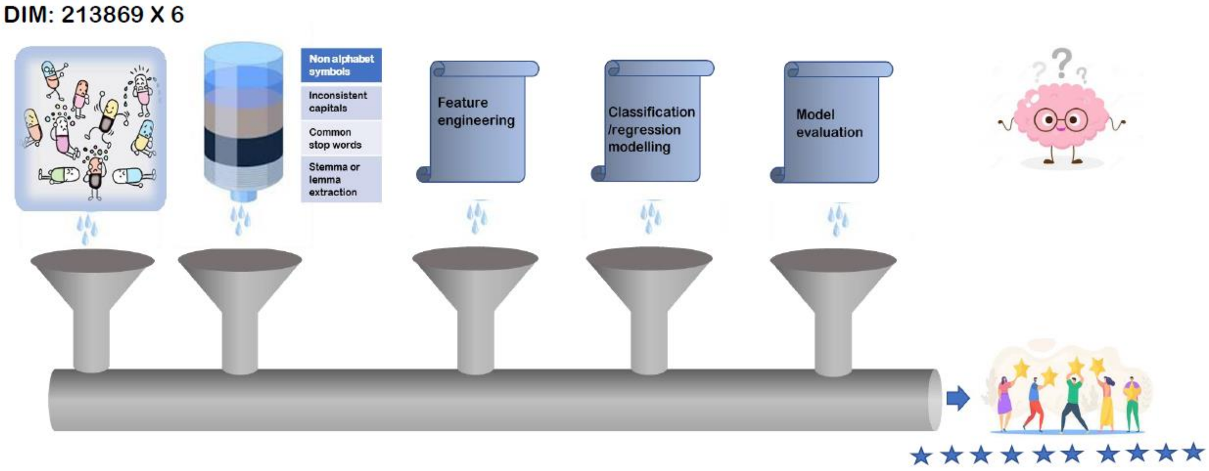
Drug Review Rating Prediction

A story about text data exploration



Lina Gao

May 25 2021

Task Statement

Given a large set of drug review data from UCI, we will develop supervised machine learning models to predict customers' ratings.

Introduction

Online user reviews in the pharmaceutical industry contain much valuable collective information about the drug's effectiveness and side effects. They can provide indispensable interesting insights for pharmaceutical companies in post-market analysis. It can also aid information for clinical decision-making about disease diagnosis and treatment. Compared with the rigorous standard clinical trials for a limited member of patients within a fixed period, analyzing the online drug reviews can improve monitoring the public health effect of drugs in a more real-life style.

Although sentiment analysis has been extensively applied in processing text data in web media, its application in medicine is still limited. At present, there are two approaches to sentiment analysis. One is the lexicon and rule-based sentiment analysis algorithm, such as Vader in NLTK and TextBob sentiment analyzer. Both of them are developed based on sentiment lexicons by matching textual units with opinion words in dictionaries and sentiment indexes. However, sentiment analysis is also domain-dependent and context-related. Thus, as an alternative approach, machine learning algorithms can provide a different perspective by training the model on context-specific data sets to determine the sentiment levels. This project will combine these two techniques and explore the sentiment analysis of drug review data from web media.

Data Exploratory Analysis

This data was collected from online pharmaceutical websites. It included more than 200,000 entries with seven features, drug name, condition, review, rating, date, and useful count. We deleted the entries with missing condition data, ~1194 items, from the original data before data processing. The final data set comprised of totally 3667 drugs for 916 different conditions. Both rating and UsefulCount showed skewed distributions, as indicated in figure 1. We can also see that birth control, depression, and pain are the top 3 among the 916 conditions. Since date is described as month-day-year, we use pandas to extract them individually as separate features, year, month, and weekdays. Checking the trend of review counts across time features, we found a significant increase since 2014 and low review counts on the weekend in figure 2. The data didn't show an apparent pattern across month variables.

