Medicine Recommendation System

by

Vishnu Agarwal

2019BCS-072

A report submitted for Summer Project

Bachelor of Technology

in

Computer Science Engineering

2019-2023

विश्वजीवनामृतं ज्ञानम्

ATAL BIHARI VAJPAYEE-

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT

GWALIOR - 474015, MADHYA PRADESH, INDIA

Thesis Certificate

I hereby certify that the work, which is being presented in the report, entitled Medicine Recommendation System, for Summer Project in **Bachelor of Technology** in **Computer Science Engineering** and submitted to the institution is an authentic record of my/our own work carried out during the period *May-2021* to *August-2021* under the supervision of Dr. Saumya Bhadauria. I also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Date: Name and Signature of the Candidate

This is to clarify that the above statement made by the candidate is correct to the best of my/our knowledge.

Date: Name and Signature of the Research Supervisor

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Candidate's Declaration

I hereby certify that I have properly checked and verified all the items as prescribed in the check-list and ensure that my thesis is in the prop er format as specified in the guideline for thesis preparation.
I declare that the work containing in this rep ort is my own work. I understand that plagiarism is defined as any one or combination of the following:

(1) To steal and pass off (the ideas or words of another) as one's own

(2) To use (another's production) without crediting the source

(3) To commit literary theft

(4) To present as new and original idea or product derived from an existing source.

I understand that plagiarism involves an intentional act by the plagiarist of using someone else's work/ideas completely/partially and claiming authorship/originality of the work/ideas. Verbatim copy as well as close resemblance to some else's work constitute plagiarism.
I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programmes, experiments, results, websites, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/-sources.
I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report/dissertation/thesis are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable. My faculty supervisor(s) will not be responsible for the same.

Signature:

Name: Vishnu Agarwal

Roll. No: 2019BCS-072

Date: 05-08-2021

Abstract

This Project is about creating a Medicine Recommendation System based on Sentiment Analysis of Reviews using Deep Learning. It will also consist of a comparison between various Deep Learning approaches for the sentiment analysis of medicine reviews.

This Project will also consist of the deployment of the finalized model on a backend server which will be used by a mobile application and a web application. Both Applications will provide the ability for the user to be able to search for a particular condition and get the recommended list of the medicines ranked according to the confidence in the recommendation. Each user could also add their own review for a condition and medicine and request for the addition of any new medicine for a particular condition.

Contents

Chapter 1

Introduction

1.1 Context

This project is the Mini Project(BCS-2999) for BTech in Computer Science, 2019-2023. This project aims to create a Deep Learning Model for ranking medicines based on condition using patient reviews. This project focuses on creating and comparing different approaches for deep learning natural language processing model for the task, and the deployment of the finalized model.

1.2 Problem/Motivation

Medicine Recommendation Problem Problem of Medicine Recommendation System is not a new problem, but has been a problem since the discovery of medicines, following are the important points which compels us to find the solution for this problem:

- According to the statistical data from various sources, one of the most concerned and searched topics on the internet is health information. Many people search about symptoms they are having and the possible drugs for the cure. This have lead to many deaths and many getting from bad to severe symptoms.

- In the current pandemic due to Covid-19, we cannot afford more deaths and more severe patients.

- Most medical practitioners prescribe medicines based on their experiences which is quite limited, so they also often make mistakes, to solve this problem many people have tried to create a global medicine review system to gather reviews and experience of many patients and doctors, but this approach has a major drawback, and that is it is not humanly possible to read through millions of reviews for thousands of medicines for thousands of conditions and understand the ranking of medicines based on a specific condition.

Motivation for Medicine Recommendation System: Due to circumstances, I have been a regular patient myself throughout my life, and also unfortunately our family have first-hand experience with covid-19, so I understand the severity of the situation we are currently in, and also know the importance of being able to get the best recommendation for medicine as possible. This problem that this project is trying to solve will eventually save many lives by providing doctors with a powerful tool to expand their experience, and also save money for many people who would go to the doctor for mild and small symptoms, and get heavily charged. Being a Deep Learning Enthusiast and a part of this Machine Learning Community I believe we can save many lives with this project and also help to lessen the burden on doctors in the current situation.

1.3 Objectives

The objective of research is to create a medicine recommendation system using reviews, and deploy the system on both mobile and web application. Following are all the objectives of the project need to be achieved:

- **Data Preprocessing, Analysis and Visualization**

  (1) There are 2,15,063 reviews in total for different conditions and medicines in the dataset. It will show a detailed analysis of data from many perspectives, to find out the irregular, corrupted, or missing data.

