Baruch COLLEGE | ZICKLIN SCHOOL OF BUSINESS

Course:

Applied Natural Language Processing

CIS 9665 – UWA [28285] – Fall 2021

Final Project Report

Natural Language Processing with Drug Review Dataset

Professor:

Chaoqun Deng

Authors:

Team 9

Aarif Munwar Jahan, Jacob Bayer, James Muehlemann,

John Makhijani, Juna Iafelice, Shawn Meng

December 13th, 2021

Table of Contents

Section A: Project Information

Background

Our aim was to conduct analysis that allowed us to determine how prescription drugs were rated by users based on the language of their reviews. To conduct this analysis, we used methods we learned in this class and other classes. Our analysis includes data cleaning and preprocessing, feature extraction and engineering, and data modeling. The following sections will go over the background of the data, the steps in our analysis, and the real-world implications the results of our analysis could have on the drug industry.

Dataset Description

The data we analyzed contains patient reviews on specific drugs along with related conditions and a 10-star patient rating system reflecting overall patient satisfaction. The data was retrieved by crawling online pharmaceutical sites and later published in a study on sentiment analysis of drug experience over multiple facts including effectiveness and side effects of these drugs. The dataset consists of six relevant attributes:

Field Name	Description	Datatype
drugName	Name of drug	String
condition	Name of condition	String
review	Patient review	String
rating	Patient Rating out of 10	Integer
date	Date of review entry	Datefield
usefulCount	Number of users who found the review useful	Integer

Section B: Research Information

Research Question

Can we use the text of drug reviews to predict whether the review is highly negative or not highly negative?

Research Motivation

Our first motivation was to create a back-end technology to enable a platform to conduct open-source sentiment analysis in order to automate review scoring. By applying a front-end to our analysis that asks a user for a review, the analysis we did would allow the user to know what this review would be rated on a number scale.

Our second motivation would be an improved version of this tool that could convert a textual data type into a numerical type by taking the textual review, identifying its sentiment, and assigning it a numerical rating. This is part of a long-term vision but is not included in our current implementation.

Section C: NLP Methods

Data Preprocessing

Before we did our analysis, we first had to deal with the missing data for which there were 59 records in our subset of 10,000. Since all of the cases of missing data in this dataset were drugs that didn't fall under a condition category, we defined these cases as conditions of their own (named after their drug names) in order to avoid them being grouped together under the same condition. This data was still usable, but for this small number of cases we couldn’t engineer features based on condition later on.

Before classifying the review text data, we first had to pass it through a function that cleaned and prepared it for analysis. This cleaning process involved lower-casing all the words, tokenizing them, and then lemmatizing using the Lancaster stemmer. Finally, numbers, punctuation, and English stop words were removed before being used in the analysis.

Analytical Methods

Based on our research question, the primary analysis in this project uses text classification method to classify raw text reviews into negative or not negative reviews.

As a point of further research, the project explores the topic modeling method to discover latent semantics within the raw text reviews.

See Appendix D for analytical method definitions.

Evaluation Metrics

The primary metric we will be using to evaluate our models will be the area under a receiver operating characteristic curve. The receiver operating characteristic curve is known as the ROC curve, and the area under the curve is known as AUC. An example ROC curve is shown on the above. It is a correlation curve showing the performance tradeoff between the true positive rate and the false positive rate of a classification model. True Positive Rate is basically what we learned in class as Recall, and False Positive Rate is all False Positives as a ratio of all Negatives.

The AUC or the Area Under the ROC Curve is the primary performance evaluation metric in our project. It indicates the probability that the model ranks a random positive sample more highly than a random negative sample which provides an aggregate measure of performance across all possible classification thresholds. AUC ranges from 0 to 1, where 1 is perfect performance, and 0 is perfectly wrong performance. An AUC of 0.5 means the prediction is as good as a random guess. We will be using them as secondary evaluation metrics after AUC in this project.

