Rebeca R. Mahr
May, 2021

Text-Based Advertisement Feedback Topic Modeling

Springboard Data Science Bootcamp Capstone 2 Project

Table of Contents

Background

Qualitative feedback in the form of text-based responses from an advertising campaign's target audience can be invaluable in determining ad message comprehension as well as in making decisions for future rounds of campaign flighting and creative development. However, human analysis of text-based responses requires reading through thousands of responses and manually coding them into themes or topics. This task can be very time-consuming and subjective to the reviewer. Additionally, in order to ensure reliability[1] in findings, the same text needs to be reviewed by multiple reviewers. Utilizing a machine learning algorithm to identify topics based on textual responses can decrease the amount of manual labor hours spent in analysis and minimize bias due to reviewer subjectivity. This type of algorithm can be utilized by market and advertising researchers, advertising campaign evaluators, and/or marketing analytics teams to supplement digital analytics reporting.

The field of data science dedicated to giving machines the ability to understand human language is known as Natural Language Processing (NLP) (Kedia & Rasu, 2020). For this project, NLP tools and models were used to analyze the frequency of words and generate topics from text-based ad feedback. This feedback was gathered in surveys distributed among a nicotine-vape-prevention campaign's target audience. Respondents were shown three video ads and asked what they thought the main message of each video was. These text-based ad feedback responses were analyzed for this project with the goal of identifying three to six key topics that summarize the message comprehension responses.

Data Overview

This project utilized survey data collected by Rescue Agency Public Benefit LLC. via the Qualtrics online survey platform. The data was exported, processed, and de-identified to protect respondent privacy. To

[1] Reliability refers to an index of the consistency of a measuring instrument in repeatedly providing the same score for a given respondent. (Graziano & Raulin, 2010)

further ensure respondent and client privacy and for compliance with Institutional Review Board (IRB) protocol requirements, the data is not publicly available and ad names and regional information have been encoded. To give readers visibility into the data structure, a sample dataset is available here.

The survey data was gathered from two annual evaluation surveys that were distributed among a nicotine-vape-prevention campaign's target audience across two US states. As part of the surveys, respondents were shown three video ads that were flighted prior to the campaign evaluation. In response to each video ad shown in the surveys, respondents were asked to explain what they thought the main message of the video ad was. Within the surveys, respondents were also asked to share demographic information including their age, race, gender, and region. All demographic information was reviewed at the aggregate level to ensure respondent privacy.

This survey data was split into two datasets. The first is a stacked, response-level, dataset with the text responses and the encoded video ad that each text response corresponds to. The second is a flat, respondent-level, dataset with demographic information.

Data Dictionary

- Dataset 1 – Ad_Feedback_Text (n=1,448):
    - ID: Unique identifier for respondent
    - Text: Includes text responses
    - Ad: Indicates the specific video advertisement the response was for (encoded)
        - DD: ad message related to vape companies deceiving teens
        - DF: ad message related to vapes making smokers vulnerable to viruses
        - ST: ad message related to exposing the chemicals in vapes
- Dataset 2 – Ad_Feedback_Demos (n=724):
    - ID: Unique identifier for respondent
    - CalculatedAge: Respondent age
    - Race: Respondent self-reported race
    - Gender: Respondent self-reported gender
    - Segment: Audience segment the respondent belongs to (encoded)
    - Region: State where survey was distributed (encoded)
    - Urban_Rural: Indicates whether respondent region is urban or rural based on population density of the city they live in

Data Wrangling

The data wrangling step of this project included loading and inspecting the datasets, merging the datasets, and an initial cleaning of the text data.

The demographics dataset contained columns that could create confusion, so the following edits were made prior to merging the datasets:

- To avoid confusion as to what the age was calculated from, "CalculatedAge" was renamed to "Age"
- To ensure a tidy dataset for processing as well as avoid confusion about column values, binary/dichotomous variables were dummy encoded [2].

[2] Dummy encoding refers to the process of creating one column for each value of a categorical variable where a row gets a value of 1 if the row contains a value for that category and 0 if not (Hale, 2018).

- "Gender" was dummy encoded to "Female" and "Male" was dropped
- "Urban_Rural" was dummy encoded to "Urban" and "Rural" was dropped
- "Region" was dummy encoded to "Region_1" and "Region_2" was dropped

The demographics dataset was then merged into the ad feedback text dataset using a left join on the "ID" column to maintain the stacked level of the ad feedback text dataset.

In order for a text corpus to be utilized effectively in a NLP machine learning algorithm, the text must be standardized and stripped of any characters that are not text (for the English language) (Mysiak, 2019). The ad feedback text required the following to be "clean" for NLP:

- Contractions were expanded to ensure words retained their meaning after removing additional characters
- All non-alphanumeric characters were removed (including punctuation)
- All text was standardized in lowercase
- All gibberish / nonsense words were removed using Gibberish-Detector[3] (this step is particularly important with survey text responses)

View Data Wrangling Code

Preprocessing & Feature Engineering

While the preprocessing step is typically conducted after the exploratory data analysis (EDA) step in the data science process, NLP does require preprocessing of text data in order to explore it. For this project, some initial preprocessing was conducted before EDA. Then, final preprocessing and feature engineering was conducted based on findings from the EDA prior to modeling.

