Drug Recommendation System

A recommendation system for doctors based on patient reviews

Ayushi Prasad
MT2023145
IIIT Bangalore
Bengaluru, India
Ayushi.Prasad@iiitb.ac.in

Anant Pandey
IMT2020524
IIIT Bangalore
Bengaluru, India
Anant.Pandey@iiitb.ac.in

Deepanjali Ghosh
MT2023119
IIIT Bangalore
Bengaluru, India
Deepanjali.Ghosh@iiitb.ac.in

Drishti Gupta
MT2023099
IIIT Bangalore
Bengaluru, India
Drishti.Gupta@iiitb.ac.in

***Abstract*—This project explores the feasibility of utilizing customer reviews to develop a drug recommendation system. The project investigates sentiment analysis techniques to analyze the vast amount of textual data present in customer reviews of medications. By identifying positive and negative feedback on factors like effectiveness and side effects, the system can learn user experiences and recommend drugs with a higher likelihood of patient satisfaction. The report discusses the potential benefits and limitations of such a system, emphasizing the importance of its role as a supplementary tool for medical professionals and not a replacement for proper diagnosis and prescription practices.**

***Index Terms*—Customer Reviews, sentiment analysis, patient satisfaction**

I. Introduction

This report explores the development of a drug recommendation system that analyzes real-world experiences reflected in customer reviews. By leveraging sentiment analysis techniques, the system can extract insights from vast amounts of textual data. By identifying positive and negative feedback on factors like efficacy, side effects, and ease of use, the system can learn user experiences and recommend drugs with a higher likelihood of patient satisfaction.

II. Purpose

In medicine, when there is a particular research for a cure for any treatment of a disease, as soon as the tests are conducted, and the new medicine is rolled out, we can consider that there is a new cure for that disease that did not exist before. However, on similar lines, there is also a new problem that doctors face, which is of the collective impact of the medicine, and how it actually effects the people. There is a significant amount of testing done beforehand to make sure the medicines are safe. There is still, a need for a feedback from the patients about the ground impact, as it becomes relevant for future studies, and also understanding the integrity of the standards that are approved worldwide. But the big issue for approaching this issue is the amount or kind of data that determines the actual impact of the medicine. In today's age, however, we have a leverage of interconnectivity across the world, which enable online sale or purchase of medicine, and the reviews collected by these users can also be considered as a personal feedback, on the impact the medicines have made. Our project is a very high level model that tries to measure the sentiment of the users relating to the impact of the medicine. There are further prospects, which are related to studying the impact of the side effects of the medicine, that might have come up in a closely related study, and also combine that with customer reviews to get how serious those conditions are, or even more importantly how rare or frequent they are. But our project is more related to the users giving positive or negative reviews. and understanding the effectiveness of the medicines, which can be used by doctors to recommend medicines to patients, apart from the knowledge of the symptoms, and conditions the patients suffer from. This would ease out the doctor's dilemma about recommendation, and actual impact of the medicine.

Identify applicable funding agency here. If none, delete this.

III. Project Flow

In this project, we have used a publicly available dataset to build a review system, which predicts rating from a given review, and this review can be used to determine the effectiveness of a medicine, based on the impact it has had on consumption. We have plotted various columns available in the dataset, to compare how many different conditions are possible for a given medicine, and did data analysis to refine the data for predictions. Then we tried to use different models, starting from simple ones, to others which required more training. Finally we compared with a pre-trained model to compare the results obtained. We also stored the average rating for a medicine and a given condition and updated the value for the given input, and then reordered the values predicted. These would give the top five medicine for each condition, and would update the value for each given input. And we also propose the more granular approach to get the side-effects for the medicine and incorporate that into our dataset, to get more accurate prescription based on the mention of the side-effects in the reviews.

