Brooke Ann Coco
Project 4 Summary
Feedback Classifier
May 28, 2019

Overview:
Whether we are working for a company that sells products or selling our own value to a potential employer, everyone is a salesperson. One of the best ways to ensure we have the best product on the market is by utilizing user feedback. To this end, the goal of this project is to utilize Natural Language Processing (NLP) in conjunction with unsupervised learning to classify consumer feedback by topic, enabling developers and designers to streamline the optimization of their product. Ideally, this model would be used to supplement user research teams by not only confirming human findings, but also capturing insights that the team may have missed, overlooked, or deemed unimportant.

Tools:

Platform	Purpose	Tool
Python	Web Scraping	BeautifulSoup4
	Cleaning	Pandas
		Re
		TextBlob
		NLTK
	Tokenize	NLTK
	Modeling	Scikit-learn
	Visualization	Matplotlib
	App Development	Flask
Microsoft Office	Presentation	Powerpoint

Data:
The scope of this pilot study focuses on the online product reviews regarding a Point-of-Sale system (POS) named Aloha. POS systems are central to functioning of many restaurants where their roles vary from payroll and labor management to sales, inventory, and customer analytics. Product review data was obtained via scraping G2 and Capterra, two popular consumer review websites, where I obtained reviews from nearly 200 individuals.

Modeling:
Before modeling, I performed several standard pre-processing techniques to prepare the data for analysis. This included converting all capital letters to lowercase, eliminating numerical digits, removing urls, expanding common contractions, deleting unnecessary punctuation, correcting spelling errors checking (via TextBlox), stemming the data (via LancasterStemmer), lemmatizing the data (via WordNetLemmatizer), then, finally, removing standard stopwords.

Next, I tokenized the data by sentence (via NLTK), then I vectorized the using count vectorization, yielding over 800 tokens for analysis. However, upon the realization that the model was loading on common, but semantically neutral words, I decided to vectorize my data with frequency-inverse document frequency (TF-IDF) instead. From there, I reduced the dimensionality of my data first via non-negative matrix factorization (NMF), then latent semantic analysis (LSA). Because I was unable to discern much difference in performance between the two, I focused most of my time trying to optimize the NMF model. From there, the data was normalized, then clustered it via K-Means clustering.

Because the aim of this project was topic modeling, I did not utilize any objective scoring metric to evaluate model performance. Instead, model evaluations were made based on a subjective basis, where I manually inspected the results to determine if the clustering made sense. After reviewing the model results, adjustments were made to the pre-processing pipeline. Most of my time was spent adjusting my preprocessing pipeline, rather than testing other modeling methods. For instance, some modeling iterations loaded heavily on trivial terms, such as 'aloha', 'pos', or 'system,' all of which give us little information as to what the review is about. Conversely, other iterations made nontrivial distinctions between analogous terms, such as 'customer service' and 'customer support'. Keeping to the popular data science mantra "garbage in, garbage out," I felt that I would make bigger gains in my model by adjusting the pre-processing pipeline rather than employing more complex modeling techniques, especially given that my input data was messy and unstructured. Only in parsing through the data myself could I begin to understand how the model reacts to specific data.

Conclusions:

The best model resulted from vectorizing via TF-IDF, reducing dimensionality via NMF, then clustering the result into five clusters via K-mean clustering. The table below depicts the model's resulting clusters as well as an example of a review that was found within each cluster.

Topic	Example
Customer Service	"Their customer support is dreadful."
User-friendly	"It is user friendly for servers and management."
Ease of Use	"Ease with front-of-the-house interaction and training."
Cost/ Price	"There are probably better and less expensive solutions on the market for small operators."
Miscellaneous	"Slightly outdated in my opinion."

The model performed reasonably well, though not as well as I would have hoped. There are several reasons for this, many of which are addressed below. Although the scope of this project only allowed me to focus on one specific POS system, given more time, this algorithm has the potential to scale not only to other POS systems, but to any number of products ranging from iPhones to hospitals.

Deliverables:

With the resulting model, I was able to put together a working prototype of what my resulting product would look like using Flask (see Appendix A). The program would enable users to submit transcripts from interviews, focus groups, usability tests, diary studies, and