Accuracy is the proportion of correct predictions, precision is the proportion of positive predictions that are actually positive, Recall is the percentage of actual positives that are identified correctly
Finally, F1-Score is an aggregated score based on precision and recall (also known as their harmonic mean)

Section D: Feature Generation

Methods

To generate features from text review data, we applied three feature generation methods: Bag of Words, TF-IDF, and Word2vec Embeddings. The features generated by these methods are trained, tuned and evaluated to identify the best feature inputs to the text classification classifiers.

See Appendix B for a detailed explanation of the feature generation methods we chose to use.

Parameter Tuning

We created several different datasets using each of these methods, and trained a logistic regression model for each generation method to predict whether a review falls into the highly negative class or the not highly negative class. Where possible, we created different versions of these features, using both unigrams and bigrams. We then selected the feature generation method with the best performance.
Using our data, we determined that the best performing n-gram ranges for Bag of Words and TF-IDF are as shown in the table.

The optimal n-gram range was found to be 1,1 for bag of words, and 1,2 for TF-IDF.

Using an n-gram range from 1 to 2 means that we will include both unigram and bigram tokens.

N-gram Range = (1, 2)
Text = " *an apple a day keeps the doctor away"*
Becomes: ['an', 'apple', 'a', 'day', 'keeps', 'the', 'doctor', 'away', 'an apple', 'apple a', 'a day', 'day keeps', 'keeps the' , 'the doctor' , 'doctor away']

Feature Engineering

Using the date feature and drug name and condition we added features to the TF-IDF features. We added date based features based on the date of the review, these were month, year, day of week. We also added features to indicate whether a review mentioned the name of a competing drug, how many times a review mentioned competing drugs, the number of different competing drugs that were mentioned, the proportion of total competing drugs in this dataset that were mentioned, and the number of times the name of the drug being reviewed was mentioned.

Section E: Classifiers and Performance Evaluation

Classifier Methods

Using the dataset created using TF-IDF features and our engineered features, we tried to estimate our predictor variable using logistic regression, naive bayes, random forest, gradient boost models in our study. We performed hyperparameter tuning on each model and selected the model (using the best estimators) with the highest AUC. This model was Naive Bayes.

See Appendix C for classifier definitions

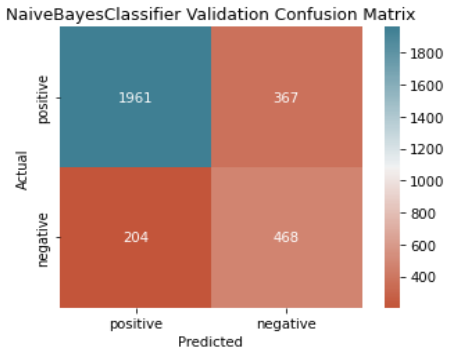
Hyperparameter Tuning

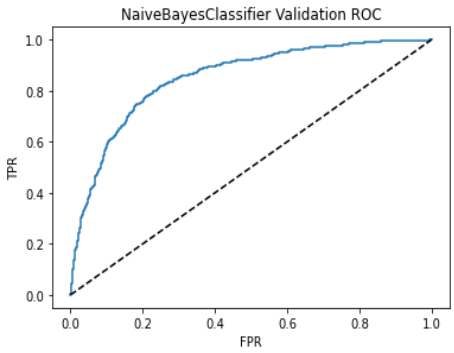
For each model we used the GridSearchCV function to search a range of parameters for the optimal hyperparameters. We selected the hyperparameters that yielded the highest AUC as the best parameters.

See Appendix E for a summary on hyperparameter tuning for each classifier method.

Performance Evaluation

For our models, we used AUC (area under the ROC curve) as the metric for model evaluation. AUC measures the entire two dimensional area underneath the ROC curve. AUC provides an aggregate measure of model performance across all possible probability thresholds for classification.
We evaluated the performance of our model by using true positives, false positives, true negatives, and false negatives to calculate precision, accuracy, recall, f1 score, and plot an ROC curve, which we used to calculate AUC. We plotted a confusion matrix which we used to visualize the success of the model. We did this for both in sample (training) data as well as out of sample (test) data in order to understand whether our model was overfit to the training data. Using a combination of these metrics, we were able to select the model that demonstrated the best performance.