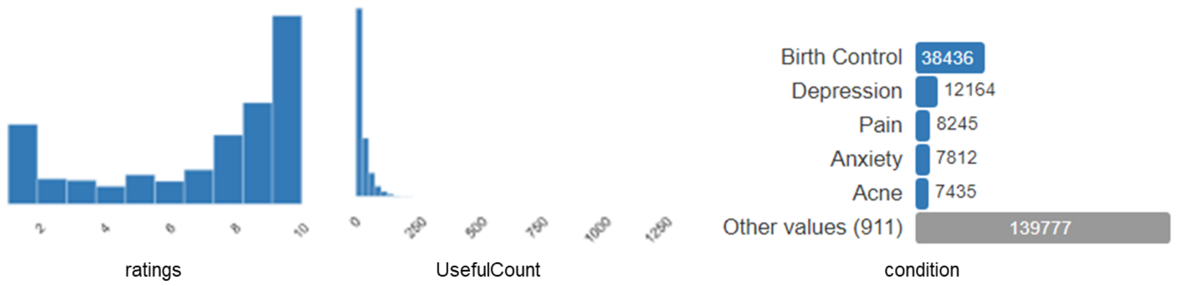


Fig. 1 Distribution of ratings, UsefulCount and top conditions

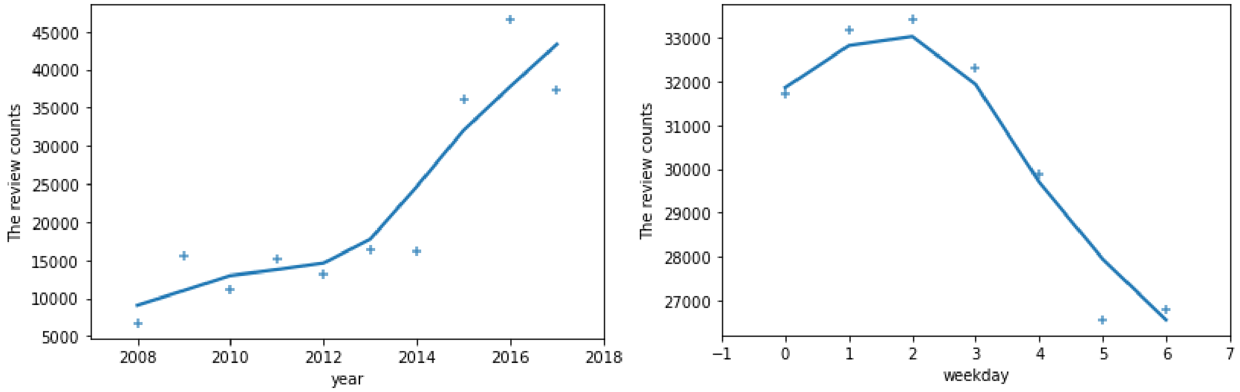


Fig. 2 Review counts across year and weekday

Considering the complexity of the data, we examined the features one by one in detail. We observed the various distribution of review counts cross month and weekday in different years for time-related data, as depicted in figure 3. Even when we focused only on one condition, such as birth control, their distributions are still varying. As for condition variable, the top list had been changed from pain in 2008 to birth control in 2017 gradually. The top drug names also switched from pain relievers such as acetaminophen, hydrocodone in 2008 to birth control drugs like levonorgestrel and Ethinyl estradiol in 2017. Besides, we noticed the complicated connections between drugs and conditions. One drug can be applied to multiple conditions, and multiple drugs can treat one condition. As for ratings, we found people became more critical as time went on. Although most reviewers gave high ratings in 2008, both low and high ratings of reviews became dominant in 2017. Even the top three conditions across ratings are diverse, although birth control is persistently the top 1 condition. Most reviews have useful count of less than 100. There are only four reviews with the highest useful count, more than 1000. The distribution of UsefulCount vs. review count is similar regardless of the ratings of the reviews.

We used word cloud to exam the review feature in the drug data. Considering the wide variety of the review context, we chose the top 3 conditions, birth control, depression, and pain, to study. After proper text data cleaning, we applied both unigram, bigram, and trigrams to extract the