Removal of Stop Words

Stop words are words that occur frequently in the English language (in this case), but that do not provide additional meaning to text such as pronouns and connector words that are used to construct a sentence (Kedia & Rasu, 2020). If not removed, stop words result in topic models that are comprised of mostly useless words due to their frequency in the text corpus. Several Python NLP packages include lists of stop words from pretrained language models. For this project, a combination of stop words from the Gensim[4], spaCy[5], and WordCloud[6] Python libraries were removed from the ad feedback text as recommended in Kamil Mysiak's article about preprocessing text data (Mysiak, 2019).

Tokenization

Tokenization is the process of splitting a text corpus "into chunks called tokens. Each token carries a semantic meaning associated with it" (Kedia & Rasu, 2020). Tokenization of the ad feedback text was conducted using the spaCy Tokenizer for its robust approach of developing a syntactic tree for each sentence in a text corpus (Malhotra, 2018). Considering a majority of the ad feedback text responses were only comprised of one sentence per response, within-sentence tokenization was critical to modeling this particular text.

[3] Gibberish-Detector is a Python program used to detect gibberish in text (rrenaud, 2015).
[4] 'Gensim is a Python library for topic modelling, document indexing and similarity retrieval with large corpora.' (RaRe-Technologies/gensim, n.d.)
[5] 'spaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python.' (spaCy 101, n.d.)
[6] WordCloud is package for generating word clouds in Python. (Mueller, 2020)

Lemmatization

Lemmatization takes tokenization a step further by converting tokenized words into their base form for analysis (Kedia & Rasu, 2020). SpaCy's lemma_ method was used for lemmatization due to its ability to detect the part-of-speech tag by default in lemmatization (Prabhakaran, n.d.).

TF-IDF Vectorizer

The final text corpus input into the NLP topic modeling algorithm must be transformed into a vectorizer object with a column for each word/token in the text corpus and rows with a frequency value for the word in the corresponding column in each document (or ad feedback response in this case). In order to account for the importance of certain frequent and infrequent words and weight them accordingly, the term frequency-inverse document frequency (TF-IDF) vectorization approach was used. The TF-IDF vectorization approach follows the following formula for weighting terms:

$$TF(w) = \frac{Number\ of\ times\ the\ word\ w\ occurs\ in\ a\ document}{Total\ number\ of\ words\ in\ the\ document}$$

$$IDF(w) = log\frac{Total\ number\ of\ documents}{Number\ of\ documents\ containing\ word\ w}$$

$$weight(w,d) = TF(w,d) \times IDF(w)$$ [7]

For this project, the TF-IDF vectorizer was developed using scikit-learn's feature_extraction.text.TfidfVectorizer (Pedregosa et. al, 2011) with the parameters max_df set to .95, min_df set to 2, and use_idf set to True. The ad feedback text corpus included 750 unique words, so excluding words that occurred in more that 95% (max_df) of ad feedback responses would only exclude the word "vape" if not weighted to less than 95% by the inverse document frequency. Because each ad feedback response typically included just one sentence, only words that occurred 1 time (min_df) in the ad feedback text corpus were excluded.

Train-Test Split

Because the ad feedback text was not pre-labeled with a topic, the topic modeling algorithms were considered to be unsupervised learning algorithms. Typically, splitting data into training and test sets is not required for unsupervised learning because there are no labels to evaluate the model's performance. However, because hyper-parameter tuning was used for topic modeling, the data was split into training and test sets to ensure that "unseen" data was not used in optimization. The split was conducted using scikit-learn's model_select.train_test_split (Pedregosa, 2011) with 30% of the ad feedback text data reserved in the test set.

View Preprocessing and Feature Engineering Code

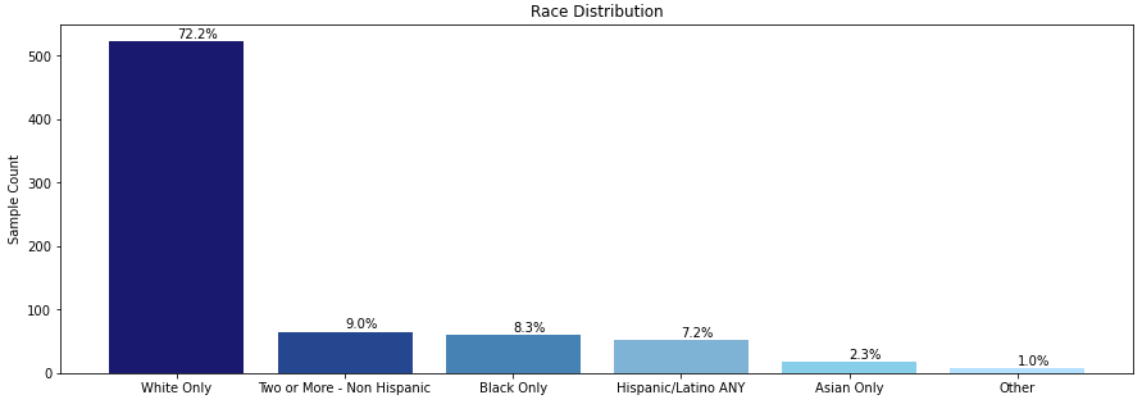
Exploratory Data Analysis (EDA)

EDA of Sample Demographics

[7] TF-IDF formula (Kedia & Rasu, 2020, pp. 85)

Although not a primary objective of this project, the sample demographic distributions were reviewed during EDA to provide context into who provided the ad feedback and to inform whether there were limitations to the models developed due to sampling.

