Models and Sentiment Analysis of Drug Review Data

Ivette Planell-Méndez
Princeton Neuroscience Institute
ivettep@princeton.edu

Abstract

In this study we aim to develop analysis techniques that can be applied to publicly available drug review data to improve methods for matching patients to drugs that will treat their symptoms while minimizing side effects. We use sentiment analysis to extract patterns in the reviews, classification to predict the patients numerical rating of the review, and various data and user driven feature selection methods to optimize models.

1 Introduction & Motivation

The growth of the pharmaceutical industry has created a vast array of treatment options. In some ways this is helpful because it allows for customized treatment; however, it also means that patients often go through a period of trial and error before finding the drug that works best for them and is most effective while minimizing side effects. This issue can be particularly difficult in the case of certain conditions. Birth Control, mental health conditions, and chronic pain are some of the most notorious for causing patients to go through a long and difficult trial and error period before finding the right medication [1, 2] . This process can be harmful and dangerous for patients. In the case of pharmaceutical treatment for depression or other mental health conditions, the process can discourage patients, lead to debilitating side effects, and make them swear off medications [3].

It is imperative that we find better methods of matching patients with drugs to shorten the trial and error period and to more quickly find an effective fit. Publicly available drug reviews and online communities where people discuss their experiences with different drugs might be able to provide important insights in addressing this issue. The growth of online communities in recent years had led to more people sharing their experiences online, including their experiences with medications, and there are entire communities just for this purpose.

Machine learning could provide the tools necessary to leverage these review data by extracting meaningful patterns which could help improve the chances of prescribing drugs best tailored to a patients symptoms and that minimizes side effects that they might have experienced with similar drugs [4]. In this study we will look at drug reviews and develop methods for extracting patterns and sentiments from them. Furthermore, we consider how various feature selection methods can help uncover meaning.

2 Exploratory Data Analysis & Preprocessing

2.1 Description of the Data

The data used in this project was retrieved from the UCI Machine Learning repository [5]] and was originally presented at the 2018 International Conference on Digital Health [4] which looked at how sentiment analysis of drug reviews can be used to predict overall satisfaction, effectiveness, and side effects. The dataset was collected from two webpages where users can write reviews on different drugs and obtain information regarding their pharmaceuticals: Drugs.com and Druglib.com. It contains over 215,000 reviews and, in addition to a text review of the drug in question, each entry also

contains a unique and anonymized user ID, the name of the drug, the name of the condition being treated, a numerical rating from 1-10, the date the review was written, and the number of users who found a given review useful.

Given the tremendous size of the data set and the limited computational power available to use, we randomly down sampled the dataset from the original size to 40,000 reviews, but wrote all analysis in a way such that the could be run on the larger dataset on a different machine. This smaller subset of data that we considered contained reviews for 2386 different medications for 679 different conditions. This means that there is a 3.5:1, drug to condition ratio (similar to what we saw in the original dataset). Upon closer inspection we saw that the ratio of drugs to conditions varies dramatically across the data set and there were some conditions with over 140 different medications (fig. 1A). Furthermore, the most common conditions (fig. 1B) were the same conditions that has the most drug options. We also looked at the most common drugs (1C) and indeed they were drugs that target the most common conditions reported. In fact eight out of the top fifteen most common drugs were for birth control medications and six out of the top fifteen are used to treat depression.