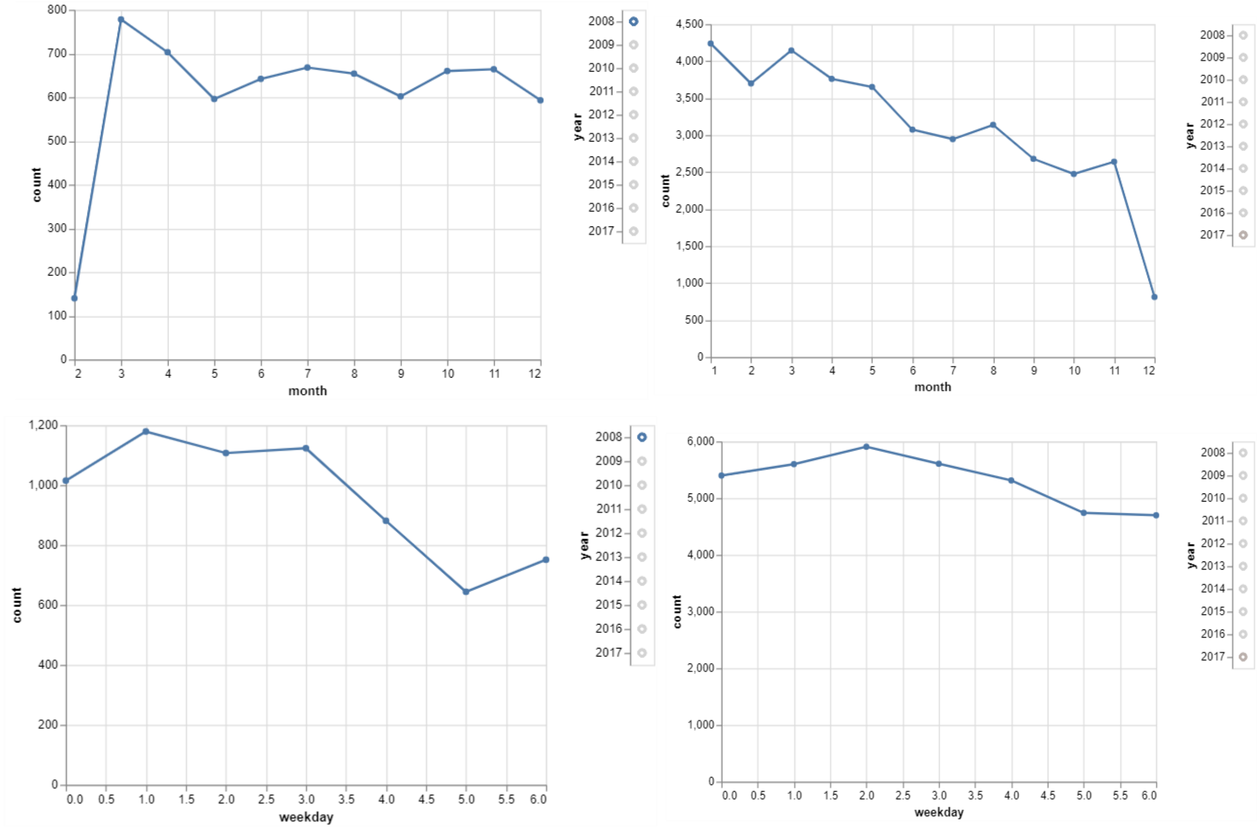


Fig. 3 The diverse trends of review count across weekday and month in different year

essential information. Unigram word cloud without distinguishing high and low ratings gave us mixed bags of words, including positive and negative information. However, we obtained much better information after differentiating high rating (9-10) reviews from low rating (2-1) reviews. The same situation happened for bigram and trigram word cloud.

Meanwhile, we found the specific information from unigram, bigram, and trigram word cloud. For example, we found drug names mentioned more frequently in the unigram word cloud but not in bigram and trigram. However, bigram and trigram can give us a more detailed syndrome

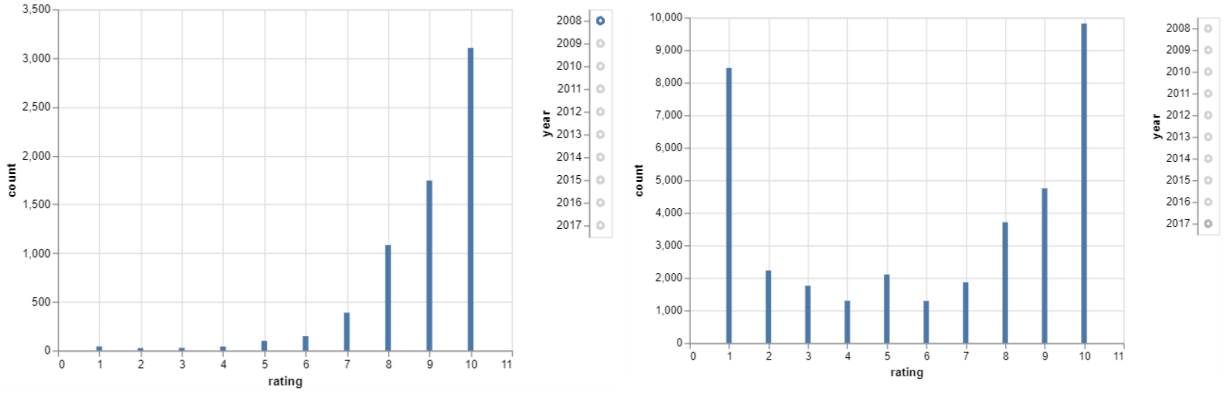


Fig. 4 The diverse trends of ratings cross year

description compared with the unigram word cloud. Therefore, the situation is better depicted in figure 5, taking pain reliever reviews as an example.