IV. Data Exploration

For this project, we have used dataset from University of California, Irvine ML repository. We have refined this dataset using Exploratory analysis to obtain

978-1-7281-4387-3/19/$31.00 © 2019 IEEE

V. Exploratory Data Analysis

A. Dataset

We got the dataset from University of California,Irvine ML Repository.This dataset consists of columns userId, drugName, condition, review, rating, date and usefulCount. It has 215063 rows and every user has given 1 review only.It has 916 conditions.Ratings are ranging from 1 to 10.

B. No of drugs per conditions(top 20)

We have many drugs per condition, and we have plotted how many drugs for each of the conditions are available in the dataset.

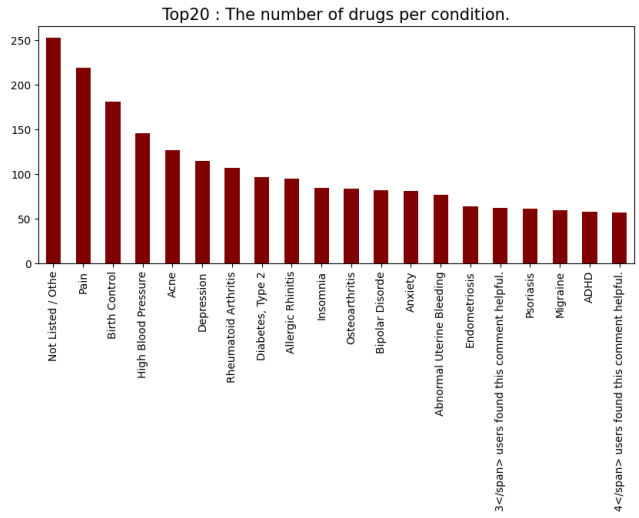

Fig. 1. Number of drugs per condition

We can see that for pain, high blood pressure and birth control, we have the highest number of available drugs, and for others we have similar amount of drugs, as can be described from the graph.
For the following graph, we have plotted the most common conditions, that is the conditions with highest number of available rating values, and we can see that birth control have the highest number of rating values, and others have significantly less frequency.

C. Top 10 most common conditions

Similarly, we have also plotted the top ten most common drugs, that is drugs with maximum number of reviews, and we can see that top three drugs are levonorgestral, etonogestral and norethindrone, which have a high number of ratings, and the number of ratings decrease gradually for other drugs.

D. 10 Most common Drugs

- In this analysis, we aimed to identify the most frequently mentioned drugs in our dataset. To achieve this, we plotted a bar chart illustrating the top 10 most common drugs.

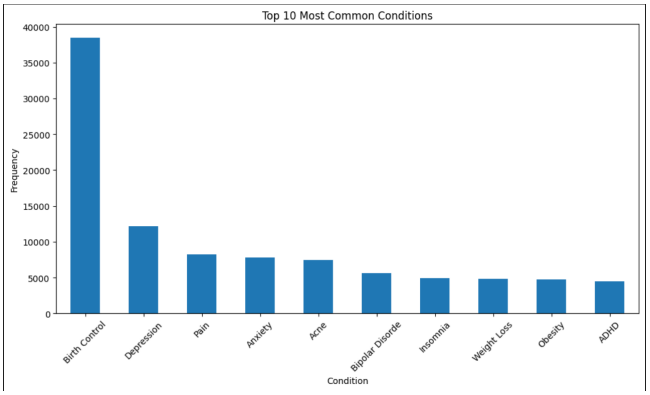

Fig. 2. Top 10 most common conditions

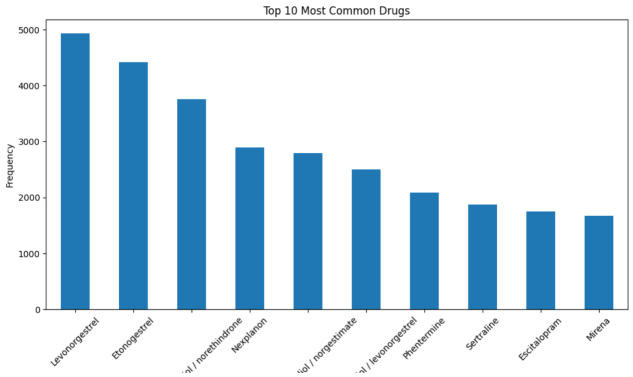

Fig. 3. 10 Most common Drugs

- We used the value counts method on the drugName column of our DataFrame (df). This method counts the occurrences of each drug name, allowing us to determine their frequency.
- From the list of all drugs, we selected the top 10 based on their frequency of occurrence.
- Using Matplotlib, we created a bar chart to represent these top 10 drugs.