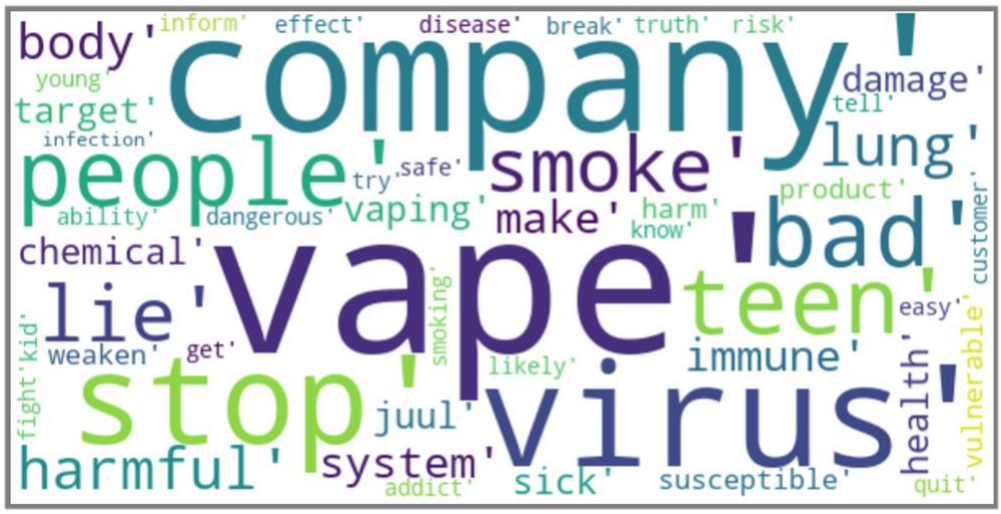
The survey sample included 724 teenagers ranging in age from 13 to 19 years old and about 17 years of age on average. The sample distribution was fairly skewed in terms of gender and race/ethnicity with 64% having self-identified as female and 72% having identified their race as white. The full race distribution is shown in the figure below. Clearly, Black, Hispanic, Asian and other races were under-represented compared to the national distribution. However, this distribution did more closely reflect the race/ethnicity distribution within the two states surveyed.



EDA of Ad Feedback Text

To explore the ad feedback text, a term frequency analysis was conducted. The purpose of the term frequency analysis was two-fold. First, it provided a glimpse into possible topics that might emerge from the topic models. Second, and more importantly, it identified high frequency words that might not provide additional context to topics and need to be removed prior to modeling. The term frequency analysis was conducted using nltk's[8] probability.FreqDist class. A word cloud of the top 50 words in the ad feedback text corpus was developed using WordCloud (Mueller, 2020) as shown in the image below. The top 8 most frequently used words included "vape" at 31%, "company" at 5%, and "virus", "stop", "people", "bad", "teen", and "smoke" at about 4% each.

[8] "The Natural Language Toolkit (NLTK) is an open source Python library for Natural Language Processing (Bird, Klein, & Loper, 2009)."



The word "vape" clearly outweighed other words which is logical considering the campaign is for a vape-prevention messaging brand and that theme is common across all of the ads. Based on this finding, "vape" was considered as a word to remove from modeling. Some of the other top words including "company", "lie", "teen" could be related to the DD ad while top words like "virus", "bad", "stop" could be related to the DF and ST ads.

View EDA Code

Modeling & Hyperparameter Tuning

A total of four topic models were developed and evaluated including two Latent Dirichlet Allocation (LDA) topic models and two Nonnegative Matrix Factorization (NMF) topic models.

LDA Topic Models

LDA is a "generative probabilistic" unsupervised machine learning model used for topic modeling in natural language processing (Blei, Ng, & Jordan, 2003). LDA assumes that the distribution of the topics that the LDA algorithm aims to uncover in a text corpus and the distribution of the words in each of those topics are Dirichlet distributions (Sharma, 2020). For this project, scikit-learn's decomposition.LatentDirichletAllocation was used for developing the LDA topic models (Pedregosa, 2011). The LDA topic models were evaluated based on the distribution of topics by ad, the distance and size of topics per pyLDAvis's Intertopic Distance Map[9], and their perplexity score[10].

Initial LDA Topic Model

Input: TF-IDF vectorizer object (training set)
Parameters (otherwise default):

- n_components: 6 topics – assuming two topics could be identified per ad

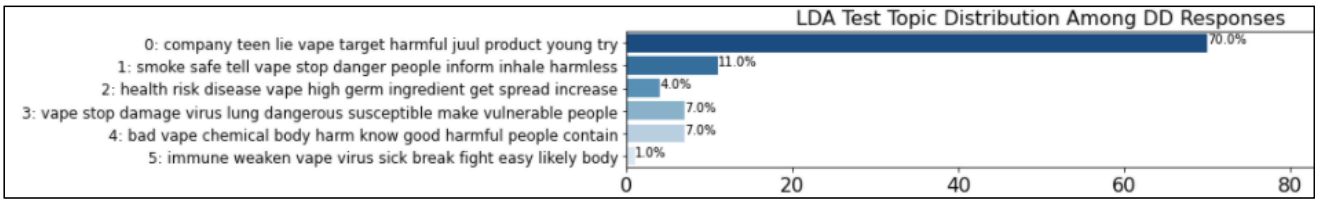
[9] PyLDAvis "Transforms the topic model distributions and related corpus data into the data structures needed for the visualization" (Topic Models (e.g. LDA) visualization using D3, 2015).