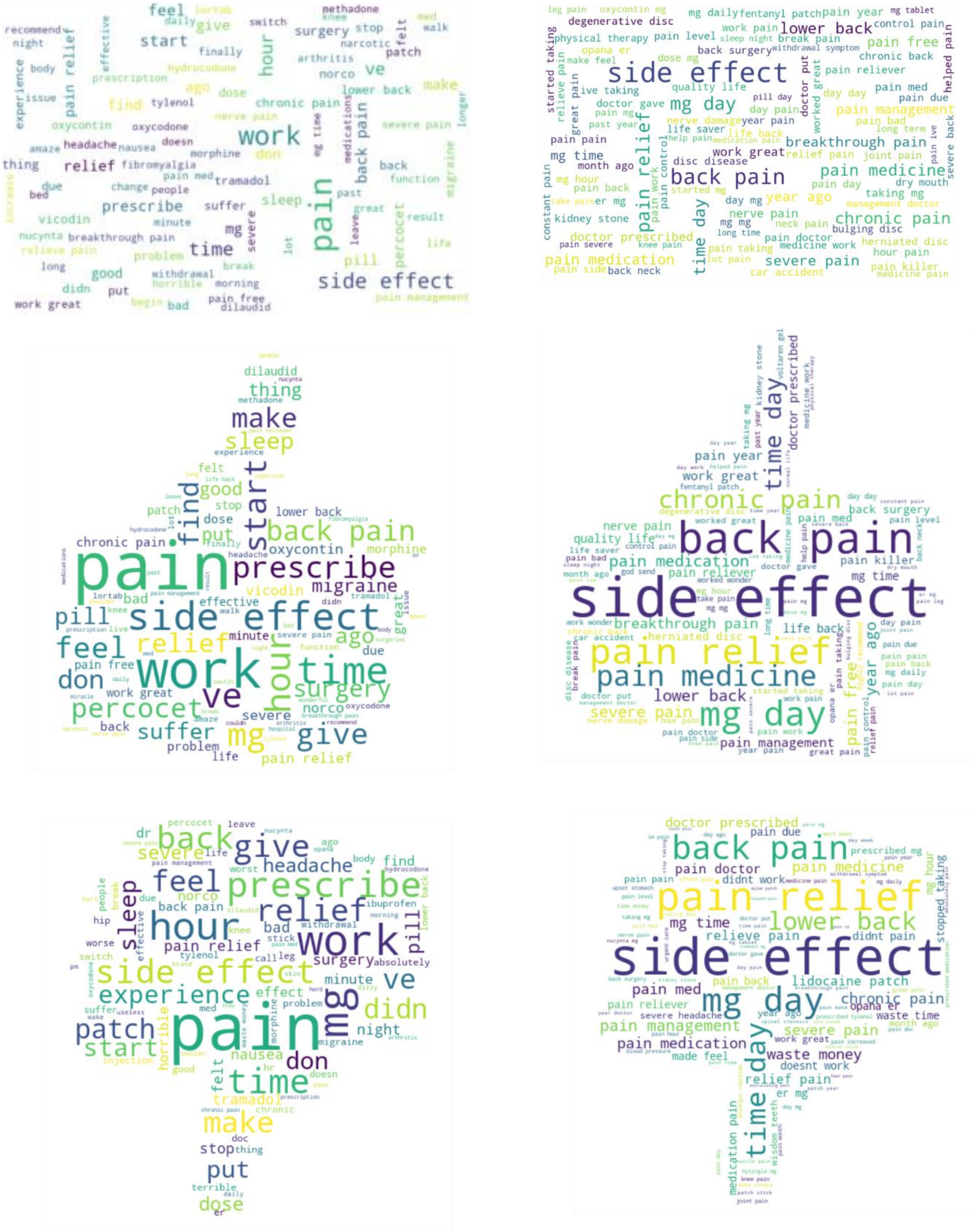