E. Count of rating values

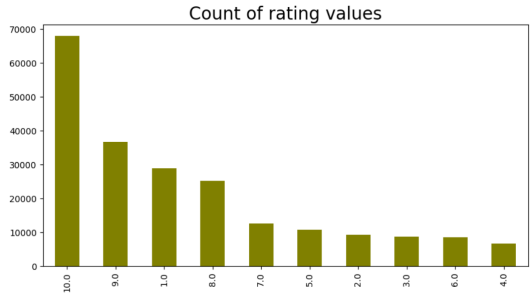

Fig. 4. Count of rating values

To understand the distribution of rating values in our dataset, we created a bar chart that displays the frequency of each rating.

F. Mean Rating per year

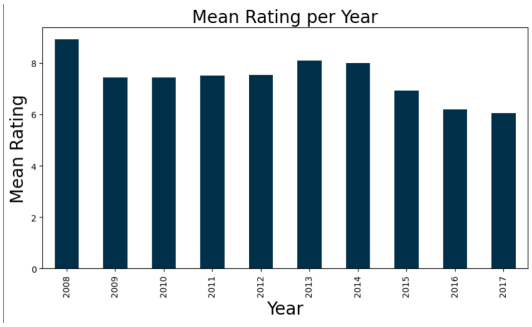

Fig. 5. Mean Rating per year

To analyze the trend of ratings over time, we created a bar chart that displays the mean rating for each year.

G. Total Missing Values

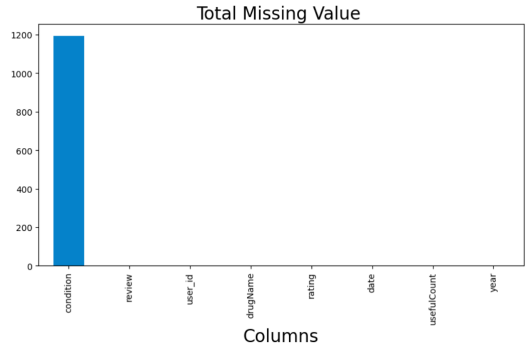

Fig. 6. Total Missing Values

To identify and understand the extent of missing values in our dataset, we created a bar chart that displays the total number of missing values for each column.

VI. Data Balancing

A. Upsampling

We employed upsampling to address the imbalance in our ratings column, which was skewed towards ratings of 9, 10, and 1. By randomly duplicating instances from the minority classes until they reached the same count as the majority class, we effectively balanced the dataset, ensuring each rating class had equal representation for more robust model training.

B. Downsampling

To rectify the imbalance in our ratings column, characterized by an overrepresentation of ratings 9, 10, and 1, we implemented downsampling. This involved randomly removing instances from the majority classes until all rating classes were equally represented in the dataset.

C. SMOTE

In our efforts to mitigate the imbalance in the ratings column, particularly the overrepresentation of ratings 9, 10, and 1, we utilized SMOTE. By generating synthetic instances for the minority classes, SMOTE augmented the dataset to achieve a balanced distribution of rating classes.