Performance Test Custom Function

We built a python function that predicts the likelihood of a review being positive or negative. The review is given as an argument to this function after it has been cleaned (stop words eliminated, lemmatized, converted to all lowercase, only alpha characters retained).

Section F: Practical Implications

Ratings Analysis for Drug Companies

Using this model, we learned which words/features are most associated with negativity and positivity. Using the concordance function in nltk, we can look at the context around the negative words to further gain insight on the source of negative feedback on a product. A drug company can use this information to prioritize improvements to their drug.

For example, it may be useful for drug companies to know which words have the most influence in predicting the rating. If the word “headache” is useful in predicting negative reviews for a certain brand of drug, the manufacturer can attempt to reduce the severity of that symptom in their product. Using our engineered features that describe the reviewer’s mentions of competing drugs, a manufacturer can identify which competing products serve as their primary competition.

Use case analysis for Consumer and Researchers

From this research, a third party, such as a regulator, might be able to learn which pharmaceutical companies consistently produce inferior products. This knowledge might assist them in their regulatory capacities and help them identify which firms are producing poor outcomes for consumers.

Added Value for Drug Review Websites

The code we’ve written for this project can be used to classify raw text associated with a drug review as “likely highly negative” or “not likely highly negative” using a probability calculation ranging from 0 to 1. A website that curates drug reviews might choose to use this model to present users with a collection of positive, negative, and neutral reviews on the first page of their website. This also gives a drug review website insight into the customer’s review that the customer may not be aware of themselves. For example, a customer might consider their review to be merely neutral, and therefore give it a neutral rating on a scale from 1 to 10. While neutral to them, the text of their review might indicate a highly negative criticism of the drug company. Similarly, a customer may have arbitrarily given a low rating, when the text of their review indicates they did not have a highly negative experience. Someone who is interested in doing market research on the state of the pharmaceutical industry might be interested in using this objective evaluation of customer reviews in place of a customer’s subjective self-evaluation of their review.