[10] "Perplexity is a statistical measure of how well a probability model predicts a sample. As applied to LDA, for a given value of n, you estimate the LDA model. Then given the theoretical word distributions represented by the topics, compare that to the actual topic mixtures, or distribution of words in your documents (Topic modeling, 2021)."

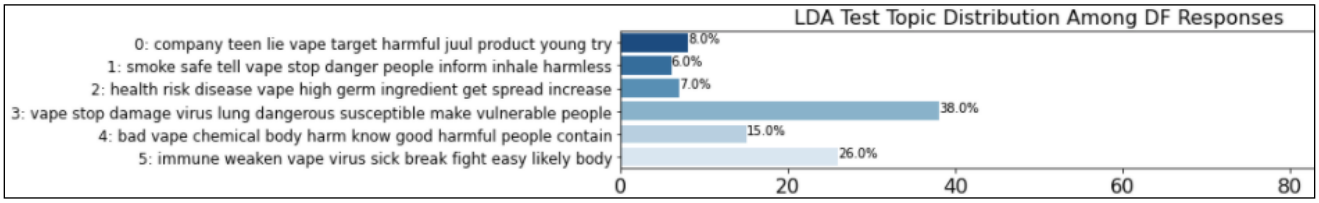
- max_iter: 250 iterations – based on the number of unique word in the training set
- learning_method: online Bayes method for speed of processing

Results (test set):

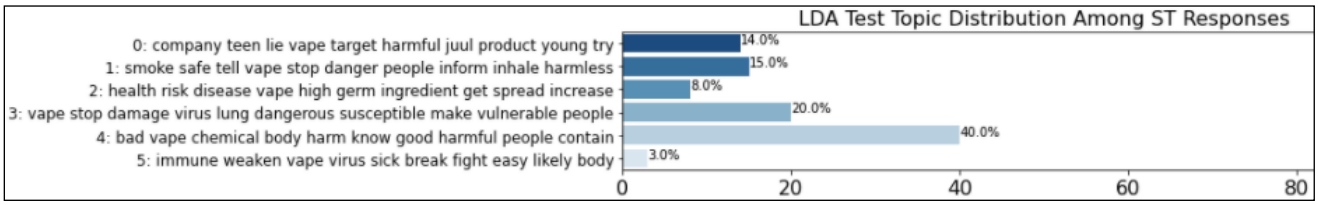
- Topic Distribution by Ad



The DD ad message is related to vape companies deceiving teens, so topic 0 was the most relevant. This was a strong mapping with 70% of DD responses having the highest LDA topic probability for topic 0.



The DF ad message is related to vapes making smokers vulnerable to viruses, so topics 2, 3 and 5 were the most relevant. The mapping for topic 2 was weak, but a total of 64% of DF responses had the highest LDA probability for topics 3 and 5.



The ST ad message is related to exposing chemicals in vapes, so topic 4 was the most relevant. Mapping for ST responses had the most mixed result across topics. This may indicate a weak model or challenges with message comprehension.

- Visualization of Initial LDA Topic Model

The intertopic distance map also showed good initial results with each topic having a large marginal topic distribution and each topic being well distanced from the others.

- Perplexity Score of Initial LDA Topic Model:
  1124.55
  Without another model to compare to, it was unclear whether the perplexity score of 1124.55 is strong.

Hyperparameter Tuning of LDA Topic Model

The randomized search hyperparameter tuning approach was utilized to identify hyperparameters that would result in an optimal LDA topic model for the TF-IDF vectorizer text corpus. For efficient use of computational resources, only the n_components (number of topics) and max_iter (maximum number of iterations) hyperparameters were optimized. Scikit-learn's model_selection.RandomizedSearchCV (Pedregosa, 2011) was used to conduct the randomized search.

Randomized Search Settings & Results

- Hyperparameter grid:
  - n_components: 3-12 to test options of 1 topic per ad to 4 topics per ad
  - max_iter: 50-500 with increments of 50 iterations
- Other Parameters (otherwise default):
  - Iterations: 50 to search 50% of parameter space
    - (50 iterations / (10 n_components options * 10 max_iter options)
  - cv: 5-fold cross-validation
- Results:
  - n_components: 3 topics
  - max_iter: 450 iterations

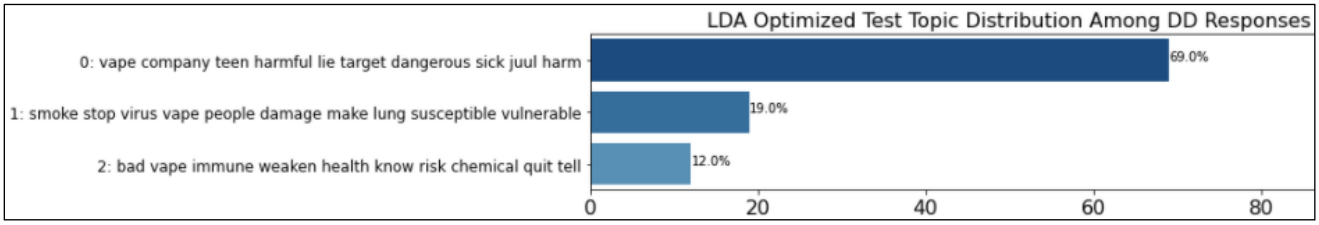
Optimized LDA Topic Model Based on Randomized Search Results