VII. Data Preprocessing

A. Basic text preprocessing

- Drop Null Rows: Removed any rows containing null values to ensure completeness and avoid errors during analysis.
- Lower All Words: Converted all text to lowercase to maintain uniformity and avoid case-sensitive discrepancies.
- Remove Special Characters and Punctuations: Stripped special characters and punctuation marks to simplify the text and reduce noise.
- Remove Numbers and Alphanumeric Words: Eliminated numbers and words containing both letters and digits to focus on purely textual content.
- Remove Numbers and Alphanumeric Words: Eliminated numbers and words containing both letters and digits to focus on purely textual content.
- Remove Multiple Spaces: Replaced multiple consecutive spaces with single spaces to normalize spacing.
- Remove URLs: Removed URLs to eliminate links and irrelevant references.

B. Lemmatization

- As part of our text preprocessing efforts, we implemented lemmatization to standardize the words in our dataset. Lemmatization is a crucial step in natural language processing (NLP) that reduces words to their base or root form, known as a lemma. This helps in reducing the complexity and variability in the text data, making it more suitable for analysis.
- Each sentence in the text data was tokenized into individual words using word tokenize, which helps in isolating words for lemmatization.
- The lemmatized words were then rejoined to form the processed sentence.

C. Tokenizer

- To prepare our textual data for machine learning models, we implemented tokenization using the Tokenizer class from the Keras library. Tokenization is a crucial step in text preprocessing that converts text into numerical tokens, which can be effectively used as input for various models.
- We converted all the reviews in the 'review' column of our DataFrame into a list format. This transformation is necessary for passing the data to the tokenizer's fit on texts method.
- The fit on texts method was called with the list of reviews. This step builds the vocabulary index based on

the frequency of words in the text data. Each unique word is assigned a unique integer index.
- We used the texts to sequences method to convert each review into a sequence of integers. Each integer in the sequence corresponds to a word in the review, as per the tokenizer's vocabulary index.
- To prepare our text data for model input, we implemented padding sequences.
- Padding ensures that all input sequences are of the same length by adding zeros (or any specified value) to the end (or beginning) of the sequences.

D. Glove Word Embeddings

- GloVe (Global Vectors for Word Representation) is a pre-trained word embedding model that transforms words into dense vector representations based on their co-occurrence statistics in a large corpus.
- By using GloVe embeddings, we leveraged pre-trained vectors that capture semantic meanings and relationships between words, leading to improved model performance.

VIII. Models Used

A. Recurrent Neural Network

RNNs process inputs sequentially. This is important in tasks where context or order in data points is important, such as sentences in text or lists in video.RNNs have the ability to sequentially recall existing inputs through their internal state, which is updated at each time step This is through feedback to neural networks some results re-enter the network with subsequent cases.The same weights are used at different time steps, which allows the RNN to generalize to different positions in the input sequence. This parameter sharing reduces the total number of parameters compared to a fully connected network dealing with the same sequence.

We created a Sequential model named "RNN" using TensorFlow and Keras.We added 3 layers embedding layer, SimpleRNN layer and Dense layer.Embedding layer transforms integer-encoded tokens into dense vectors.SimpleRNN layers utilized two SimpleRNN layers with relu activation and appropriate units.Dense layer output layer with softmax activation for multiclass classification.We Compiled the model with Categorical cross-entropy loss function for multiclass classification,Adam optimizer for efficient gradient descent. Accuracy metric to evaluate model performance.We trained the model using class weights to handle imbalanced data and improve model learning,trained on training data with batch size 64 and 2 epochs,validated on test data to assess generalization performance.We Calculated and printed the model score (loss and accuracy) on the test data to evaluate model performance.

Limitations: Training of RNNs can be complicated by the gradient problem of missing fluxes and explosions, where the gradients may be too small or too large, making the learning unstable.Basic RNNs may struggle to recognize long-term references due to their limited memory.RNNs can be computationally intensive and slow compared to feedforward neural networks because sequences are processed one step at a time. Accuracy achieved: 42 percent (on test data)

B. Long Short Term Memory

At the heart of the LSTM's ability to rely on long-term memory is its memory cell, a structure that can hold information for long periods of time.LSTMs use three types of gates to control the flow of information: Input Gate: Controls how much new information from the current input is added to the memory cell. Forget Gate: Decides how much of the previous cell state should be retained and discarded. Output Gate: Specifies the amount of information in the memory cell to be transferred to the next time step.