  (2) It will provide Visualization of data in many ways and after each preprocessing step, to have the better understanding of data and also be able to find out the factors that might affect the results after training on model.

(3) It will perform many data correction and preprocessing steps to correct irregularities, remove corrupted data, perform required operations according to the missing data and also normalization of data to minimize the affecting factors on training in further steps.

(4) This step is considered to be one of the most important step in any Natural Language Processing Task, as a small improvement here can lead to a significant improvement in the overall accuracy of our model.

- **Deep Learning Models**

(1) It consists of designing, training and comparison of various deep learning models for the sentiment analysis on our medicine reviews dataset.

(2) It also consists of text preprocessing steps like tokenization and word embedding techniques used for each model.

(3) Selection of the best performing model than further optimizations. Final model will provide sentiment analysis on reviews.

(4) Final score of each review for a particular condition and medicine will be than normalized and with the total value of useful count for each review, it will be used for the final ranking of medicines for each condition.

- **Ranking Generator**

(1) To create a python script to rank the medicines.

(2) It will provide a solution to the ranking problem, i.e for ex- Let a medicine A has many 45 positive reviews and 5 negative reviews and a medicine B has 10 positive and no negative reviews, than which one is better?. Ranking Generator will be able to calculate the probability of getting a better result with medicine A than B.

- **Backend Server**

(1) Final model will be deployed on a backend server. Which will be used by both our Web and Mobile Application.

(2) This backend server will provide http api calls for the addition of reviews, perform sentiment analysis on the review.

(3) It will also directly operate with the database, and perform CRUD api for the applications if required.

(4) It will also create new ranking of medicines for each condition, automatically after a specific amount of updates are performed on database and model.

(5) Administrator will still be allowed to perform forced ranking update.

- **Mobile and Web Application**

(1) It will provide a user interface to search for medicine recommendation based on the condition or symptoms.

(2) It will also provide machine learning powered suggestions based on the condition searched to users.

(3) It will provide an interface to add your own review or request for adding a new medicine in database.

1.4 Research Work flow

According to the research objectives, the report will describe the work flow as below:

Step 1 We are going to use UCI ML Dataset[6] as our dataset for the project, it contains patient reviews for specific conditions and medicines, with also their provided rating and useful counts per review. A detailed data visualization and analysis will be provided as part of the results of the research.

Step 2 Develop three models for sentiment analysis according to three popular approaches for deep learning sentiment analysis, and perform comparison between models and finalize the best performing model.

Step 3 Create python scripts for text preprocessing of reviews before they are provided to sentiment analysis model and a python script to rank the medicines according to the scores obtained from the reviews.

Step 4 Deploy the whole Medicine Recommendation System on backend and create one android and one web app for the users to search for conditions and ask for medicine recommendations for the condition.

Chapter 2

Literature review

2.1 Background

Machine Learning is widely used in Recommendation Systems. A Recommendation System is a system that is used to recommend particular types of items based on certain factors according to the use case. There are many Applications for these systems, like in information retrieval systems, for video/post recommendation used by both youtube and facebook, route recommendation system used in various navigation services etc.

Medicine Recommendation System is also similar to other recommendation system, there have been various researches done in an attempt to create this system, each with their different approaches and targeted users, some of them are:

- **Diabetes Medication Recommendation System**[1]: Authors of this article were able to create a recommendation system based on domain ontology and SWRL for anti-diabetic drugs selection, By thorough analysis, their system first builds ontology knowledge about the drugs' nature attributes, type of dispensing and side effects, and ontology knowledge about patients' symptoms. It then utilizes SWRL and JESS to induce potential prescriptions for the patients. While this is a great work done by the authors, it can only recommend medicines for diabetes only, and to expand this approach for a general domain of common diseases is considerably hard and less feasible. And it becomes almost impossible if we want to recommend medicines based on symptoms alone using this approach.

- **Traditional Medicine Recommendation System**[8]: Authors of this article have created a recommendation system for herbs, to provide traditional herbal medicines to patients, using ontology-based knowledge representation technique that employs Web Ontology Language (OWL) to process and describe data in the ontology. There work helps users to provide herbs according to the disease and their health information. This work is promising but cannot be applied in common use case scenarios, since it is not that easy to perform same approach with millions of artificially made medicines and different diseases. And it also has the same drawback as earlier approach, i.e it cannot recommend medicines based on symptoms alone.

2.2 Analysis

From the above research on Medicine Recommendation System, we get to see different approaches for creating a medicine recommendation system, but from neither one we can directly get easily accessible medicine recommendation for directly entering symptoms. Also above approaches are not much practically possible to implement for global symptoms and medicines. As we can see from above approaches, it is either for a particular diseases or with herbs with less side effects and complexity when compared to artificially created medicines.

