Pill Recommendation System

Aditi Goyal (B19EE003), Darshit K Jain (B19EE024), Harsh Rajiv Agarwal (B19EE036)
Github Repo: *PRML-project* , Deloyed at *PRML-Project-Frontend*

Abstract

The safety of a medical product is evaluated through pre-defined protocols and clinical trials. These clinical studies are carried under regulated conditions on a limited number of aspects and within a specified time constraint and may not be able to model the real-life scenarios and the associated potential risks. A preeminent solution to this conundrum can be achieved by looking at post-marketing drug surveillance methods such as drug review analysis to monitor drug-related issues. Users are often looking for stories from "patients like them" on the Internet, which is hard to find among their friends and family. Text mining on drug reviews will be useful not only for patients to decide which drug to take, but also for pharmacy companies and clinicians to improve consumer safety by assisting in the reduction of medication errors and obtain valuable summary of the public feedback. Therefore, we have tried to build a platform where patients and clinicians can search by ailments and drug names which enables them to get drug recommendations and obtain insights into patients' portfolio.

Index Terms

Sentiment Analysis, User Reviews and Ratings, Neural Network, Long Short-Term Memory Network, Random Forest, Bayes Classification, Support Vector Machine, Light Gradient Boosting Machine, Neural Network, LSTM

I. Introduction

WE understood how Data Science and Text Mining have been of significant importance in the health care industry and aim to answer the following questions through our platform: How to use sentiment analysis and predictive modelling to recommend the most effective drugs for the given condition? What is the emotional inclination of users towards a chosen drug?
In this project the main aim is to examine the use of sentiment analysis on drug reviews that can aid in identify new opportunities and challenges for any pharmaceutical business. The project aims at classifying the various reviews on the specified drugs based on their polarity with the aid of their rating.

II. Data Description and Preprocessing

The dataset provides patient reviews on specific drugs along with related conditions and a 10 star patient rating reflecting overall patient satisfaction,the date of recording, and finally the number of other users who found that such review useful.

Preprocessing

The first step was to remove null elements from the dataset. Since there were only *1194* conditions which were reported as null, we dropped them. Removed stopwords using *nltk.corpus.stopwords* baring the common negative words so as to preserve the sentiment of the review. Since we had to finally display top 5 recommended drugs for each condition, conditions with less than 5 drugs were dropped. Thus led to *1,56,432* datapoints. Cleaned up the data further by converting reviews to lowercase, removed all non-letters, HTML tags and finally applied stemming to the data from *nltk.stem.snowball.SnowballStemmer*.
Transformed the data into a sparse matrix representing the entire document using Count Vectorise with maximum features as 5000. Split the data into training and testing in 75:25 ratio.
We trained various machine learning models and deep neural networks to classify the sentiment of review text into 3 main categories (Positive, Neutral, Negative). We created the target labels for sentiment classification by categorizing the rating as follows:

Sentiment	Rating	Sentiment Score
Positive	Rating >= 7	2
Neutral	6 <= Rating <7	1
Negative	Rating <= 5	0

III. Exploratory Data Analysis

Exploratory Data Analysis is primarily used to see what the data can reveal beyond the formal modeling or hypothesis testing task and to provide a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate or not. Since there were a lot of features **(uniqueID, drugName, condition, review, rating, date, usefulCount)**, it was important to understand the important features and the relationship between them. Upon exploring the data, it was found that the number of reviews increased significantly in the years 2015-2017 (ranging up to 40k+ reviews) as compared to less than 20k reviews in the years before. This shows how fast the information was spreading by the means of internet and people actually started relying more on the reviews they saw found online and could relate to it.

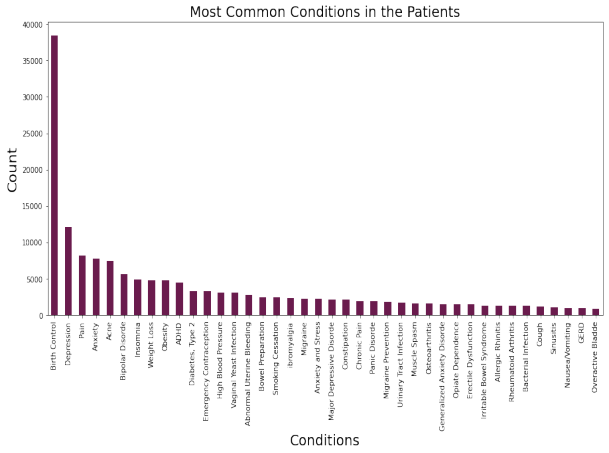

Fig. 1. Most Common Conditions

It was expected that conditions like Birth Control, Depression and Anxiety would have more reviews than something like ADHD or Anthrax.