Input: TF-IDF vectorizer object (training set)
Parameters (otherwise default):

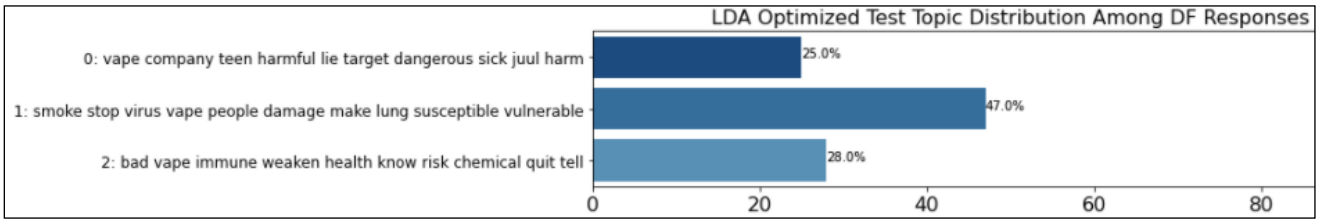
- n_components: 3
- max_iter: 450
- learning_method: online Bayes method for speed of processing

Results (test set):

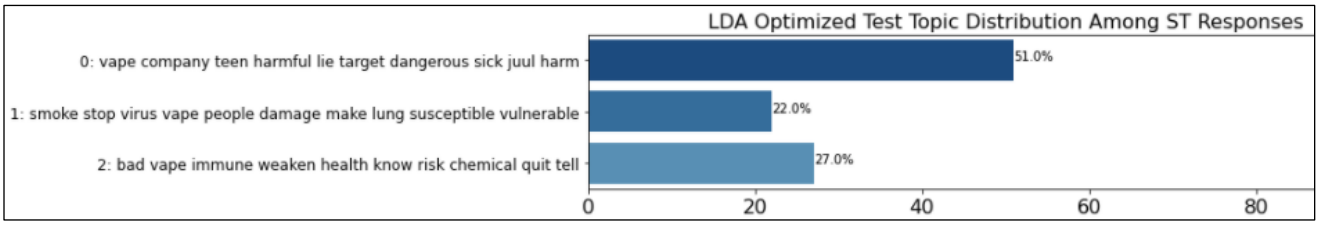
- Topic Distribution by Ad



The DD ad message is related to vape companies deceiving teens, so topic 0 was the most relevant. Similar to the initial model, this was a strong mapping with 69% of DD responses having the highest LDA topic probability for topic 0.

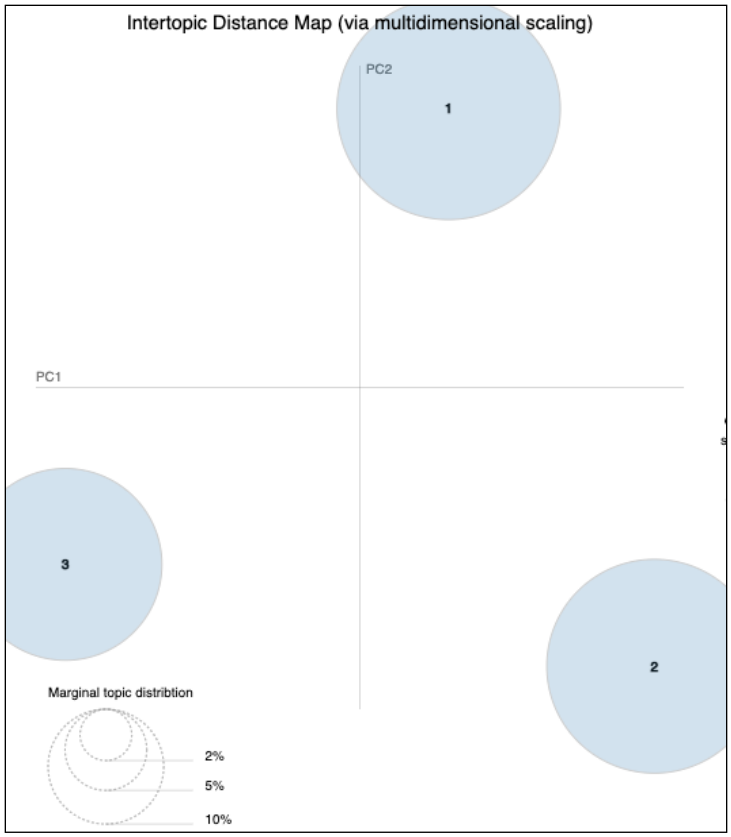


The DF ad message is related to vapes making smokers vulnerable to viruses, so topics 1 and 2 were most relevant. This mapping was also strong with a total of 75% of DF responses having the highest LDA topic probability for topics 1 and 2.



The ST ad message is related to exposing chemicals in vapes, so topic 2 was likely the most relevant, although topic 1 could be viewed as relevant as well since the message implies that vape companies are hiding the dangers of the chemicals in vapes. This mapping is not as clear as those for DD and DF.

- Visualization of Optimized LDA Topic Model



The intertopic distance map shows improvement over the initial LDA model with a larger marginal topic distribution for each topic and more distance between each topic.