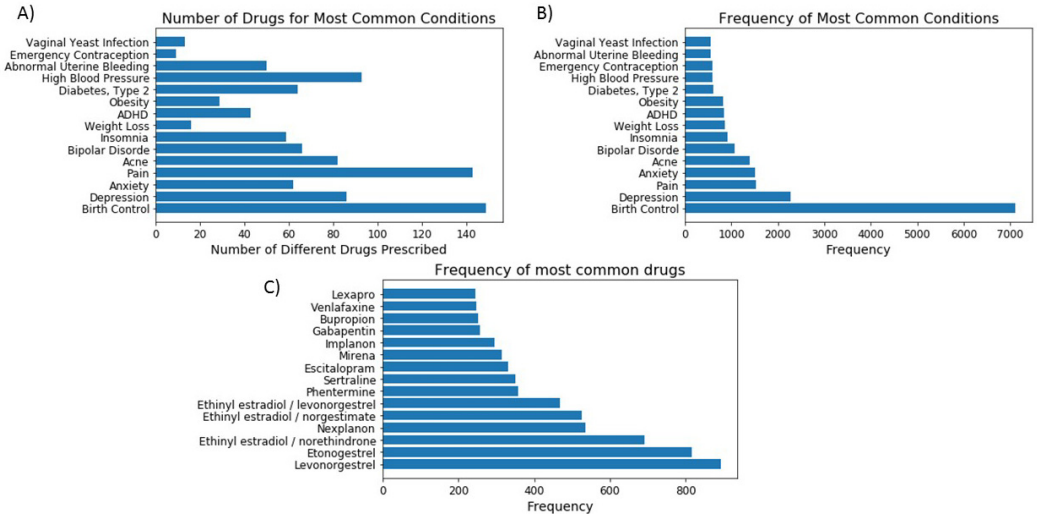

Figure 1: **A)** How many different drugs are used in the dataset for the top conditions. **B)** Top conditions in the data set and how many people report the condition. **C)** The names of the most common drugs and how frequently they appear in the dataset.

We also considered the most frequent words used in each rating category (1-10) to get an idea if there were any readily emergent patterns or predominant types of words in each category(fig. 2. We saw that words such as ”day”, ”month”, ”effect”, were present in almost all of the rating categories. This indicated that, while these words show up frequently in the reviews, they are not predictive of which specific rating the user gave. This suggests that exploring different methods of feature extraction might be useful in creating more accurate classifiers for determining rating categories. Non-predictive words such as these could be taken out of consideration when building classifiers. This would result in a smaller and more predictive feature set which could improve the accuracy and computational time of our classifiers.

2.2 Preprocessing

We used a 75/25 train/test split which resulted in a training set consisting of 30,000 reviews and a testing set consisting of 10,000 reviews. The latter was used do develop classification models, feature selection methods, and unsupervised dimensionality reduction for topic analysis. The former was used to evaluate the efficacy of these models.

We processed the training data using a script that implemented the Python NLTK module [6]. First the text file of training reviews was divided into separate words (tokenized), made lower case, and had stop words such as ”the” and ”a” discarded. Next, the words were lemmatized and stemmed using the Porter stemming methods [7]. This step is important because it allows words with the same

Figure 2: Word clouds for most frequent words used in reviews that were paired with each different rating category (1-10).

stem (e.g. watching and watched) to be considered as one for purposes of building our classifiers since they hold the same general meaning.

Finally, we filtered out words that created less than a minimum number of times and we imposed six different minimums to create six different bag-of-word representations. Given the different minimum word requirements, this resulted in different dictionary sizes for each bag-of-words representation (fig. 3). These different dictionaries of the training set vocabulary was then used to create six different bag-of-words representations for the test set.

Bag-of-Words Representations	
minimum word requirement	resulting dictionary size
10	4780
20	3497
40	2484
80	1688
160	1119
320	670

Figure 3: Resulting dictionaries used in bag-of-words representations when imposing different minimum word frequirements.

3 Methods

3.1 Classification Methods

This study implements the following five different classifiers using Python SciKit Learn module [8]. These were implemented with all of the default parameters unless otherwise specified.

1. *Naïve Bayes* (MNB): Multinomial form, alpha = 1
2. *Support vector machine*(SVM): linear with $\ell_2 penalty$
3. *Decision Tree* (dtree): gini criterion, min leaf = 0
4. *Random Forest* (randomforest): gini criterion, min leaf = 1
5. *Extra Trees* (extra trees): gini criterion

3.2 Evaluation Metrics for Classification

We performed evaluation for all classifiers by using the classifier developed with the training set of 30,000 reviews, to predict user drug ratings of the held out testing set of 10,000 reviews and then compared the results from the predictions obtained to the known ratings. We implemented a number of different evaluation metrics including accuracy and precision. To calculate these measures we obtained the number of positives (p), negatives (n), true positives (tp), false negatives (fn), false positives (fp), and true negatives (tn) from the confusion matrices of the results from each classifier and computed them as follows:

$$Accuracy = \frac{tp + tn}{p + n} \quad Precision = \frac{tp}{tp + fp}$$

We also looked at computational time which was an interesting metric for this case, because any practical implementation of these methods that might help doctors better prescribe their patients should be expected to perform quickly so that they can be used in at appointments while the doctor discusses medication options with the patient.