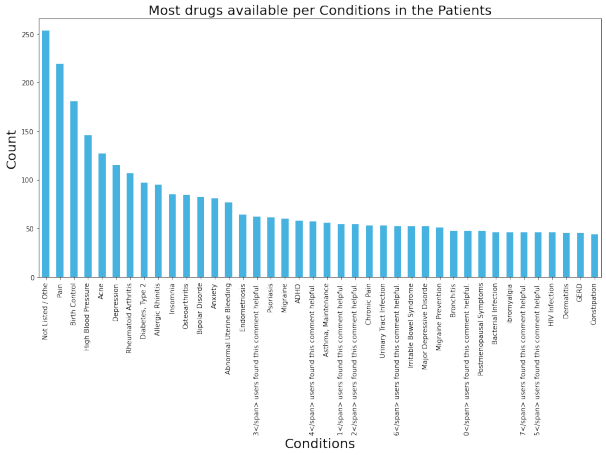

Fig. 2. Most Drugs Available per Condition

The figure illustrates the conditions which have the maximum number of drugs associated with them. There are 150+ drugs available to treat conditions like birth control, high blood pressure and pain.

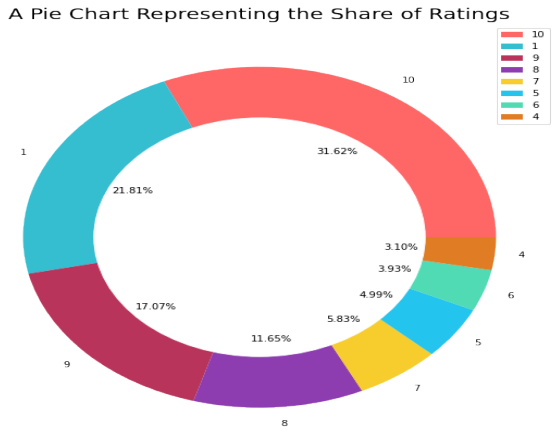

Fig. 3. Rating Distribution

Pie chart shows how ratings from a scale of 1 to 10 are distributed in our dataset. As we can see the majority of users have given a higher rating as compared to lower ones.

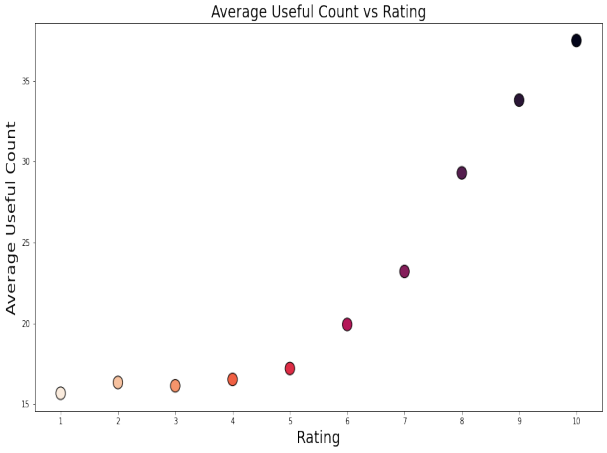

Fig. 4. Rating v/s Useful Count

Users on online health forums can rate the comments of other users through useful count feature. The plot shows a positive increasing trend of the average useful count of a review versus overall rating.