- Perplexity Score of Optimized LDA Model:
  551.04
  The perplexity score for the optimized LDA model showed substantial improvement with a 51% decrease from the initial LDA model (551.04 vs. 1124.55).

NMF Models

NMF is an unsupervised machine learning model which "decomposes (or factorizes) high-dimensional vectors into a lower-dimensional representation. These lower-dimensional vectors are non-negative which also means their coefficients are non-negative" (Salgado, 2020). For this project, scikit-learn's decomposition.NMF was used for developing the NMF topic models (Pedregosa, 2011). The NMF topic models were evaluated solely based on the distribution of topics by ad due to limited options for model evaluation.

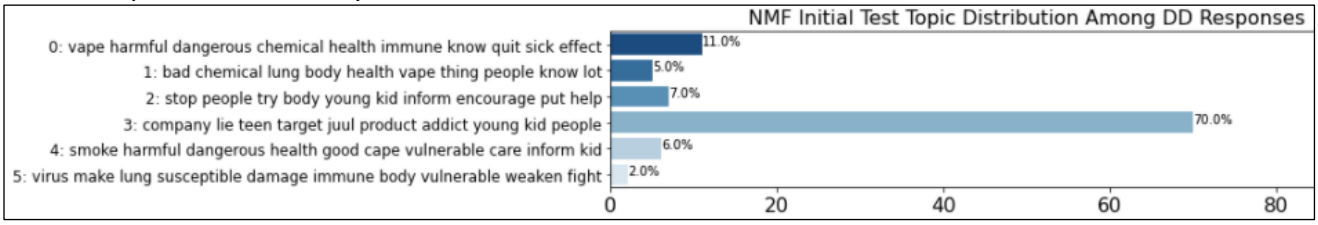
Initial NMF Topic Model

Input: TF-IDF vectorizer object (training set)
Parameters (otherwise default):

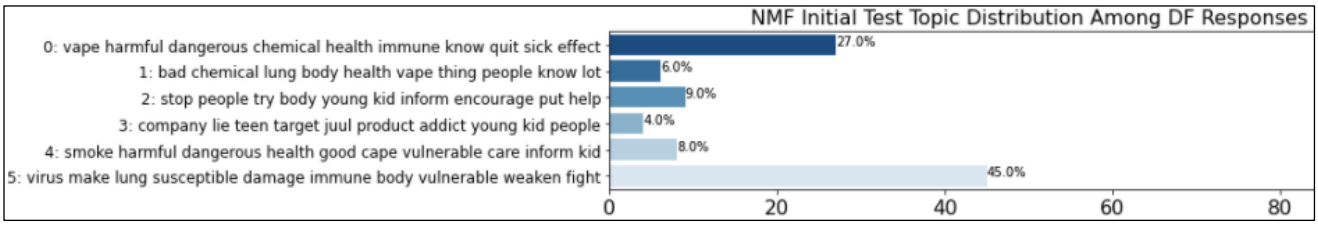
- n_components: 6 topics – for consistency with initial LDA topic model
- max_iter: 250 iterations – for consistency with initial LDA topic model

Results (test set):

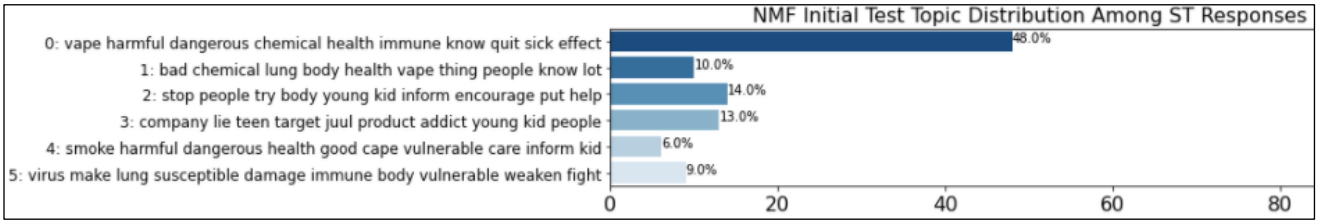
- Topic Distribution by Ad



The DD ad message is related to vape companies deceiving teens, so topic 3 was the most relevant. Similar to the LDA model, this was a strong mapping with 70% of DD responses having the highest NMF topic probability for topic 3.



The DF ad message is related to vapes making smokers vulnerable to viruses, so topics 0 and 5 were the most relevant. This was also a strong mapping with a total of 72% of DF responses having the highest NMF topic probability for topics 0 and 5.



The ST ad message is related to exposing chemicals in vapes, so topics 0 and 1 were the most relevant. Although a large percentage of responses to ST had the highest NMF topic probability for topic 0, only 10% did for topic 1. It's possible these topics could be merged into just one topic in an optimized model.

Hyperparameter Tuning of NMF Topic Model

For consistency between models for comparison, the same randomized search hyperparameter tuning approach was utilized to identify hyperparameters that would result in an optimal NMF topic model for the TF-IDF vectorizer text corpus.