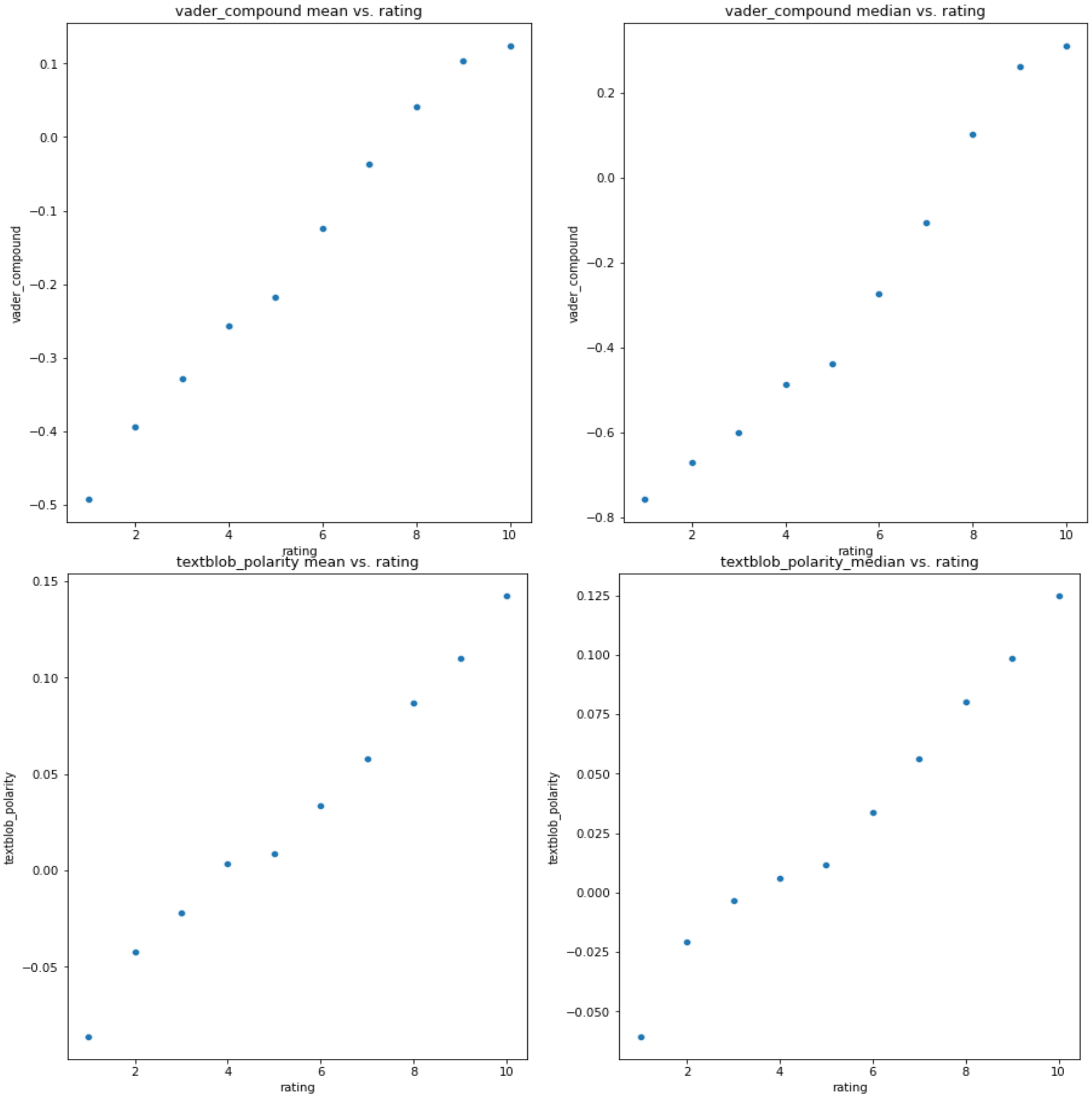