4 Results

4.1 Effect of dictionary minimum word of classifier performance

When we evaluated our classifiers across the different bag-of-words representations that had different minimum frequency requirements, we saw that making this requirement more strict (i.e. requiring the words to repeat more times across reviews in order to be included in the dictionary) did not improve performance accuracy. In fact, accuracy remained approximately constant across each different condition for all classifiers implemented (fig. 4A). Depending on the classifier, accuracy was between 0.35 and 0.42 and while on the low end this performance was promising because it was still significantly above chance performance (0.1). Since requiring more repetitions effectively reduces the dictionary size as seen in figure 3, the computational times across these different conditions was greatly reduced among all classifiers except for support vector machine (fig. 4B).

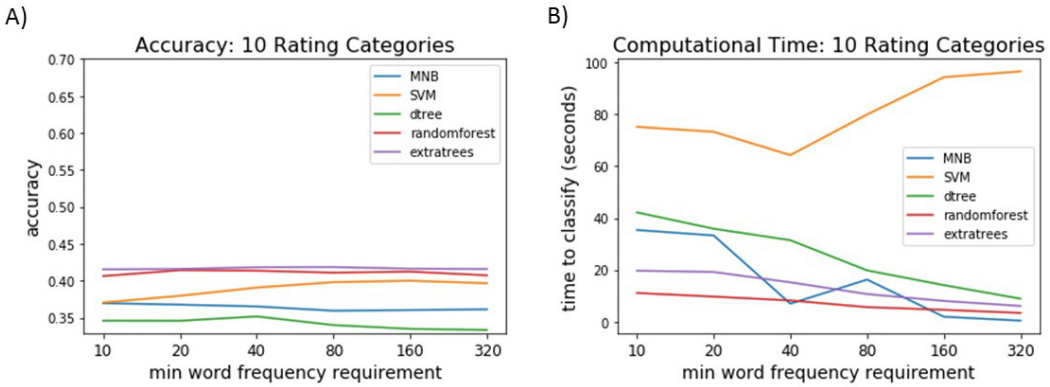

Figure 4: **A)** Accuracy across the bag-of-words with different minimum word frequency requirements. **B)** Computational time across these classifications.

4.2 Redefining ratings as negative, neutral, positive

While the previous classification results were promising and indicative of potential in the dataset, we wanted to see what might be causing difficulty in reaching higher accuracy. When we looked at the distribution of ratings across the dataset we saw that people tended to rate things on the high end or very low and that there was much less representation of middle ground ratings (fig. 5). Intuitively this makes sense since people that are more likely to post a review online are those that either had a really positive experience or a really negative one. However, this poses a problem for training classifiers for optimal performance because there proportionally way less data for certain

ratings. Furthermore, it might be the case that a 10 point rating scale has way too fine of a resolution. Perhaps there really is not a big difference when a user rates a drug as a '1' or '2' or as a '5' or '6' and these category differences are blurred.

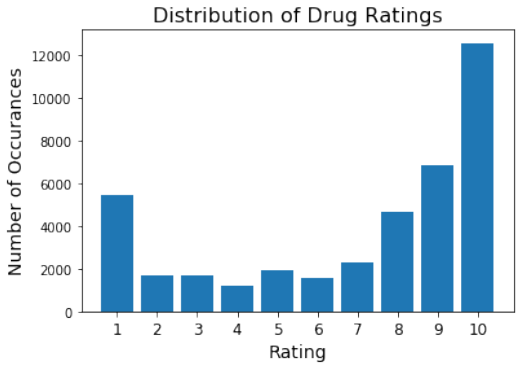

Figure 5: The number of times each rating is represented in the data set. The highest and the lowest ratings have the highest frequency.