Randomized Search Settings & Results

- Hyperparameter grid:
  - n_components: 3-12 for consistency with LDA hyperparameter grid
  - max_iter: 50-500 with increments of 50 iterations
- Other parameters (otherwise default):
  - Iterations: 50 to search 50% of parameter space
    - (50 iterations / (10 n_components options * 10 max_iteration options)
  - cv: 5-fold cross-validation
- Results:
  - n_components: 3 topics
  - max_iter: 450 iterations

Optimized NMF Topic Model Based on Randomized Search Results

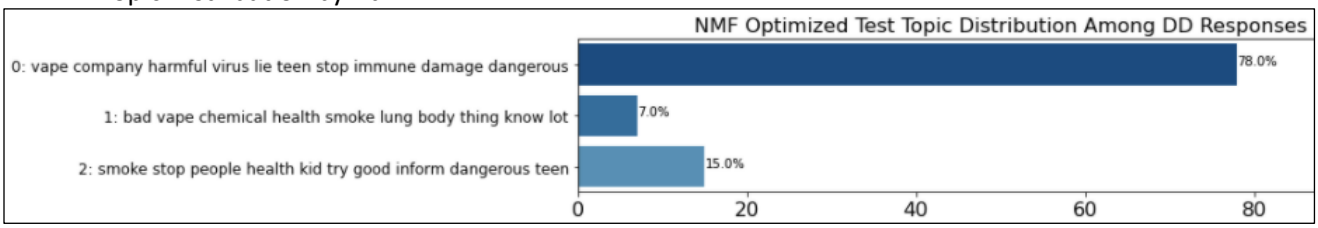
Input: TF-IDF vectorizer object (training set)

Parameters (otherwise default):

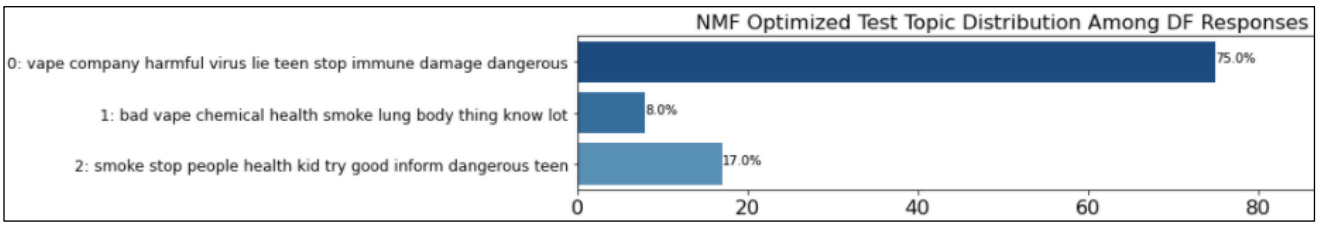
- n_components: 3
- max_iter: 450

Results (test set):

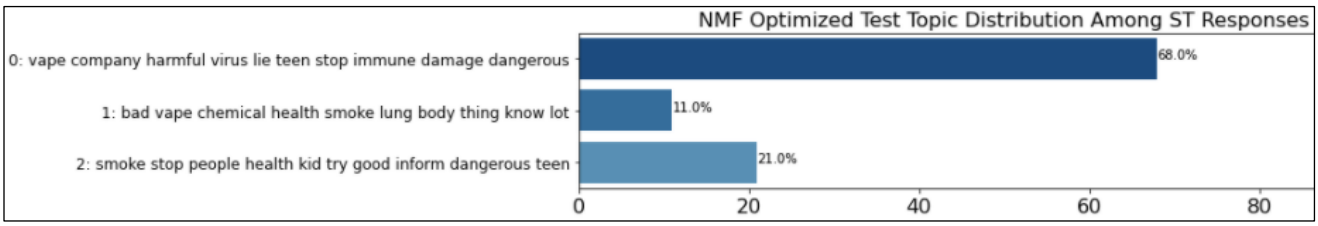
- Topic Distribution by Ad



The DD ad message is related to vape companies deceiving teens, so topic 0 was the most relevant. This is the strongest mapping so far with 78% of DD responses having the highest NMF topic probability for topic 0.



The DF ad message is related to vapes making smokers vulnerable to viruses, so topic 0 was the most relevant as well. This mapping is also strong with 75% of DF responses having the highest NFM topic probability for topic 0. However, the inclusion of the words, “company”, and “lie” in this topic might mean that this model joined too many distinct terms into a single topic.



The ST ad message is related to exposing chemicals in vapes, so topic 1 appeared to be the most relevant. Just over 10% of ST responses had the highest NMF topic probability for topic 1 and 68% of ST responses were mapped to topic 0. This shows that topic 0 dominated the topic distribution and the model may not be as strong as other models.

View Modeling Code

Model Selection & Conclusion

Based on the results of the initial and optimized LDA and NMF topic models, the optimal topic model for analysis of the ad feedback text was the **optimized LDA topic model** with 3 topics and 450 iterations. The topics generated by the optimized LDA topic model were:

0 vape company teen harmful lie target dangerous sick juul harm
1 smoke stop virus vape people damage make lung susceptible vulnerable
2 bad vape immune weaken health know risk chemical quit tell

The optimized LDA topic model mapped very well to 2 of the 3 ads. Over two-thirds (69%) of responses to DD (related to vape companies deceiving teens), had the highest LDA topic probability for topic 0. A total of 75% of responses to DF (related to vapes making smokers vulnerable to viruses)

had the highest LDA topic probability for topics 1 (47%) and 2 (28%). Mapping to ST (related to exposing the chemicals in vapes) was less clear with 51% of responses having the highest LDA topic probability for topic 0 although topic 2 (27%) might be more relevant. Results were similarly unclear for ST across all models, however.