Fig. 5 Wordcloud for pain reliever review

(left:unigram, right: bigram)

Feature Engineering

To extract sentiment information efficiently as features for modeling, we utilized built-in sentiment analyzer VADER (Valence Aware Dictionary and Sentiment Reasoner) from NLTK and TextBlob sentiment analyzers. VADER can give us a compound score, which is calculated by normalizing the sum valence scores of each word in the lexicon according to the rules. Its value is between -1 (most extreme negative) to +1 (most extreme positive). TextBlob sentiment analyzer produces two features, polarity, and subjectivity. The polarity value lies in the range of [-1,1], where one means positive statement and -1 means a negative statement. Subjective sentences generally refer to opinion, emotion, or judgment, whereas objective refers to factual information. Subjectivity is also a float that lies in the range of [0,1]. Combining compound score, polarity, subjectivity with the other features, we roughly evaluated the importance of these features by plotting their mean or median cross rating. We can easily identify the importance of compound score, polarity, and useful count, as shown in figure 6.



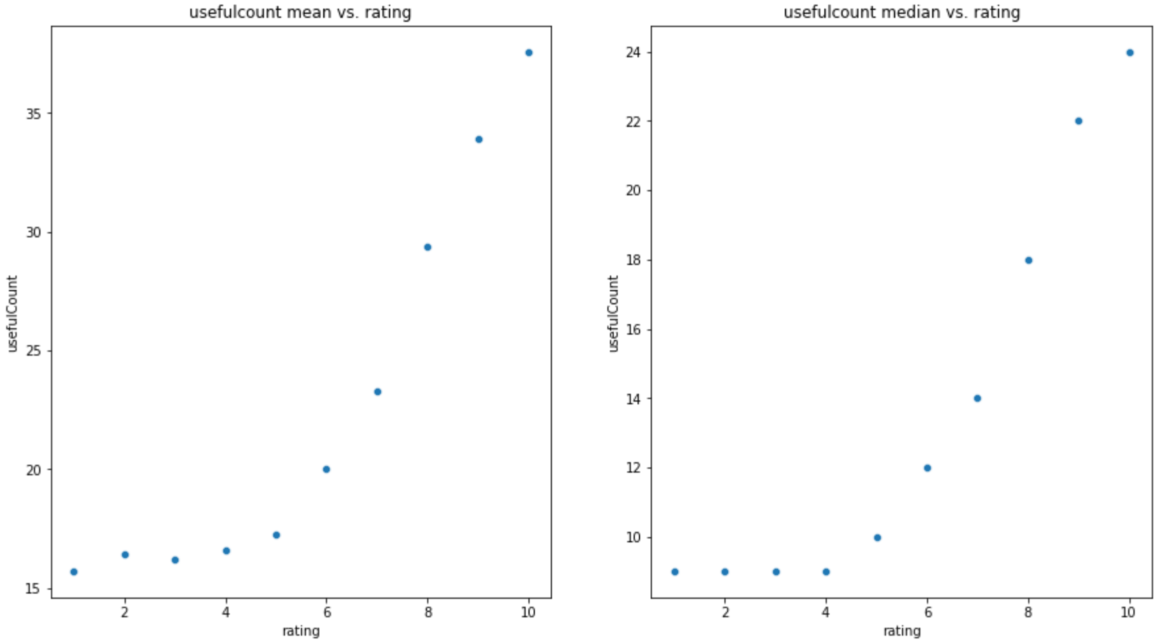


Fig. 6 The scatter plot of important features and rating

Moreover, we applied TfidfVectorizer to transform the review data and got 288 new features. We obtained the data set, tfidf_df, for the modeling step by combining these features with the other original features. We also did dimension reduction using truncated SVD to extract information from these new features, just like PCA analysis and obtained 110 components. Then we pooled these components with the

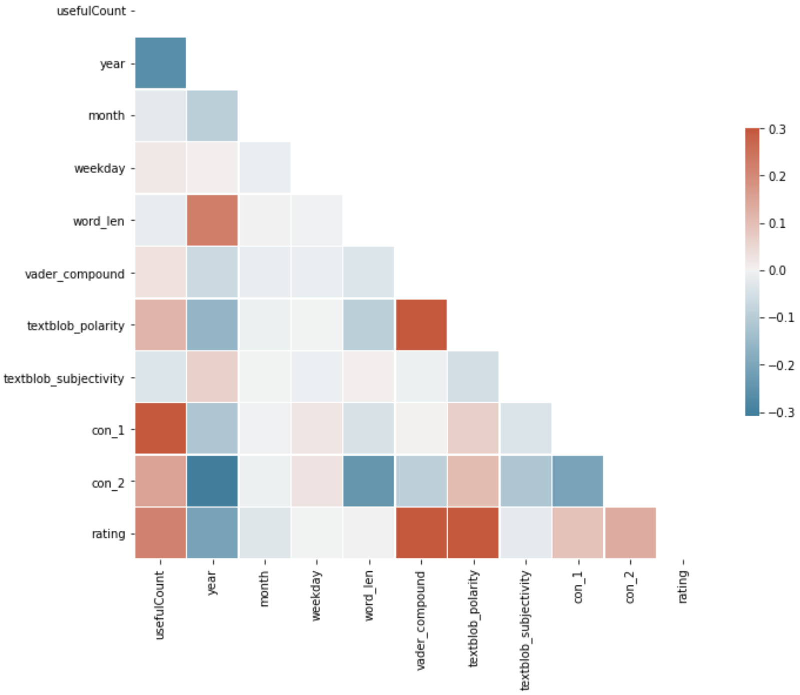


Fig. 7 The correlation matrix of the features in this study

other original features as tfidf_lsa_df as well. Although latent semantic analysis (LSA) is regarded as a text mining technique identifying keywords to summarize an extensive collection of text information, it can also become a feature generator in this study.

Considering the diversity of the review context, we chose the top 3 condition-related entries as our study data set. They are birth control, depression, and pain. With conditions converted into dummy variables in these data set, we got the correlation between all features except TfidfVectorizer transformed ones, as indicated in figure 7. Luckily, there was little correlation between them.

