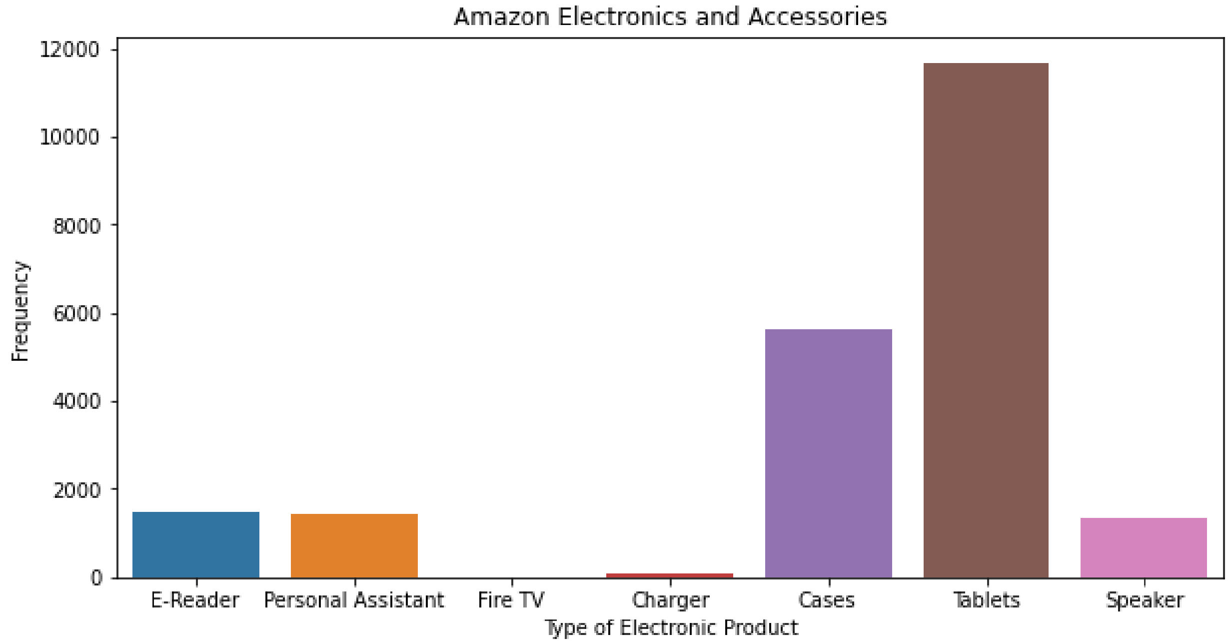
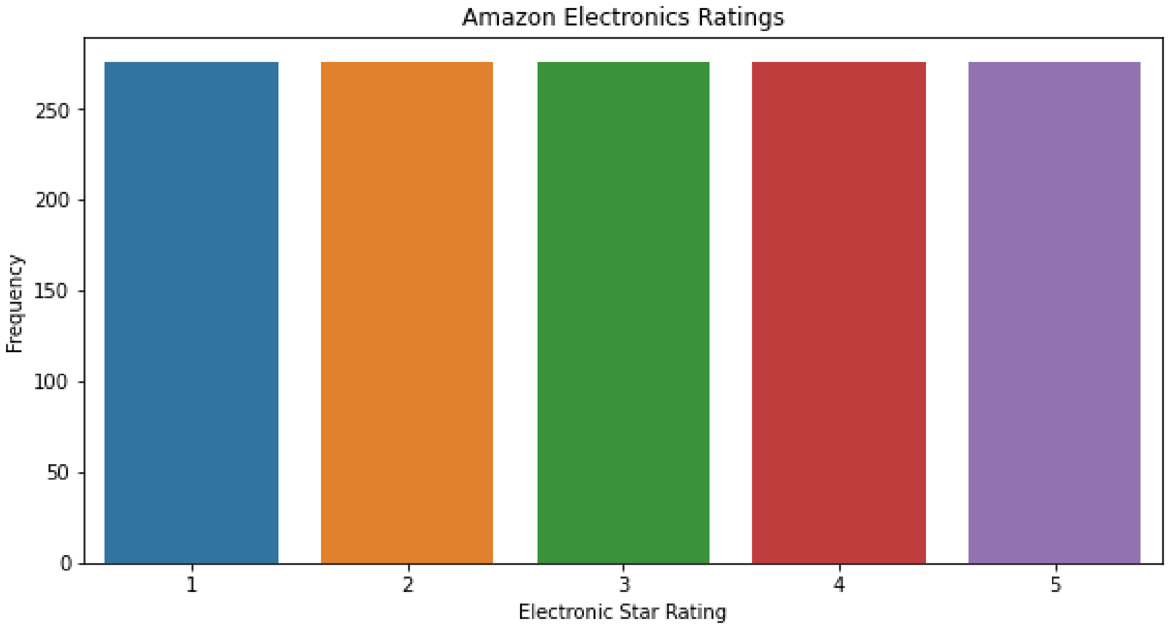