The visualization of the optimized LDA topic model also showed two positive signs. First it showed that each of the three topics were well distanced from one another and further apart than the topics in the initial model. Second, the marginal topic distribution was larger for the three topics than for the topics in the initial model. The optimized LDA topic model also had a 51% improvement in the perplexity score from the initial LDA topic model.

Applying the Model to the Business Problem

Under the assumption that the topics generated by the optimized LDA topic model indeed reflect the topics of the ad feedback text responses, it can be inferred that ad message comprehension was higher for DD and DF while there may have been some confusion regarding the main message of ST. A next step to confirming these conclusions might be to manually review a small random subset of responses for each of the ads, ST in particular, where there was a "mismatch". If responses truly indicate a high level of confusion, likely the most cost-effective recommendation for clients and stakeholders would be to update just written ad copy in the ad video to clarify the main message without the need for video or audio edits prior to the next round of campaign flighting.

Limitations

1. Sample selection:
   The data utilized for this project was based on survey respondents who opted-into participating in research, so there is a level of sample selection bias.

2. Sample size:
   The entire ad feedback text corpus only included 1,448 responses from 724 respondents. A production-ready model would ideally have more ad feedback responses among a larger sample of respondents.

3. Sample distribution:
   As mentioned in the EDA section, the sample demographic distribution was disproportionately female and white. In order to draw conclusions that are generalizable to a campaign's target audience, the sample demographic distribution should reflect that of the campaign's target audience.

4. Subjectivity in topic interpretation:
   Although the machine learning model helped to eliminate reviewer subjectivity in identifying topics, a level of subjectivity was still present in the human interpretation of the word clusters identified in each topic generated by the model.

5. NMF topic model evaluation:
   Unlike the LDA topic models, the NMF topic models did not have a visualization of the topic space or a perplexity score to evaluate each model's performance.

6. Limited computational resources:
   Due to limited computational resources, the randomized search approach to hyperparameter tuning was utilized and only a limited number of hyperparameters were able to be optimized. With greater computational resources, utilizing the grid search approach would have resulted in a more thorough optimization.

References

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python.* O'Reilly Media Inc.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 993-1022.

Graziano, A. M., & Raulin, M. L. (2010). *Research Methods: A Process of Inquiry Ed. 7.* Boston: Pearson Education Inc.

Hale, J. (2018, 09 10). *Smarter Ways to Encode Categorical Data for Machine Learning*. Retrieved from Towards Data Science: Smarter Ways to Encode Categorical Data for Machine Learning

Kedia, A., & Rasu, M. (2020). Understanding the Basics of NLP. In *Hands on Python Natural Language Processing* (p. 7). Birmingham - Mumbai: Packt Publishing Ltd.

Malhotra, A. (2018, 06 24). *Introduction to Libraries of NLP in Python -- NLTK vs. spaCy*. Retrieved from Medium: https://medium.com/@akankshamalhotra24/introduction-to-libraries-of-nlp-in-python-nltk-vs-spacy-42d7b2f128f2#:~:text=NLTK%20is%20a%20string%20processing%20library.&text=As%20spaCy%20uses%20the%20latest,sentence%20tokenization%2C%20NLTK%20outperforms%20spa

Mueller, A. (2020). *WordCloud for Python Documentation*. Retrieved from GitHub.io: http://amueller.github.io/word_cloud/

Mysiak, K. (2019, 04 19). *NLP Part 2| Pre-Processing Text Data Using Python*. Retrieved from Towards Data Science: https://towardsdatascience.com/preprocessing-text-data-using-python-576206753c28

Pedregosa, e. (2011). Scikit-learn: Machine Learning in Python. *JMLR 12*, 2825-2830. Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html?highlight=tfidfvectorizer#sklearn.feature_extraction.text.TfidfVectorizer

Prabhakaran, S. (n.d.). *Lemmatization Approaches with Examples in Python*. Retrieved from Machine Learning Plus: https://www.machinelearningplus.com/nlp/lemmatization-examples-python/#spacylemmatization

RaRe-Technologies/gensim. (n.d.). Retrieved from github: https://github.com/RaRe-Technologies/gensim/#documentation

rrenaud. (2015). *Gibberish-Detector*. Retrieved from GitHub: https://github.com/rrenaud/Gibberish-Detector

Salgado, R. (2020, 04 16). *Topic Modeling Articles with NMF*. Retrieved from Towards Data Science: https://towardsdatascience.com/topic-modeling-articles-with-nmf-8c6b2a227a45#:~:text=Non%2DNegative%20Matrix%20Factorization%20(NMF,into%20a%20lower%2Ddimensional%20representation.

Sharma, A. K. (2020, 10 06). *Understanding Latent Dirichlet Allocation (LDA)*. Retrieved from Great Learning: https://www.mygreatlearning.com/blog/understanding-latent-dirichlet-allocation/

spaCy 101. (n.d.). Retrieved from spacy.io: https://spacy.io/usage/spacy-101

Topic modeling. (2021). Retrieved from Computing for the Social Sciences: UChicago: https://cfss.uchicago.edu/notes/topic-modeling/#perplexity

Topic Models (e.g. LDA) visualization using D3. (2015). Retrieved from pyLDAvis: https://pyldavis.readthedocs.io/en/latest/modules/API.html