Regression model development for rating prediction

We used both tfidf_df and tfidf_las_df to explore the regression model development for review rating prediction in this study, as depicted in figure 8. We selected three algorithms, regularized linear regression, decision tree, and support vector machine. The Elastic-Net in sklearn uses both penalties L1 and L2 of the Lasso and Ridge regression methods. It can help us to reduce the dimensions if there are multiple correlated features. As powerful yet flexible supervised machine algorithms, support vector machine is very efficient in high dimensional data analysis, mainly because of only the subset of the training data involved in the decision function. Tree models are popular considering their low requirement for data processing and advantage in interpretation. However, when we applied all of these algorithms for regression modeling, the results were not

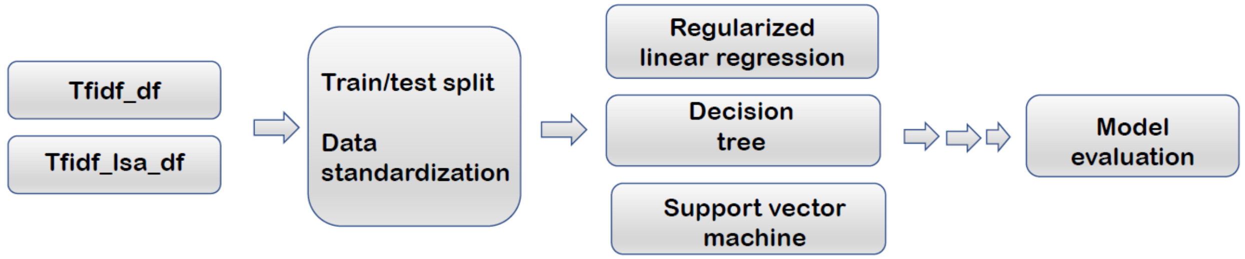


Fig. 8 Regression model development for rating prediction

Table 1. Regression model fitting results using the test

Data	Algorithms	Run time	hyperparameter	MSE	MAE	R^2
Tfidf_df	Regularized linear regression	159	Alphas, L1_ratios	6.12	1.99	0.43
	Decision tree	97	Max_depth	5.03	0.97	0.53
	Support vector machine	1908	Default setting	6.12	1.99	0.43
tfidf_lsa_df	Regularized linear regression	51	Alphas, L1_ratios	6.38	2.04	0.40
	Decision tree	140	Max_depth	5.18	0.98	0.52
	Support vector machine	735	Default setting	6.38	2.04	0.40

promising, as shown in table 1. The predicted results for the test set were not good, as indicated by R^2, MSE, and MAE. Another issue is running time, especially for support vector machine. It took a long time to finish. As for the ensemble algorithms like random forest or XGB, we had to give them up in the middle of the testing because of the long run time.

Classification model development for rating prediction

To convert the problem into a classification one, we defined a new categorical variable for rating. If the rating is larger than 5, we regarded the review as a positive one, assigned 1. Otherwise, it was considered as the negative one, assigned 0. The distribution of rating in the training set after such conversion is 28859 positive reviews vs. 12332 negative ones. In the testing set, there were 12368 positive and 5286 negative ones.