Due to this we need a universal approach for medicine recommendation system which could recommend medicines that are easily accessible by users and also safe to take. That is why, we have proposed a new approach using patient reviews for recommending medicines. It reduces the complexity considerably and can be directly used to recommend medicines based on symptoms alone. These recommendations will be for common and mild symptoms and conditions, and as these recommendation will be for medicines which are already targeted for the specific condition only, which makes it safer and easier to recommend as we don't have to consider the compounds each drug and try to make analysis based on that, which compounds will be safe and effective against particular condition like earlier approaches. Recommended Medicines would have already been tested and recommended by their respective companies for that specific condition.

In case of severe symptoms, we recommend only certified medical practitioners to use it to recommend medicines with better accuracy and take advantage of recommendation provided with the wide range of data. This particular system is quite useful in case of medicines for mild symptoms and will help many peoples from preventing high charges from doctor for mild symptoms. It can also save lives by preventing users from using any medicine they get by searching on google for a particular condition.

2.3 Key Related Research

Data Preprocessing NLP[4]:

This article tells us about all the most common and important text preprocessing steps before the training on our Natural Language Processing Model. There are mainly 5 steps of text preprocessing, lower casing, stemming, lemmatization, stop words removal, and tokenization. These steps are performed before sending our text to our model, but there are many steps of data preprocessing before the final text preprocessing.

Word Embedding for NLP[5]:

This article tells us about some common word embedding methods used in Natural Language Processing. There are mainly two most common methods used which are BoW(Bag of Words) and Word2Vec(Word to Vector), there many optimizations which can be used on both methods according to your particular use case. We have used both methods according to the different model we have used for training.

N-Gram and ANN for Sentiment Analysis[3]:

It provides us with the brief overview on how to use a combination of n-gram and artificial neural network for the sentiment analysis on twitter comments. We have used this technique as one of the models for sentiment analysis of our medicine review database.

BiLSTM Model for Sentiment Analysis[11]:

It gives us the detailed information on the method of using BiLSTM Deep Learning Model for the Sentiment Analysis. We have also used this model in our project for performing sentiment analysis on our medicine review dataset.

CNN-BiLSTM for Sentiment Analysis[9]:

This article provides us with the comparison between BiLSTM and CNN-BiLSTM combination Deep Learning Model for the document level Sentiment Analysis. We have also used this model in our project for the sentiment analysis of our medicine review dataset.

Bayesian View of Amazon Resellers[2]:

This article provides us the solution which can be implemented to solve the ranking problem, using a bayesian beta distribution approach to compare between different medicines according to total number of positive and negative reviews for each medicine and condition.

Chapter 3

Methodology

3.1 Mechanism/Algorithm

3.1.1 Sentiment Analysis Models

I have tried 3 Deep learning approaches for the sentiment analysis and finalized the best, following are the 3 models tried:

- **N-Gram Deep Learning Model**: In this model, after all preprocessing and tokenizing the words in the text, we count the frequency of grouped n words and tokenize, each group of n words such that, in our deep learning model the frequency of each token is passed for each review and our deep learning model tries to identify patterns in positive and negative review using the following inputs. Value of n is pre-determined based on analysis of data for different values of n.

- **BiLSTM Deep Learning Model**: In this model, after preprocessing and tokenizing of text, we convert text into list of tokens and this list with padding to a constant length is provided the deep learning model. Here inside the model there is added a Word Embedding Layer to convert each token into a vector for fuzzy clustering and than these list of vectors is further processed by a BiLSTM layer which is a RNN, most widely used to understand the context of the text, and finally a deep neural network is used after this to get the final output from the model.

- **CNN-BiLSTM Deep Learning Model**: This approach is similar to last model, but after the word embedding layer a 2-D Convolution Layer is added which are widely used to extract features from 2-D matrices and after this layer BiLSTM layer and Deep Neural Network is added to get the final output.