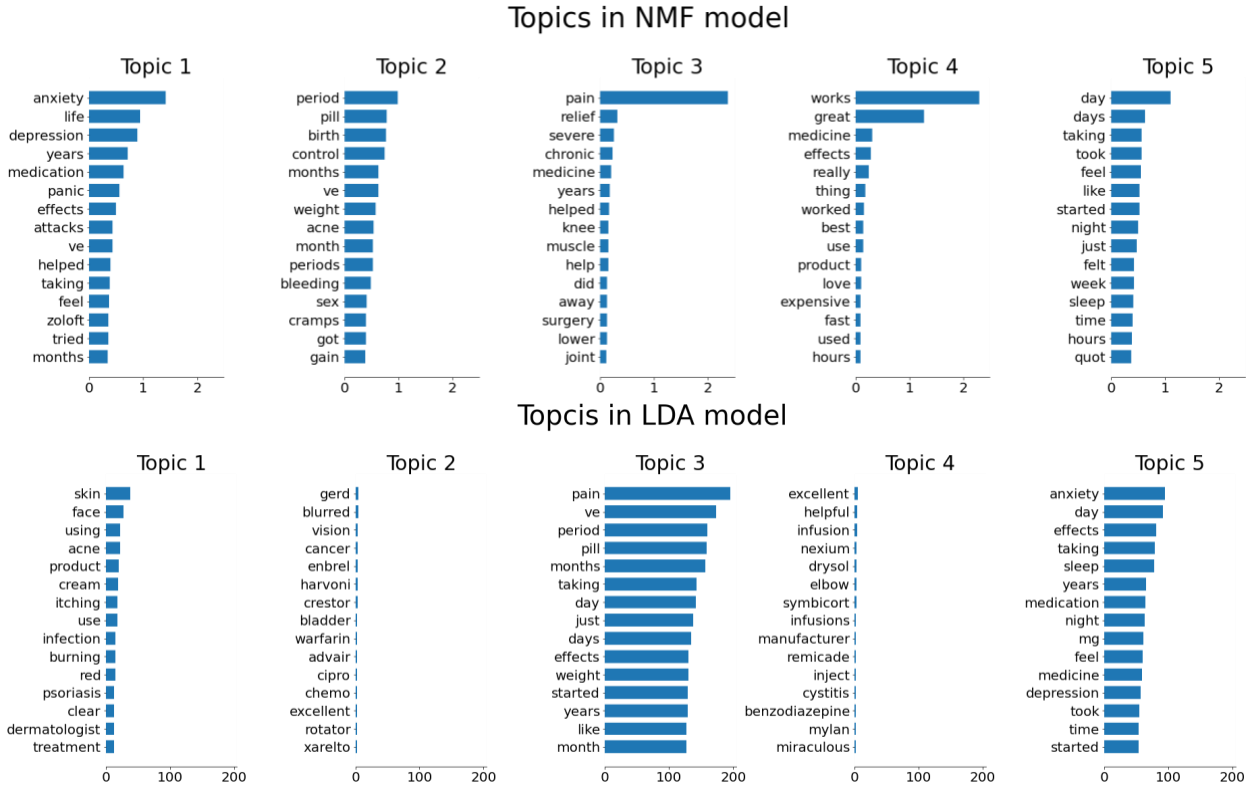
Section G: Further Research

Topic Modeling

During our study we found that it is hard to manage a large amount of text review data. We performed topic modeling on the drug review datasets. Topic modeling provides us a better way to manage unstructured text data.

We investigated two models for topic modeling. The first model is Nonnegative Matrix Factorization(NMF) which is a deterministic algorithm that decomposes a high dimensional matrix into two lower dimensional matrices.[3] The second model is Latent Dirichlet Allocation which is a probabilistic model that automatically classify any individual document within the corpus.[4] Some researchers suggested that NMF model generally performs better that LDA with shorter texts like tweets or reviews.[5] Furthermore, NMF runs much faster than LDA in our study.

In order to choose the right number of topics for our topic modeling study, we performed Kmeans clustering with the elbow method. Combining the result of the elbow method and our understanding for the dataset, we choose 5 topics for the topic modeling study. The top tokens for the topics from the NMF and LDA model can be seen in the graphs below.



Based on the results for NMF and LDA models and our common knowledge about drugs, we summarized the topics in the following table.

Topics	NMF	LDA
Topic 1	psychiatric	topical
Topic 2	feminine	chronic
Topic 3	pain	pain
Topic 4	effects	injectible
Topic 5	time	mood/endocrine

Because of the unstructured nature of text data, it is a challenge to manage large amounts of text data especially in the era of big data. Topic modeling provides us a practical method of categorizing unstructured data into finite groups. Not only in text mining, topic modeling has its potential in the study of other unstructured data like computer vision, bioinformatics and precision drug discovery, including oncology study.[6][7]

Appendix A: References

[1] Harris Zellig (1954). “Distributional Structure”. ***Word***. 146-162
[2] Tomas Mikolov (2013). “Distributed representations of words and phrases and their compositionality". *Advances in Neural Information Processing Systems.* arXiv:1310.4546
[3] Y. Wang and Y. Zhang (2013), "Nonnegative Matrix Factorization: A Comprehensive Review," *IEEE Transactions on Knowledge and Data Engineering,* vol. 25, no. 6, pp. 1336-1353, doi: 10.1109/TKDE.2012.51.
[4] David Blei, Andrew Ng, Michael Jordan (2003), “Latent Dirichlet Allocation”. *Journal of Machine Learning Research* 3(4-5): pp. 993-1022. doi:10.1162/jmlr.2003.3.4-5.993
[5] Yong Chen, Hui Zhang, Rui Liu, Zhiwen Ye, Jianying Lin,(2019) “Experimental explorations on short text topic mining between LDA and NMF based Schemes,” in *Knowledge-Based Systems,* Volume 163, Pages 1-13, ISSN 0950-7051, doi.org/10.1016/j.knosys.2018.08.011.
[6] Cao, L., & Fei-Fei, L. (2007). “Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes.” ***IEEE 11th International Conference on Computer Vision*** (pp. 1-8). IEEE.
[7] Valle, .F,Osella, M., Caselle, M. (2020). “A Topic Modelling Analysis of TCGA Breast and Lung Cancer transcriptomic Data” *Cancer.* 12(12): 3799, doi:10.3390/cancers12123799
[8] Treadway, A. (2021, December). *Cis 9660 - Lecture 8 - Text Mining & Nlp*. Lecture.