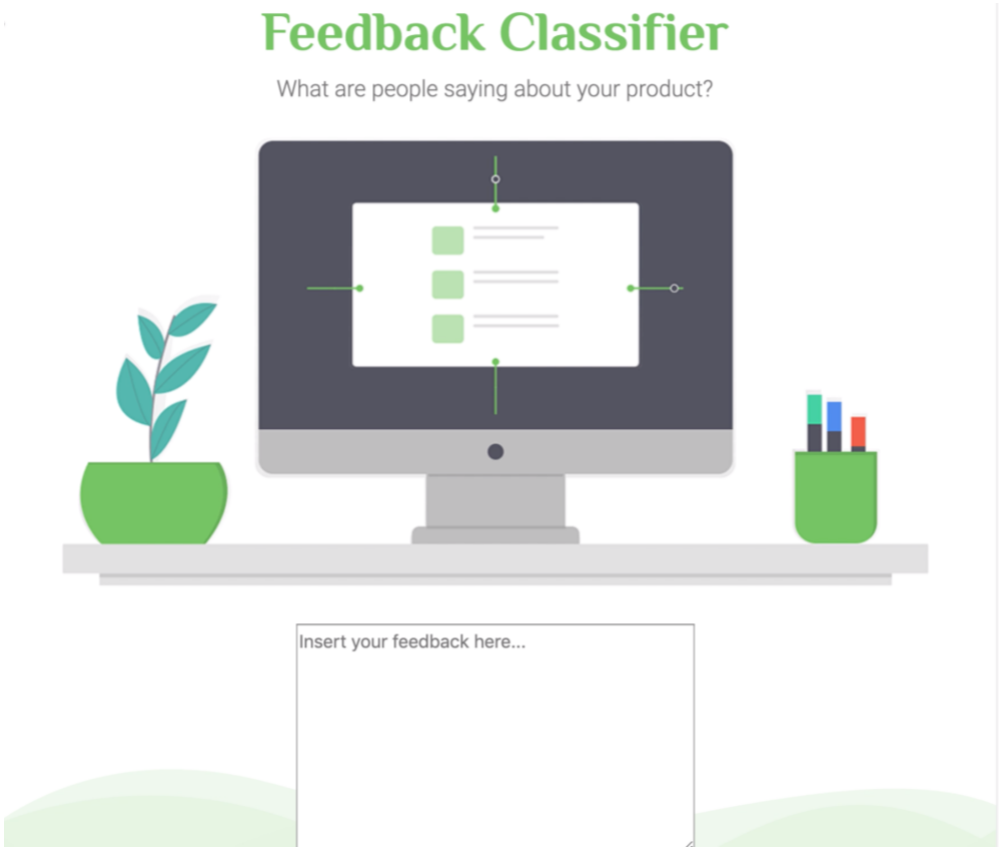
participatory design workshops, etc., leaving the model to cluster the data, then group it with its corresponding cluster. The product is intended to supplement user research teams in analyzing user research data, enabling them to give quick and actionable insights towards optimizing their products.

Future Improvements:

There are several improvements I would like to make to my model as I move forward.

- Originally, I performed a sentiment analysis on my data before feeding it into my model. However, because was so poor at accurately dividing positive from negative reviews, ended up re-consolidating my data for modeling. Many product review sites prompt user reviews with questions such as, "What do you like best about this product?" or "Where could this product be improved?" Given more time, I would like to parse the online reviews based on the question that prompt them. From there, I would like to experiment with sentiment analysis further on my data.
- As I mentioned previously, the model had a difficult time connecting semantically similar words, like "customer support" and "customer service". For this reason, I would like to test the utility of word embeddings within my model.
- The data had a fair bit of noise given that every sentence did not contain relevant information about the product under review. Consider the following token, "I suppose anyone who has ever worked in the restaurant can understand how embarrassing that is." Further improvements on the model would play round with how the data is tokenized. For instance, perhaps each review should be its own document.
- As of now, my app prototype displays the clustered data after it has been processed. With more time, I would like the app to display the original, unprocessed reviews.
- Finally, I would like to add reviews from other POS systems and train a corpus that would be able to generalize to any POS system. Eventually, I would like this product to scale to any type of product.

Appendix A:



Category 1

1.the only thing that i like about it is that i have some familiarity with the system.
2.i've had of my brand new terminals replaced after they went bad and corrupted my whole system.
3.alpha has created more problems than it is solved.
4.alpha is pretty user friendly for staff like that i can add modified to help encourage up-selling.
5.was frustration to have our entire database copied from a location that has been open for almost years.
6.so it takes a really long time to report.
7.took a long time to work out all the errors learn how to do all the programming.
8.But i can usually figure things out.
9.really expensive for help desk calls so only reach out when absolutely necessary.
10.frustration when get a programme that does not know the answers to my questions have to get charged extra while they figure it out.
11.expensive can be difficult to program without help able to ring guests up quickly start tabes if necessary.
12.modified venus help encourage staff to up-sell.
13.security controls help keep staff from dismounting/come'ing things they should not have access to or defending cash.
14.from the back end to the front of house and all the cloud products we use it is very easy to use and works well for us.
15.our biggest challenge is the system not being chip card ready.
16.i know it is in the works .
17.they're an a+ company and should meet your needs.