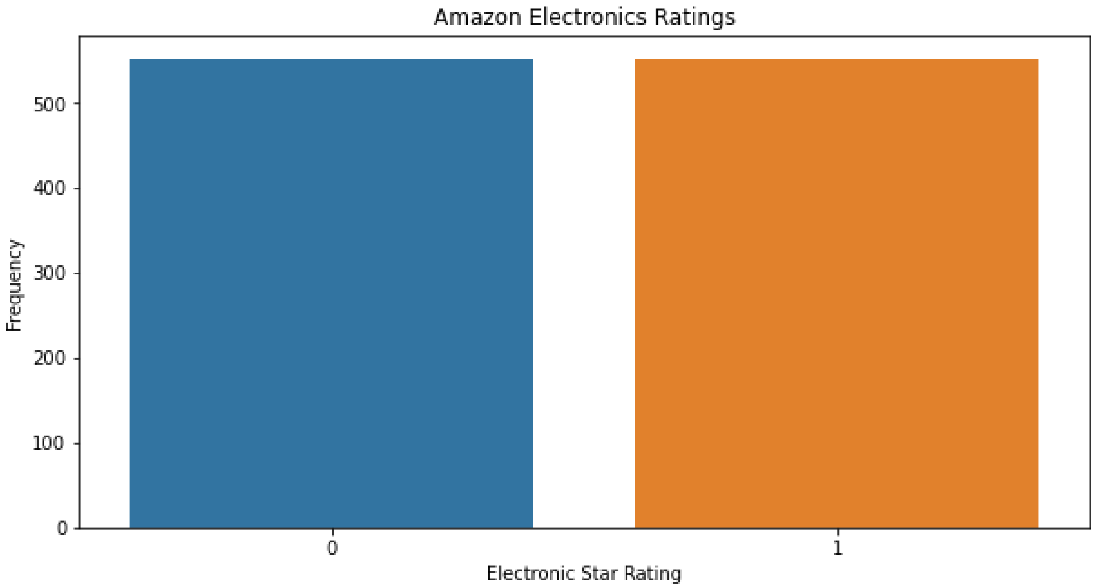
Figure 1: Star Ratings and the Electronics Data before resampling

VISUALS PART 2

Figure 2: WordCloud (Most Common words used in 5-star Ratings)

DATA AFTER RESAMPLING





Ratings distribution after resample (graph on left is star rating, graph on right denotes whether the rating is positive (1) or negative (0)

MODELING

Goals/Questions

- Create a model that helps predict the rating of a product from the review effectively
- Which model most effectively can predict the star ratings of products?
- What words are most important in predicting reviews?