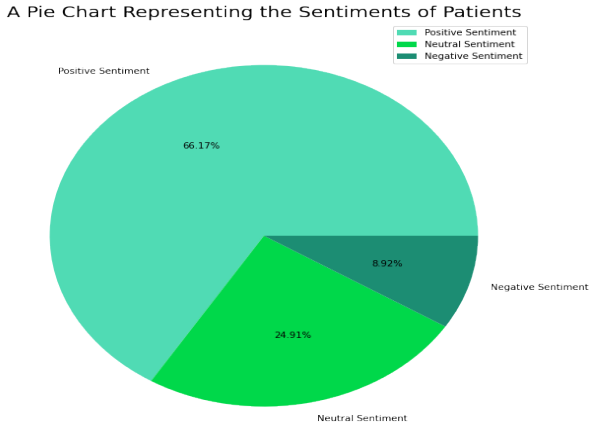

Fig. 5. Sentiment Distribution

A large proportion of the dataset has positive reviews

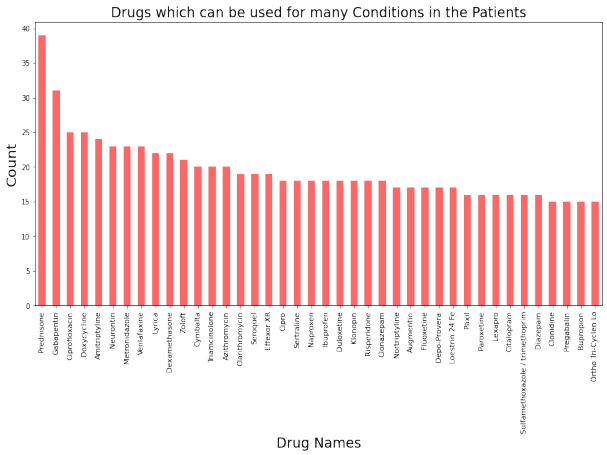

Fig. 6. Drugs which can be used for multiple conditions

IV. Machine Learning Models

A. Random Forest

The training algorithm for random forests applies the general technique of bootstrap aggregating to tree learners. Given a training set X with responses Y, bagging repeatedly selects a random sample with replacement of the training set and fits trees to these samples. After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x or by taking the majority vote in the case of classification trees. This bootstrapping procedure leads to better model performance because it decreases the variance of the model,without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Random forest was fitted with a large range of estimators ranging from 50 to 300. We find that the best accuracy of 46% was achieved at estimators= 80. We can say that multiclass classification could have acted as a deterrent to the performance of Random Forest, since the average accuracy for all trees was around 45% too.

B. Naive Bayes - Multinomial

Multinomial Naive Bayes was used. Since the reviews are not evenly distributed , Multinomial Bayes was preferred over Gaussian. Text Analysis is primarily not done using Bayes Classifier, hence a smaller accuracy of 57.696% is acceptable

C. Light Gradient Boosting - LGBM

LGBM classifier trained on usefulCount instead of the reviews to verify the trend and check the correlation. Remarks - Just usefulCount does not help in giving good predictions. Accuracy of 66.201% .

Predicting sentiment using TextBlob

TextBlob is a python library and offers a simple API to access its methods and perform basic NLP tasks. The sentiment function of TextBlob returns two properties, polarity, and subjectivity. Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. Using TextBlob we predicted the sentiment as a float value once on cleaned reviews and then on the raw reviews. We find that the polarity scores obtained from the raw reviews have better correlation with the actual userRating and Sentiment Score than that from cleaned reviews.

Feature Engineering

Implemented feature engineering and calculated various nuances of the data for exploration to see what importance/effect they have on the predictions. These include unique word count, total words in each review, length of review, count of words with capital letters, count average length of words.

Trained LGBM again on these new features extracted from feature engineering along with the sentiment predicted from TextBlob. We find that this gives a high accuracy of *79.259%* .

Feature Importance

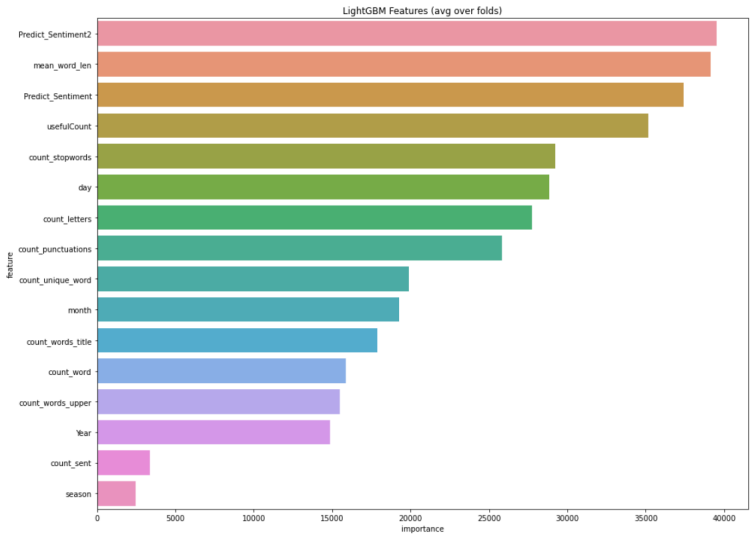

Fig. 7. Feature Importance

Features such as sentiment predicted by TextBlob, mean review and word length and useful count have a high importance which is a valid conclusion.