In order to combat this issue, we redefined the rating categories from a ten point scale (1-10) to a three point scale (negative, neutral, positive). We assigned user ratings 1-3 as "negative", ratings from 4-7 as "neutral", and ratings 8-10 as "positive". We then trained our five classifiers on the bag-of-words representations with the different minimum word frequency requirements and the results were quite different then when we trained on the 10 point scale. For minimum word frequency 10, 20, 40, ad 80 the the accuracy achieved by the classifiers on the three point scale was more or less identical to that which was achieved on the 10 point scale. However, for the most strict minimum word frequency requirements (160 and 320) accuracy doubled for classification on the three point scale (fig. 6A). This confirmed our idea that perhaps sentiments were blurred out across to fine or a resolution with the ten point scale. In this case, computational time again decreased with smaller dictionary size, but in most cases was comparable to computational time on the ten point scale (fig. 6B). There was one exception. In the case of the 3 point rating scale computational time also decreased for support vector machine instead of increasing like it had when using the ten point rating scale.

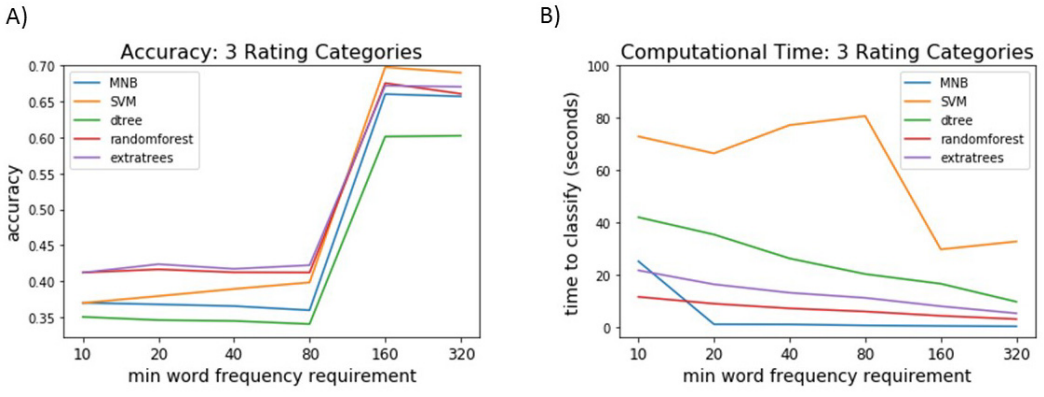

Figure 6: **A)** Accuracy across the bag-of-words with different minimum word frequency requirements. **B)** Computational time across these classifications.

4.3 User condition for feature selection

We also investigated whether developing classifiers for only one condition at a time (a form of feature selection) would yield more better models. We hypothesized that this might be the case because it users taking medications for the same condition might have similar experiences with the

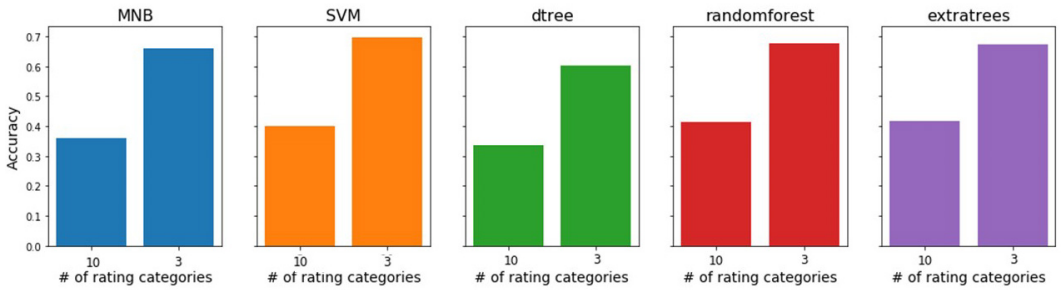

Figure 7: Comparative performance between classification using the original 10 rating classes (1-10) or the redifined 3 classes (negative, neutral, positive).

medications and this might result in patterns that are easier to identity. We tried this for three of the most common reasons that users took medications: birth control, depression, and pain. We found that the performance achieved by selecting for certain conditions (fig. 8) did not differ from the performance achieved for the entire set (fig. 7). We trained the models for both the three point scale rating and the ten point scale rating to see if there was any effect in either case, but we did not find any.