MODELING PART 1

- I first started by completing a model (Linear SVC) on the raw data before resampling. Then I created a cross-validation table
- The accuracy is pretty high, but it is not a valid classification method as the data is very skewed in favor of 5-star reviews

	precision	recall	f1-score	support
1	0.53	0.86	0.65	69
2	0.53	0.74	0.61	69
3	0.70	0.73	0.72	221
4	0.72	0.76	0.74	1394
5	0.90	0.87	0.89	3654
accuracy			0.83	5407
macro avg	0.68	0.79	0.72	5407
weighted avg	0.84	0.83	0.83	5407

MODELING WITH RESAMPLE

- Then, I resampled the data and completed LinearSVC, Logistic Regression, Random Forest classification, and Gradient Boost, and K-Nearest Neighbors all models showed decent precision/recall/f1 for 1 and 2-star reviews, average ratings for 3 and 5-star ratings, and bad metrics for 4-star reviews

	precision	recall	f1-score	support
1	0.75	0.86	0.80	69
2	0.75	0.74	0.74	69
3	0.59	0.43	0.50	69
4	0.38	0.36	0.37	69
5	0.43	0.51	0.47	69
accuracy			0.58	345
macro avg	0.58	0.58	0.58	345
weighted avg	0.58	0.58	0.58	345

	precision	recall	f1-score	support
1	0.79	0.88	0.84	69
2	0.76	0.74	0.75	69
3	0.60	0.48	0.53	69
4	0.42	0.41	0.41	69
5	0.44	0.51	0.47	69
accuracy			0.60	345
macro avg	0.60	0.60	0.60	345
weighted avg	0.60	0.60	0.60	345

Left: Classification Matrix for Linear SVC
Right: Classification Matrix for Logistic Regression

MODELING WITH RESAMPLE PART 2

	precision	recall	f1-score	support
1	0.76	0.84	0.80	69
2	0.69	0.77	0.73	69
3	0.55	0.54	0.54	69
4	0.43	0.32	0.37	69
5	0.51	0.55	0.53	69
accuracy			0.60	345
macro avg	0.59	0.60	0.59	345
weighted avg	0.59	0.60	0.59	345

	precision	recall	f1-score	support
1	0.72	0.90	0.80	69
2	0.68	0.78	0.73	69
3	0.58	0.46	0.52	69
4	0.45	0.36	0.40	69
5	0.53	0.54	0.53	69
accuracy			0.61	345
macro avg	0.59	0.61	0.60	345
weighted avg	0.59	0.61	0.60	345

Left: Random Forest Classifier Classification Report
Right: Gradient Boost Classifier Classification Report