3.1.2 Bayesian Beta Distribution Ranking Algorithm

According to this article[2] to compare two medicines we can create a beta distribution for each, then we calculate the probability that a sample from the one distribution is greater than from the other. We can calculate the probability density function of the beta distribution for a medicine and condition by the following formula:-

$$pdf = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\,du} \tag{3.1}$$

where,

$$\alpha = \text{Num of positive reviews} + 1$$

$$\beta = \text{Num of negative reviews} + 1$$

Now according to the article[7], to calculate the probability of a sample from let distribution B is greater than the distribution A, following formula can be used:-

$$Pr(P_B > P_A) = \sum_{i=0}^{\alpha_B - 1} \frac{B(\alpha_A + i, \beta_B + \beta_A)}{(\beta_B + i) B(1 + i, \beta_B) B(\alpha_A, \beta_A)} \tag{3.2}$$

where,

$$B(\alpha, \beta) = \int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\,du \tag{3.3}$$

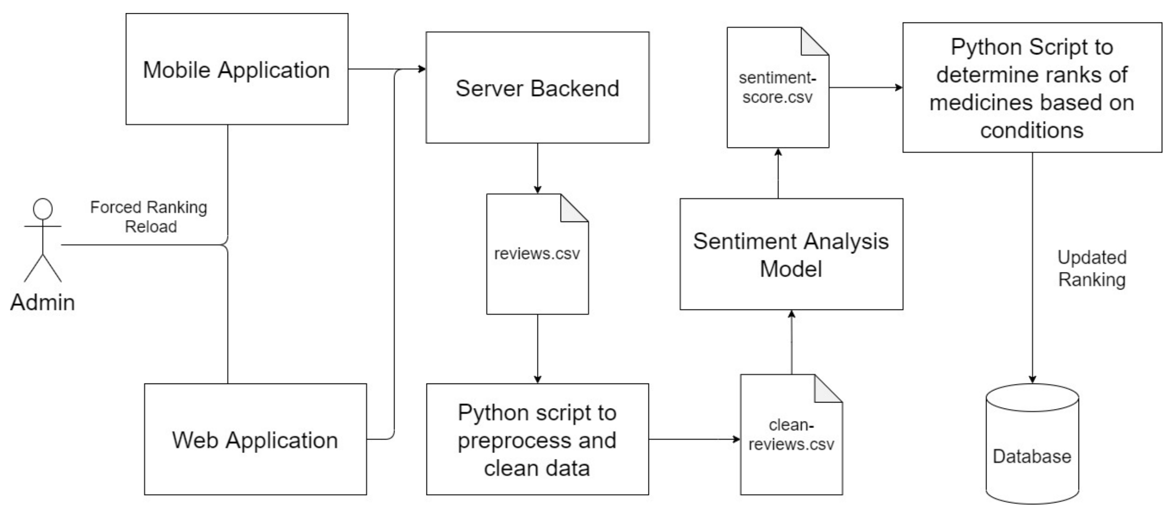
3.1.3 Semantic Search Model

For semantic search we will be using the Universal Sentence Encoder[10] and apply transfer learning to classify any search results between all inbuilt conditions.

The Universal Sentence Encoder[10] first provides us with the vector embedding of size 512 features, for each search query and then our backend will generate inner products with the pre generated features of inbuilt conditions in our system. Here now, each inner product gives us the similarity score between the query and the respective inbuilt condition. Then we will return the top 20 most similar conditions as the best matches for the given query.

3.2 System Architecture

3.2.1 Generate Ranking System Architecture

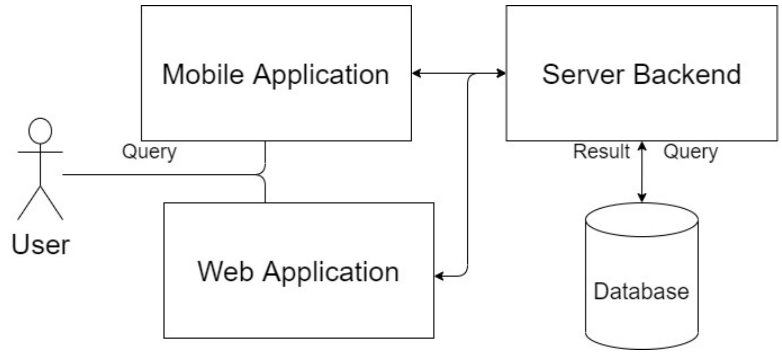


Ranking will be automatically updated and can also be forcefully updated by the Admin. To generate a new ranking, firstly our backend server will get all reviews from the database and create a reviews.csv file which will then be preprocessed by a python script. This process consists of many steps of preprocessing, filtering, and cleaning data.

After preprocessing a new clean-reviews.csv file will be formed which will be passed on to our sentiment analysis model, which will return sentiment scores for each review. Now, these scores per review will be transferred to another python script which will use Bayesian Beta Distribution comparison to generate the ranking. Finally, rankings are updated in the database.

3.2.2 Mobile and Web Application Architecture

The deployment of the medicine recommendation system model will be done, on a backend server, with mobile and web applications for the user interface.



Both user and admin can use the applications. These applications will provide the ability to Search for a specific condition and get the ranking of medicines for that condition.

The process for this will be as follows:-

(1) User search for the symptom or condition.