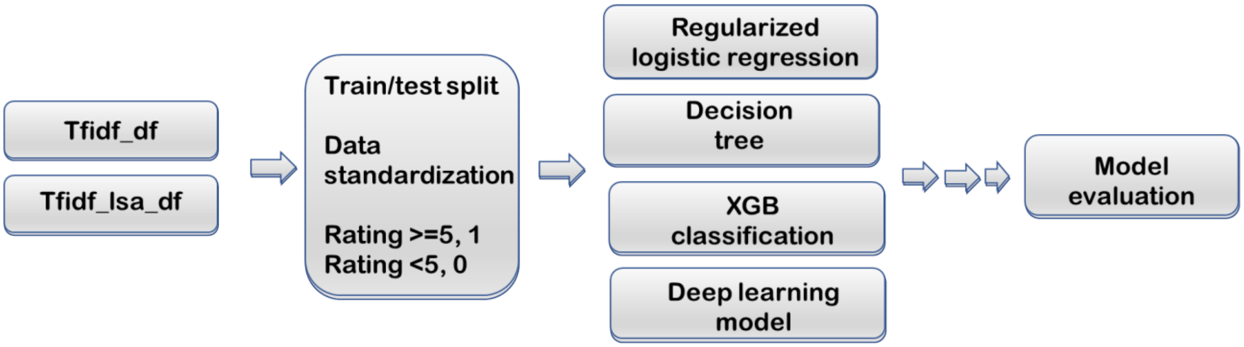
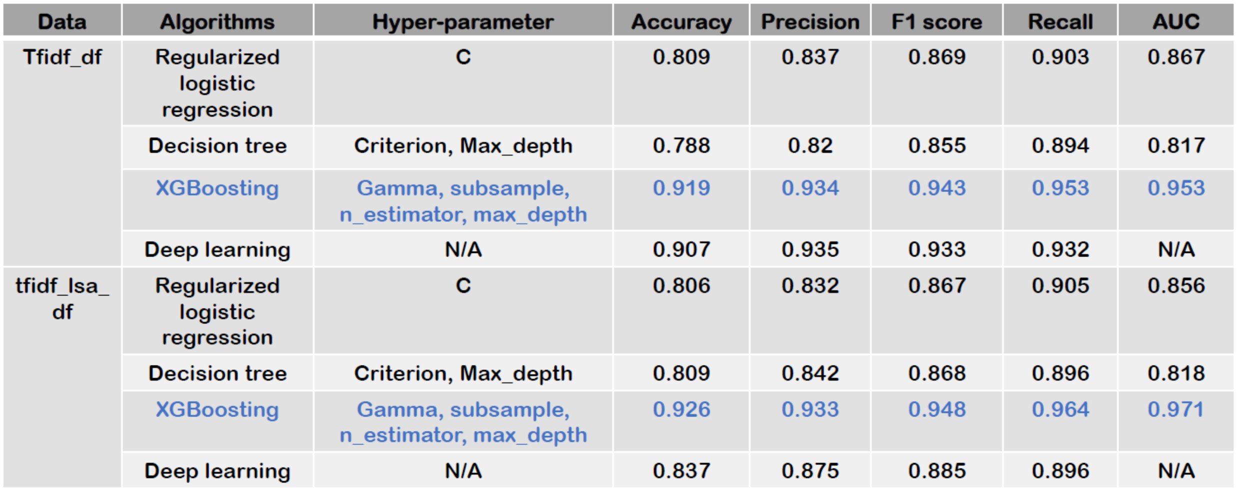


Fig. 9 Classification model development for rating prediction

We tested four machine learning algorithms for classification modeling, regularized logistic regression, decision tree, XGB classification, and deep learning models. The relevant workflow is depicted in figure 9. With random optimization of the critical hyperparameters in each algorithm, we found the best model using each algorithm and evaluated their performance with the testing set. Because the rating data do not balance after conversion, accuracy was not regarded as the golden standard any longer. We combined f1 score, recall score, AUC with accuracy to assess the models. The results were listed in table 2. XGB algorithm was the best performer considering its consistently highest f1, accuracy, recall, and precision score among all models. Besides, LSA-produced components gave us comparable results as tfidf transformed features, because we got similar results using tfidf_lsa_df to tfidf_df.

Deep learning models have been widely applied in natural language processing in recent years. In this study, we used simple RNN. It is a sequential neural network with blocks linking to each other like a chain in three layers. Without many optimizations of hyperparameters, RNN results were not as good as XGB, as shown in table 2.

Table 2. Classification model fitting results using the test set



Data	Algorithms	Hyper-parameter	Accuracy	Precision	F1 score	Recall	AUC
Tfidf_df	Regularized logistic regression	C	0.809	0.837	0.869	0.903	0.867
	Decision tree	Criterion, Max_depth	0.788	0.82	0.855	0.894	0.817
	XGBoosting	Gamma, subsample, n_estimator, max_depth	0.919	0.934	0.943	0.953	0.953
	Deep learning	N/A	0.907	0.935	0.933	0.932	N/A
tfidf_lsa_df	Regularized logistic regression	C	0.806	0.832	0.867	0.905	0.856
	Decision tree	Criterion, Max_depth	0.809	0.842	0.868	0.896	0.818
	XGBoosting	Gamma, subsample, n_estimator, max_depth	0.926	0.933	0.948	0.964	0.971
	Deep learning	N/A	0.837	0.875	0.885	0.896	N/A

Besides, vader_compound, textblob_polarity, and usefulCount are the most critical features regardless of the algorithms and data set.

Summary

- Regression models are not a good choice for rating prediction, but classification models can give us decent results
- Features created by tfidf with LSA can produce comparable results as tfidf alone engineered features.
- XGB is the best performer among all tested algorithms
- Deep learning models are not as good as XGB models in this project.

Future work

- Develop more comprehensive models including all conditions in the original data instead of the top 3.
- Optimize the hyperparameters in RNN models and test their performance again. We can also try CNN models. They are a class of deep, feed-forward artificial neural networks where the connections between nodes do not form a cycle like RNN.
- Develop a web application to give the predicted results directly to the stakeholders.
- Try auto training and fast deployment for state-of-the-art NLP models on Hugging Face