MODELING WITH RESAMPLE PART 3

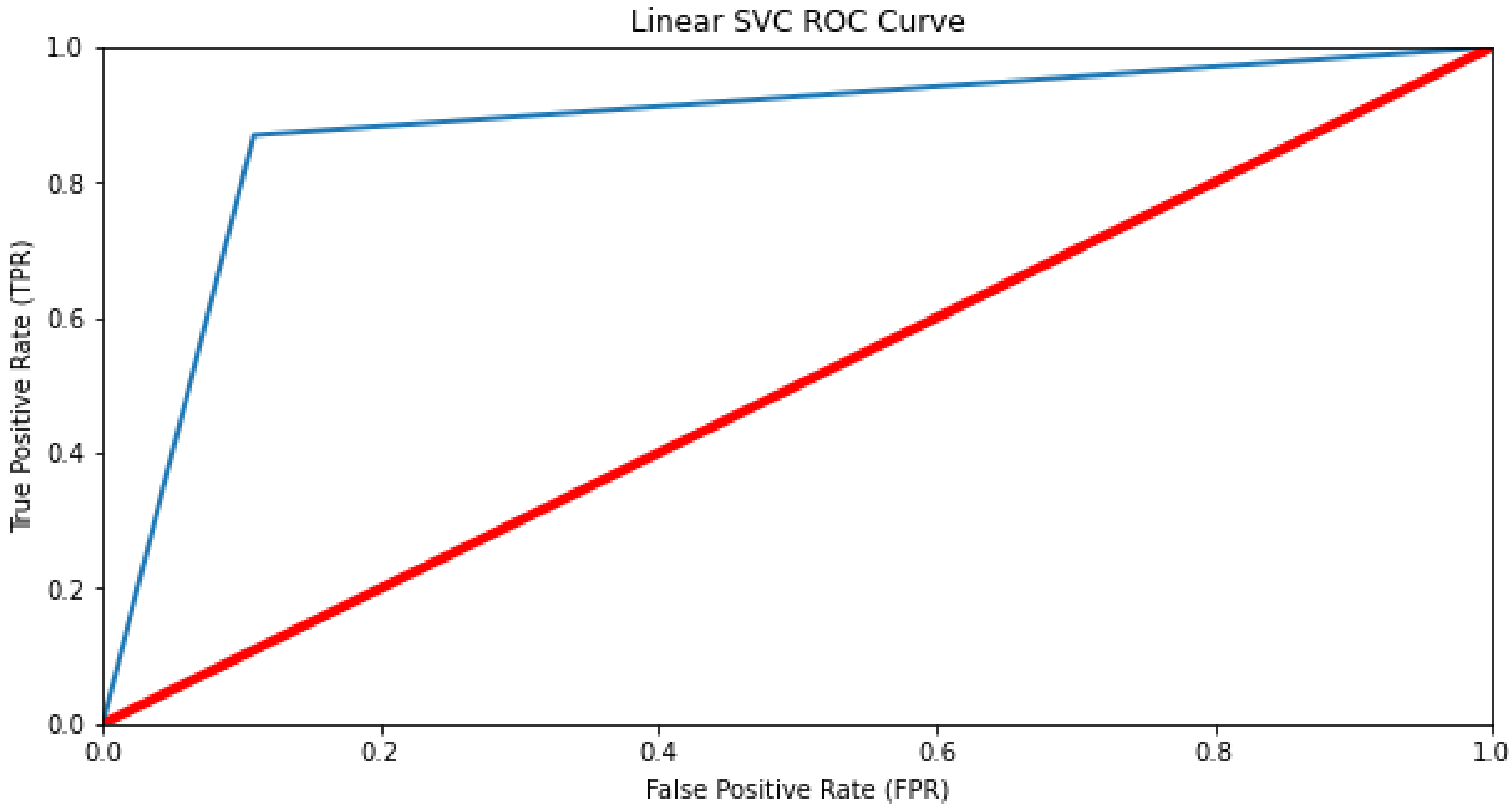
	precision	recall	f1-score	support
1	0.87	0.80	0.83	69
2	0.72	0.74	0.73	69
3	0.93	0.36	0.52	69
4	0.55	0.09	0.15	69
5	0.34	0.86	0.49	69
accuracy			0.57	345
macro avg	0.68	0.57	0.54	345
weighted avg	0.68	0.57	0.54	345

K-Nearest Neighbors Classification Report

MODELING WITH RESAMPLE (BINARY)