We created a LSTM layer with 150 nodes and a dense layer with 50 nodes and activation function "relu".In output layer we have 10 nodes and activation function "softmax".Loss used is cross entropy loss.Trained the model on training data with batch size of 128 and 3 epochs.Validated on test data to assess generalization performance.Calculated and printed the model score on test data. Accuracy achieved: 74 percent (on test data)

IX. Top 5 Drug Recommendations

In our project, we implemented a system to predict review ratings and provide personalized medicine recommendations based on user input. We developed a function, predict review rating, which takes a user-provided text review and converts it into a sequence of tokens using a pre-trained tokenizer.These token sequences are then padded to ensure uniform length before being fed into a trained LSTM (Long Short-Term Memory) model.The model predicts the rating of the review on a scale of 1 to 10. To add an element of randomness and robustness, we occasionally adjust the predicted rating based on the model's prediction confidence. We maintain a matrix with medical conditions as rows and medicines as columns. This matrix is updated with new user ratings, normalizing them relative to a neutral score of 5. This approach helps in capturing user sentiment about the effectiveness of medicines for specific conditions. After predicting the review rating, the system accepts user input for the medical condition and medicine name. It updates the corresponding cell in the matrix with the adjusted rating. By sorting the ratings for a given condition, we can identify and recommend the top 5 medicines. This personalized recommendation is based on aggregated user feedback and the predicted effectiveness of each medicine for the specified condition.

X. Comparisons of recommendation of our models with Pretrained Model

We used Zero Shot Classification as a metric to compare our results of RNN and LSTM model.Zero-shot text classification is a task in natural language processing where a model is trained on a set of labeled examples but is then able to classify new examples from previously unseen classes.Our results were same as Zero Shot Classification.

XI. Impact Of User Reviews on Recommendation

Continuous negative reviews lead to a decrease in average rating for a medicine.Over time, this decline can result in the medicine being excluded from the top 5 recommended list. Continuous positive reviews contribute to an increase in the average rating for a medicine.This upward trend can lead to the medicine gaining a position in the top 5 recommended list over time.

XII. Future Scope

A. Implementing BERT for Deeper Insights

BERT is a State-of-the-art NLP model capable of understanding context in text Pre-trained on a large corpus and fine-tuned for specific tasks.Advantages of Using BERT are enhanced understanding of language used in patient reviews,improved sentiment analysis leading to more accurate drug ratings For implementation Fine-tune BERT on healthcare-specific datasets including patient reviews and compare performance against current LSTM and RNN models.

B. Specialized Feature Handling: Side Effects

Current Limitations: Reviews mentioning side effects are treated the same as other reviews Proposed Feature: Side Effect Weighting Detect mentions of side effects using keyword extraction or NLP classification,assign negative weightage to reviews mentioning side effects. Impact: Prioritize drugs with fewer harmful side effects in recommendations to increase the safety and satisfaction level of recommended drugs.

XIII. Acknowledgement

We would like to express our sincere gratitude to Dr. Raghuram Bharadwaj for his guidance, support and valuable suggestions throughout this project. His expertise and encouragement were invaluable in developing our approach and ensuring the success of the project. We also sincerely thank Pavan Thanay Muthyala, the teaching assistant, for his continued support and constructive comments, which greatly facilitated our learning and development. Their combined efforts and dedication were instrumental in completing this project.

References

1) https://archive.ics.uci.edu/dataset/462/drug+review+dataset+drugs+com
2) https://www.geeksforgeeks.org/amazon-product-review-sentiment-analysis-using-rnn/
3) https://www.youtube.com/watch?v=fiqo6uPCJVIt=77s
4) https://www.youtube.com/watch?v=JgnbwKnHMZQt=1735s