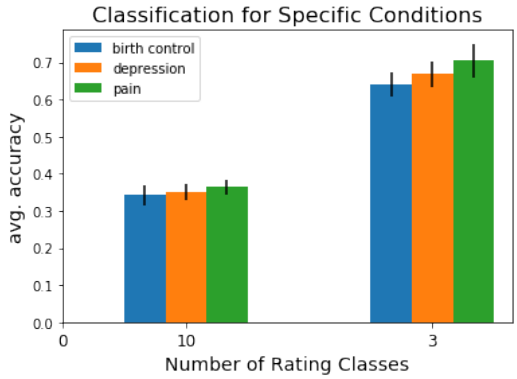

Figure 8: Average classifier performance for the case of 3 rating classes vs 10 rating classes.

4.4 User determined feature selection

Converting to a three point negative/neutral/positive rating system about doubled classifier accuracy, but we wanted to see if we could improve this even further by performing feature selection informed by the user interpretation of the reviews. In addition to reviews and ratings for different drugs, the data set also included a variable that indicated how many times other users marked a given review as ”useful”. We refer to this as the ”usefulness count”. If a review has a higher ”usefulness count” that another it means that more users found that review useful or informative.

This variable is a good candidate to inform feature selection because it is likely the case that users will mark as ”useful” those reviews which have more information and language that is more indicative about their sentiments in regard to the drug they are reviewing. Therefore, this could help weed out uninformative reviews that could confuse our classifiers. In order to explore the effect of usefulness count as a feature selection method for drug review data, we imposed different usefulness requirements and compared how the classifiers performed for these different requirements. We iteratively removed reviews from the training and testing sets that has usefulness counts under 10, 20, 50, 100, 150, and 200 (TABLE). This created datasets that had less reviews because some of the less predictive ones had been removed (fig. 9). Given that we previously achieved high accuracy with the bag-of-words that had 160 minimum frequency requirement, this feature selection was performed

on that bag-of-words to see if we could further improve performance. Therefore these analyses use a dictionary of 1119 words.

Min. Usefulness Count	# reviews
original data	40,000
10	25,928
20	17707
50	6710
100	1811
150	571
200	215

Figure 9: This table shows the number of reviews remaining in the data set after different requirements imposed for usefulness-based feature selection.

When we trained our five classifiers across these different usefulness constraints we saw that performance of all classifiers increased as the usefulness criterion became more and more strict (fig. 10A). A usefulness count minimum of 200 resulted in the highest accuracy levels, with support vector machine, decision tree, and random forest all reaching an accuracy of 0.963 (fig. 11). Additionally, computational times quickly decreased to the millisecond timescale with higher usefulness count which is much less than what it was for other methods (fig. 10B). We would expect there to be a point where the improvement in accuracy plateaus and drops off due to there not being enough reviews conforming with the higher imposed usefulness criteria, but we did not reach this point and more analysis would need to be completed to determine where this inflection point is.

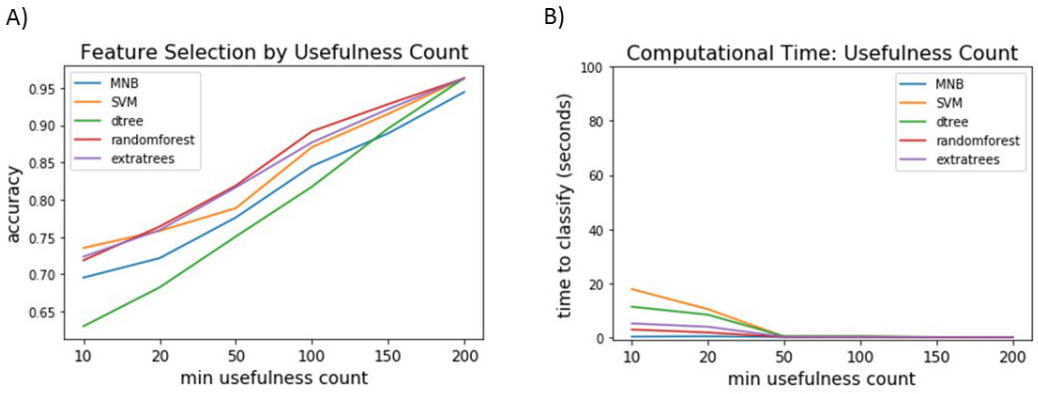

Figure 10: **A)**Accuracy across the bag-of-words with different usefulness counts used for feature selection. **B)** Computational time across these classifications.