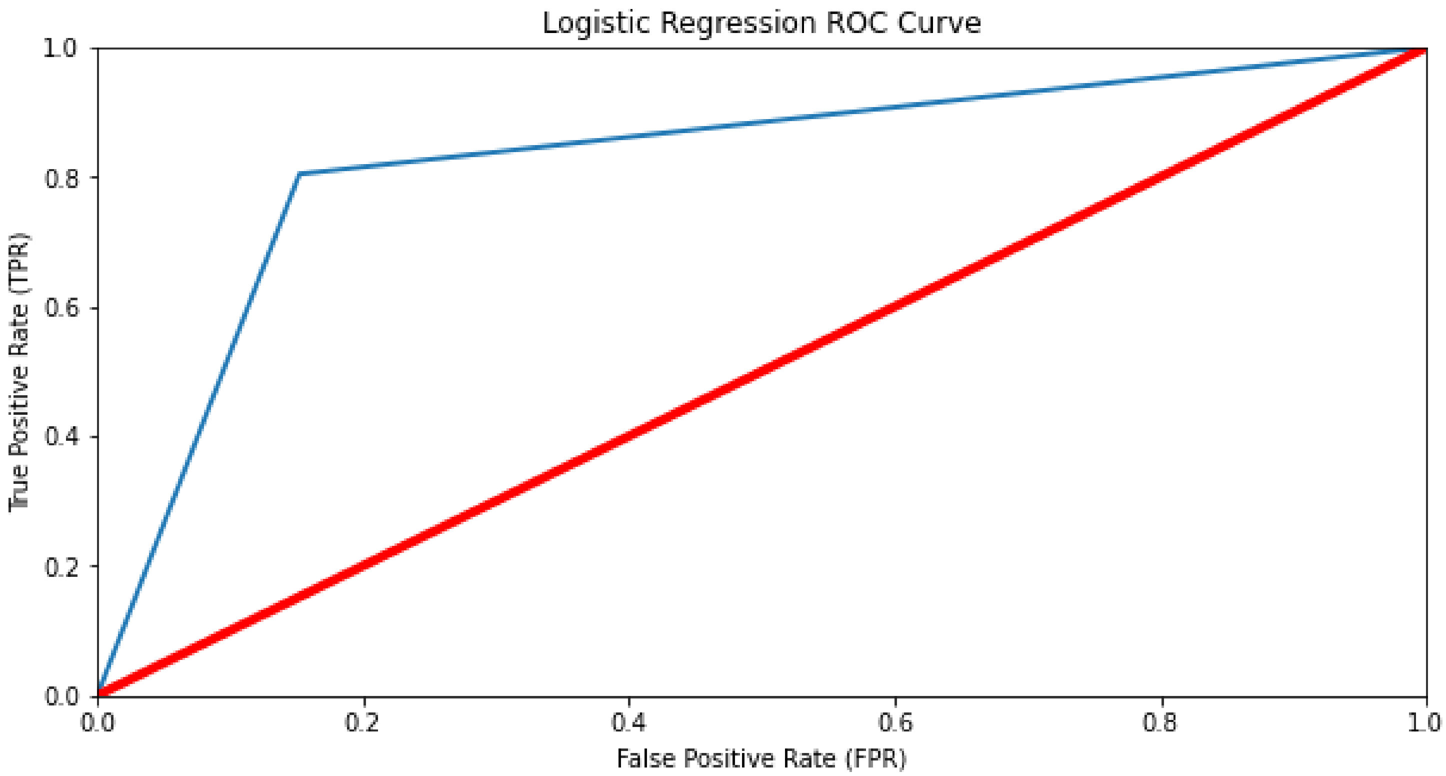
- To further improve model metrics, I turned this into a binary classification problem where 1 and 2-star reviews would be considered negative (denoted with a 0) and 4 and 5-star reviews would be considered positive (denoted with a 1), 3-star ratings were dropped completely
- I performed LinearSVC, Logistic Regression, Gradient Boost, and Random Forest Classification Models and found their classification reports and ROC-AUC curves.