D. SVM

SVM was tried over multiple hyperparameter C values in the range of [0.003, 100]. Initially at 2 class classification SVM gave an accuracy of 74%. When run over 3 classes, SVM kept crashing after a few minutes. Multiclass SVM models such as OVA and OVO were also implemented by the system was not able to support and crashed.

E. Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. The logistic regression model was trained on C=0.00007 with 200 iterations. The value of C in this model denotes inverse of regularization strength. The smaller values of C indicate stronger regularization similar to support vector machines.

F. Analysis of Conventional Machine Learning Models

The advantages of linear regressions are their properties that make them easy to interpret ; while the weakness is that they only model linear relationships between dependent and independent variables. The strengths of gradient boosting solve linear regression's weakness. Gradient boosting can optimize on different loss functions and provides several hyperparameters tuning options that make the function fit very flexible. Where as the weaknesses of gradient boosting are computationally expensive and less interpretable. Random Forest despite being quite robust did not provide very accurate results. Naive Bayes despite the data not being scaled or gaussian in any way, with uneven distribution of ratings gave a fair accuracy on the test data.

V. Deep Learning Models

A. Neural Network 1

A four hidden layers Neural Network was trained with 300,400,100 and 3 neurons in the respective layers. The first layer was accompanied by BatchNormalization and a Dropout rate of 0.5. Batch normalization applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. The Dropout layer randomly sets input units to 0 at each step during training time, which helps prevent overfitting. The activation function for all three layers was Relu (Rectified Linear Unit) except the last hidden layer which utilized Sigmoid activation function. It achieved a validation accuracy of 84.91%. However, the testing accuracy was 63.70% which indicates that the model was not able to generalise well.

B. Neural Network 2

A two hidden layers Neural Network was employed this time with 1024 and 3 neurons in the respective layers. Here, a pre-trained token based text embedding trained on English Google News 200B corpus from tensorflow hub was used. The *embedding* text has 128 dimensions and corpus is 700B. This model achieved a validation accuracy of 77.32% and a testing accuracy of 74.89% at 50 epochs. However, at higher epochs it enhanced to 77.49%.

C. LSTM

Long Short Term Memory is a kind of recurrent neural network. LSTM can by default retain the information for long period of time. LSTMs are explicitly designed remember information for long periods of time and their default behavior, not something they struggle to learn! Here, a two hidden layers model comprising of a single LSTM layer with 40 units and a Dense Layer of 3 neurons was applied with softmax activation layer and an embedding layer. Embedding layer enables us to convert each word into a fixed length vector of defined size. The resultant vector is a dense one with having real values instead of just 0's and 1's. The fixed length of word vectors helps us to represent words in a better way along with reduced dimensions. It was trained for 30 epochs and achieved a validation accuracy of 83.97% and a testing accuracy of 83.62%

D. Text CNN

Convolutional layers apply a convolution operation to the input, passing the result to the next layer. A convolution converts all the pixels in its receptive field into a single value. The most common type of convolution that is used is the 2D convolution layer and is usually abbreviated as conv2D. A filter or a kernel in a conv2D layer "slides" over the 2D input data, performing an elementwise multiplication. As a result, it will be summing up the results into a single output pixel. Here, an embedding layer was applied with 12000 features each represented by a 300 dimension vector. This input was then feeded to the Convolution-2D layer with Relu activation function. This was followed by a Max Pooling layer and Dropout Layer. Lastly, a Linear transformation was applied using varied filter sizes. Adam optimizer and Cross entropy loss functions were used while training. The model was run for 6 epochs and achieved a testing accuracy of 83.71%. This did not enhance further on increasing the epochs.

Type	Classifier	Accuracy%
ML	Random Forest	43
	Naive Bayes	57.949
	Logistic Regression	66.3
	LGBM	79.235
DL	Neural net 1	63.7
	Neural net 2	77.49
	LSTM	84
	Text CNN	83.71

VI. Emotion Analysis

Using the review for each drug besides sentiment analysis (Positive/Neutral/Negative) , Emotion analysis was also performed to understand human behaviour and underlying opinion. The *NRC Emotion Lexicon* is a list of 14k unigrams and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive).

This data for each drug will be displayed in a pi chart, to show the emotions associated with people taking the drug. With the

Word-Emotion Association, we tracked emotions in the drug reviews. We matched every word in the review text with unigram in the lexicon file and increased the count of respective emotions for each review.Finally, we took an average of the aggregated count for each drug to show the overall distribution.