Appendix B: Research methods on feature generation

Bag of Words A simplified representation used in NLP. It originally comes from linguist Harris Zellig’s article of ‘Distributional Structure’ [1]. This method focuses on the frequency distribution of each word type. However, it disregards the grammar and the order of the words.

In our study, we first used the bag of words representation to extract features from the drug reviews. The frequency of each unique word can be used as features to train the rating classifier. After the extraction, the frequency distribution for each review is stored in a dictionary and then the convert all the dictionaries to a pandas dataframe for the whole corpus of the drug review dataset.

For simplicity, we are using the unigram model for the bag of words representation. The bag of words representation is relatively sparse because only a fraction of the total unique words occurred in one review. We discarded the words which occurred less than 20 times in the dataset to reduce the dimension and speed up the training time.

TF-IDF: TF-IDF stands for Term Frequency Inverse Document Frequency. TF-IDF calculates a score for each token based on the relative importance of the token in a document vs the overall corpus. Compared with the bag of words, TF-IDF not only considers the frequency of words on the document level but also considers the frequency of words on the corpus level. There are three steps to calculate the TF-IDF score.

Normalized term frequency (TF):

$$TF\ (t,d) = \frac{Number\ of\ times\ token\ appears\ in\ document}{Total\ number\ of\ tokens\ in\ document}$$

Inverse document frequency (IDF):

$$IDF\ (t,d) = \log\left(\frac{Total\ number\ of\ documents\ in\ the\ corpus}{Number\ of\ documents\ containing\ token}\right)$$

TF-IDF Score:

$$TF - IDF\ Score\ (t,d,D) = TF\ (t,d) * IDF\ (t,d,D)$$

Word2vec Word Embeddings: A neutral network based approach that represents each word in a corpus by a vector of numeric values. It was developed by a team at Google in 2013. [2] An embedding represents a word across N dimensions. This method tries to predict a word with its surrounding context.

Appendix C: Classifier Definitions

Naive-Bayes Classifier: Probabilistic classifier technique based on Bayes' theorem that measures probability of event A happening given that event B takes place

Logistic Regression Classifier: Statistical model that measures predicted probability that given observation belongs to the "positive" class (negative review in this project)

Random Forest Classifier: Ensemble learning method that measures average predicted probabilities from an independent collection of decision tree model classifications

Gradient Boosting Classifier: Algorithm based on converting weak learners into strong learners that measures average predicted probabilities from decision tree classification by retraining each tree in sequence every iteration using the gradient decent measure

Appendix D: Analytical Method Definitions

Text Classification: Supervised machine learning technique that assigns categories to each document in a collection of documents.

Topic Modeling: Collection of unsupervised machine learning techniques in machine learning that discovers the latent semantic topics in a collection of documents.

Appendix E: Hyperparameter Tuning Summary

Hyperparameter	Naïve-Bayes	Logistic Regression	Random Forest	Gradient Boosting
Alpha	✓			
Random State		✓		
Penalty		✓		
Solver		✓		
Max_Depth			✓	✓
Min_Samples_Leaf			✓	✓
Min_Samples_Split			✓	✓
N_Estimators			✓	✓
Learning Rate				✓
Subsample				✓
Max_Features				✓