CLASSIFICATION REPORT AND ROC-AUC CURVE FOR BINARY LINEAR SVC



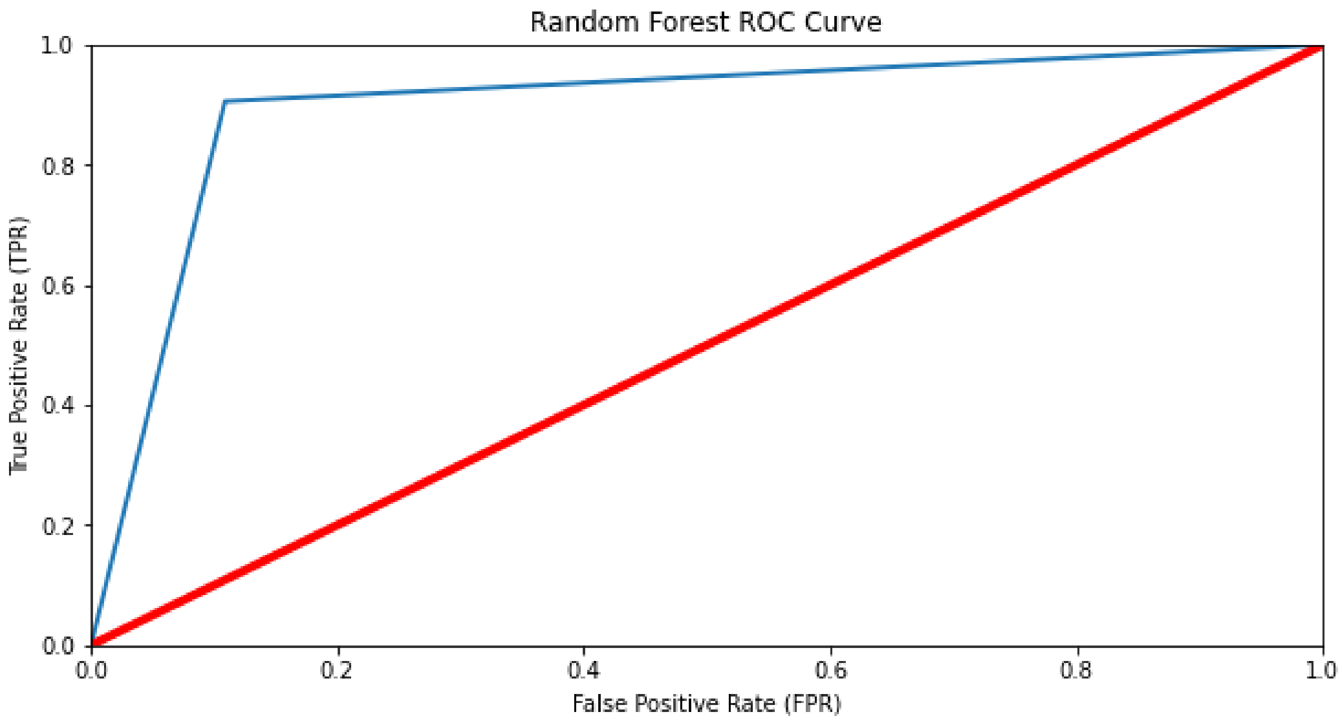
	precision	recall	f1-score	support
0	0.87	0.89	0.88	138
1	0.89	0.87	0.88	138
accuracy			0.88	276
macro avg	0.88	0.88	0.88	276
weighted avg	0.88	0.88	0.88	276

CLASSIFICATION REPORT AND ROC-AUC CURVE FOR BINARY LOGISTIC REGRESSION



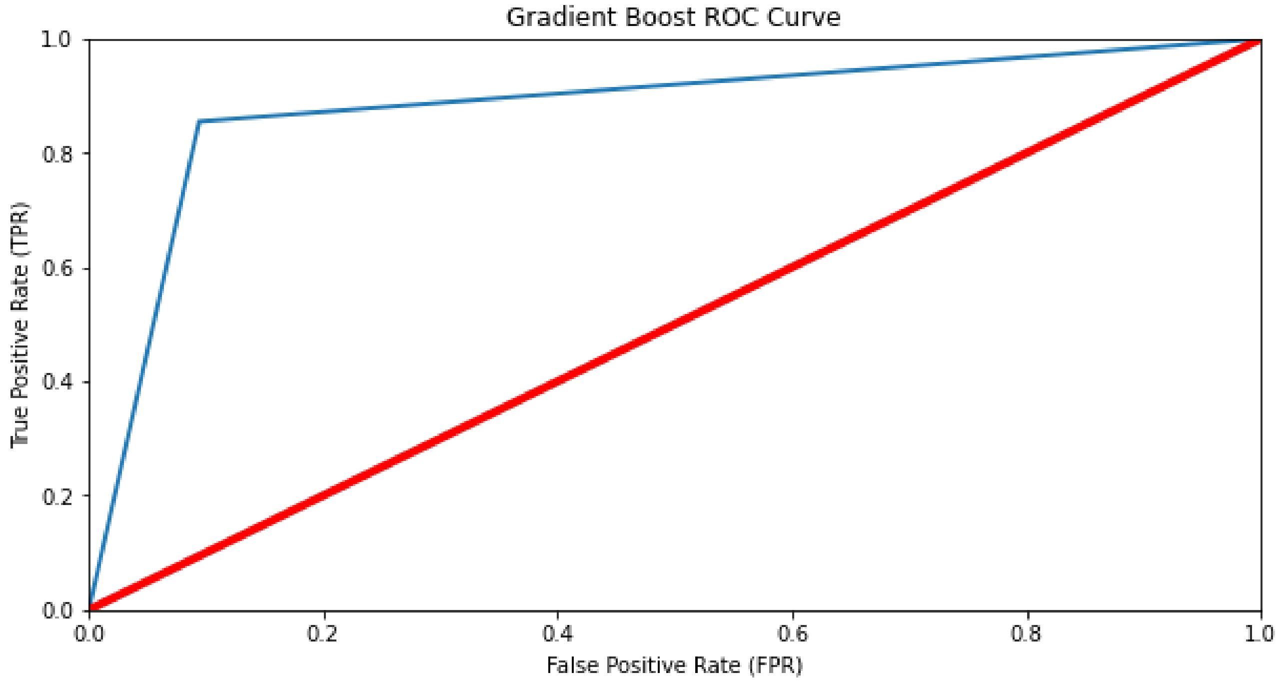
	precision	recall	f1-score	support
0	0.81	0.85	0.83	138
1	0.84	0.80	0.82	138
accuracy			0.83	276
macro avg	0.83	0.83	0.83	276
weighted avg	0.83	0.83	0.83	276