5 Discussion

In this study we compared the performance of five different classifiers in predicting numerical ratings of medications based on patient reviews. We were able to get classifiers to perform within 4%accuracy of each other and the maximum performance that we were able to reach overall was 96.3%. This was achieved by support vector machine, decision tree, and random forest classifiers. In order to optimize performance we performed cross validation on all models to optimize out parameters.

We were able to improve performance dramatically in two main ways. First, we determined that a three point classification scale (negative, neutral, positive) was best for this purpose rather than a ten

Accuracy for Min. Usefulness Count = 200	
MNB	0.944
SVM	0.963
dtree	0.963
randomforest	0.963
extratrees	0.926

Figure 11: Accuracy of the different classifiers when a usefulness count of 200 or more was required as a feature selection method.

point scale (1-10). There is so much subjectivity in giving a numerical rating on a scale of one to ten that it is difficult to extract the most meaningful differences in this way and sentiments towards the medications tend to "blur" across adjacent ratings. We adapted a negative/neutral/positive scale to ameliorate this issue and when we implemented this performance on all classifiers doubled and computational time decreased.

Our second major improvement came from exploring multiple different methods of feature selection in order to determine which feature selection metrics are most appropriate when building models of drug reviews. The feature selection methods that we considered were:

1. Imposing different word frequency requirements when building bag-of-words representations.
2. Training classifiers separately for different conditions.
3. Using "usefulness count" contributed by online readers to only include the most predictive reviews.

We found that stringent word frequency requirements alone do not improve classifier performance. They only improve performance in the most stringent work frequency requirements that we tested when paired with adaptation to the three point rating scale previously discussed. We performed feature selection based on condition for birth control, depression, and pain. Surprisingly, this had no effect on performance and classifiers performed on par for the individual conditions to how they performed for the whole data set. We believe that this needs further analysis as we only tested this on some of the most frequent conditions in the dataset. We made this decision in order to have enough reviews to train the classifiers; however, different conditions might be best classified with their own independent models and this merits more exploration. Finally, we implemented feature selection based on the reported usefulness of the reviews (this is contributed by website users). This method by far proved the have the best results and added about 25% accuracy in all cases. If this information is available this is a great way performing feature selection for drug reviews as it relies on the collective knowledge and experience of all the online users that are reading the drug reviews.

While this study uncovered a number of interesting qualities of drug reviews and developed successful methods of tailoring feature selection to the specific task of understanding user ratings, more work can be done. First of all, we recommend running these classification and feature selection methods on more data. We acquired a data set of 215,000 drug reviews; however, due to our machine's computational power randomly selected 40,000 of those reviews to focus on. Looking at the entire dataset and adding more to the dataset could provide a lot of interesting insights and could potentially lead to even better performance. When adding more drug reviews to analyze it is important to consider reviews from different online forums. This reviews in this study only came from two websites, Drugs.com and Druglib.com, it is important to diversify this in order to obtain maximum generalizability. Finally, future studies should include multiple different evaluation and comparison metrics because different metrics can shed light on different aspects of classifier performance.

References

[1] P Taylor. Why is it so hard to find an antidepressant that works, 2015.

[2] NIH. Mental health medications, 2015.

[3] C Kresser. The dark side of antidepressants, 2019.

[4] F Graber, S Kallumadi, H Malerg, and S Zaunsedr. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. *Proc. of the 2018 International Conference on Digital Health*.

[5] UCI. Uci machine learning repository, 2019.

[6] *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.

[7] Bo Pang and Lillian Lee. An algorith for suffix stripping. *Program*, 2:130–137, 1980.

[8] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, and et. al. Scikit-learn: Machine learning in python. *Program*, 12:2825–2830, 2011.