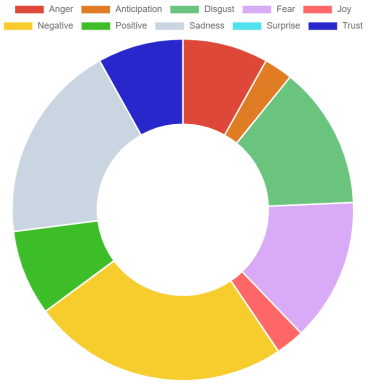

Fig. 8. Emotion Analysis Pichart for each drug

	Drug	Condition	Reviews	Rating	Website	anger	anticipation	disgust	fear	joy	negative	positive	sadness	surprise	trust
232363	adderall	adhd	add with mild ocd as well. the med is very eff...	8.0	webmd	1	0	1	1	0	3	1	1	0	1
188173	generess fe	birth control	after reading everyone's horrifying posts i wa...	9.0	webmd	1	6	0	1	1	1	4	0	3	3
14335	etonogestrel	birth control	"i had the implanon after i had my son. i was ...	2.0	drugs.com	0	0	0	0	0	0	0	0	2	0
198901	benzonatate	cough	i have taken the meds for one day and feeling ...	10.0	webmd	0	0	1	0	0	1	0	0	0	0
92841	mucinex d	cough and nasal congestion	"twice this year i have had a cold and used th...	8.0	drugs.com	1	2	0	1	0	2	3	0	0	3

Fig. 9. Emotion Analysis
A score for the 8 emotions detected is given for each review

VII. Harvard Dictionary Sentiment Analysis

Analysing emotions of a review using the Harvard emotional dictionary which has opinionated words was another way to include the sentiments of a review. This was helpful as there were a total of 11788 predefined words, along with the emotion associated with them. The General Inquirer is basically a mapping tool. It maps each text file with counts on dictionary-supplied categories - Positive and Negative.
Using the values of Harvard Dictionary we find out the number of positive and negative words in a review. Calculate the positive ratio by dividing number of positive words by total words classified in each review. We convert this data to Count Vectorise. Finally the Harvard Sentiment List is given score according to the positive ratio value. If ratio greater than 0.6 then positive (2), if between 0.6 and 0.4 then neutral (1) and if less than 0.4 then negative sentiment (0)

VIII. Classifier Combination - Voting

We need to combine all the predictions from the array of classifiers and predictive methods used. Each ML model was given a weight in proportion to its accuracy with respect to other ML models. Similar logic was followed for DL models. This gives more weight to better performing classifier each time without having to hard-code the weights. Predictions from LGBM and Harvard Sentiment Prediction were added directly since they represent separate individualistic predictions. All these 4 components were summed and multiplied by the usefulCount of the respective review.
We clubbed the drugs by their conditions and a final prediction for each drug was calculated by normalising all the predictions for each condition.

IX. Conclusion

In this project a sentiment analysis was performed on the drug review dataset using various conventional and deep learning models to evaluate their performance. It was observed that the deep learning models performed better. The accuracy seemed to be enhanced when the vector representation were used in any method. Unequal class distribution was presumed to be a major issue in training and testing the dataset and thereby yielding lower accuracy. The presence of lesser dataset could have also caused to the significant reduction in the performance of the models than what was expected.

References

[1] http://kdd.cs.ksu.edu/Publications/Student/kallumadi2018aspect.pdf
[2] https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1223context=scschcomdis
[3] https://www.ukessays.com/essays/information-technology/sentiment-analysis-on-drug-reviews-new-opportunities-and-challenges.php
[4] https://www.kaggle.com/mlwhiz/multiclass-text-classification-pytorchTrain-TextCNN-Modelhttps://www.kaggle.com/mlwhiz/multiclass-text-classification-pytorchTrain-TextCNN-Model

Contribution

- Aditi Goyal(B19EE003) - Worked on iterating different preprocessing aspects and best in-depth exploratory data analysis. She implemented models of Countvectorise vs Tfidf, LGBM with feature engineering and Naive Bayes. She worked on emotion analysis and configured the voting algorithm.
- Darshit Jain (B19EE024) - Worked on exploratory data analysis. He worked extensively on configuring SVM model and finding appropriate hyperparameters for Random Forest. He also implemented the Harvard Sentiment Dictionary. He is responsible for deploying the frontend for the project.
- Harsh Agarwal (B19EE036) - Worked on implementing the 5 elaborate Deep Learning models with embedding alternatives and Logistic Regression besides hyperparameter tuning. He helped with the emotional analysis code and the voting algorithm.