CLASSIFICATION REPORT AND ROC-AUC CURVE FOR BINARY RANDOM FOREST CLASSIFIER



	precision	recall	f1-score	support
0	0.90	0.89	0.90	138
1	0.89	0.91	0.90	138
accuracy			0.90	276
macro avg	0.90	0.90	0.90	276
weighted avg	0.90	0.90	0.90	276

CLASSIFICATION REPORT AND ROC-AUC CURVE FOR BINARY GRADIENT BOOSTING CLASSIFIER



	precision	recall	f1-score	support
0	0.86	0.91	0.88	138
1	0.90	0.86	0.88	138
accuracy			0.88	276
macro avg	0.88	0.88	0.88	276
weighted avg	0.88	0.88	0.88	276

FEATURE IMPORTANCE/ENGINEERING

Goal: I want to see which words were instrumental in predicting product reviews.

I utilized Normalized Count Occurrence and TF-IDF to complete this.

FEATURE ENGINEERING WITH NORMALIZED COUNT OCCURRENCE

	Word	Count
1379	it	0.493197
2634	the	0.328798
1053	for	0.164399
124	adjust	0.164399
2953	would	0.164399
2632	that	0.164399
195	and	0.164399
675	customer	0.164399
2683	to	0.164399
2333	service	0.164399
143	after	0.164399
2930	with	0.164399
394	brightness	0.164399
1810	online	0.164399
638	couldnt	0.082199
327	before	0.082199

FEATURE ENGINEERING WITH TF-IDF

	Word	Count
124	adjust	0.278263
394	brightness	0.261223
2333	service	0.215054
675	customer	0.204752
1810	online	0.202964
1379	it	0.191434
143	after	0.159196
2913	whitepaper	0.153696
10	120	0.153696
866	elmer	0.153696
2465	speaking	0.153696
1650	minuets	0.153696
882	english	0.153696
1309	identical	0.153696
1091	fud	0.153696
1009	female	0.153696

IMPROVEMENTS AND LIMITATIONS

- When working with classification of a star rating from 1-5, the prediction scores were not strong, this can be improved by changing my train_test_split, adding more reviews using a method like GPT, and more features for feature selection.
- I could try using the type of electronic as a feature in the models as well
- Some more text-preprocessing and checking the spelling of words might improve the model and how it was transformed by TF-IDF
- I could also perform sentiment analysis on top of what was completed

CONCLUSION AND FUTURE WORK

- The best models to carry on with are random forest and gradient boost classifiers for binary classification as they have the highest precision.
- This project could be further carried on by using a recommendation system and some neural networks.
- From here, the data is formatted into an app where we can input a review and the program outputs whether or not it is positive.