(2) Recommended conditions available in database according to the searched condition are displayed. Here these recommendations will be given by creating fuzzy word embedding on condition names.

(3) User selects the condition and clicks on get medicines. Query is than transferred to backend and it access database, which returns the ranking for the condition.

(4) Ranking for medicines are displayed on the respective application.

These application will also be used by users for adding their own reviews for specific conditions and medicine. The process for this will be as follows:-

(1) User selects the condition, like previous steps.

(2) Now user click on add review option and then submit the review.

(3) Review is send to the backend server, which adds it to the database.

Chapter 4

Results

This section discusses the various experiments pertaining to the proposed hypothesis and their findings.

4.1 Exploration, Data Analysis and Visualization

4.1.1 Data Understanding

For training of models UCI ML Drug Review Dataset[6] has been used. Sheer size of data is as follows:-

```
All Data shape: [215063, 7]
```

This is the sample of data:-

	uniqueID	drugName	condition	review	rating	date	usefulCount
0	206461	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9	2012-05-20	27
1	95260	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8	2010-04-27	192
2	92703	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5	2009-12-14	17
3	138000	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8	2015-11-03	10
4	35696	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9	2016-11-27	37

These are additional explanations for variables.

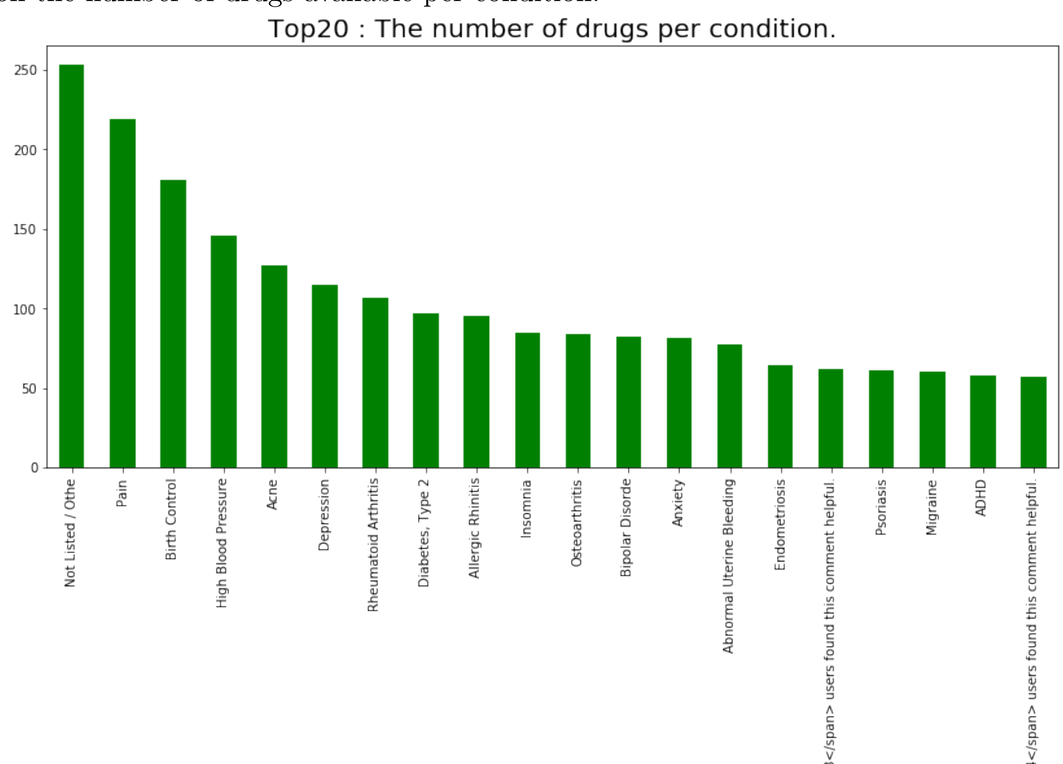
- drugName (categorical): name of drug
- condition (categorical): name of condition

- review (text): patient review
- rating (numerical): 10 star patient rating
- date (date): date of review entry
- usefulCount (numerical): number of users who found review useful

This data has been generated by following method, A patient with a specific condition is provided with a following drug from their respective medical practitioners after that the patient writes a review on the medicine and also provides a rating to that medicine. After that others who found their review useful will upvote their review, increasing the usefulCount value by 1.

4.1.2 Data Visualization and PreProcessing

In our dataset their are total number of 3671 unique drugs and 917 conditions. At an average there are about 4 drugs per condition. Let's visualize it in more details, here are top 20 conditions based on the number of drugs available per condition.

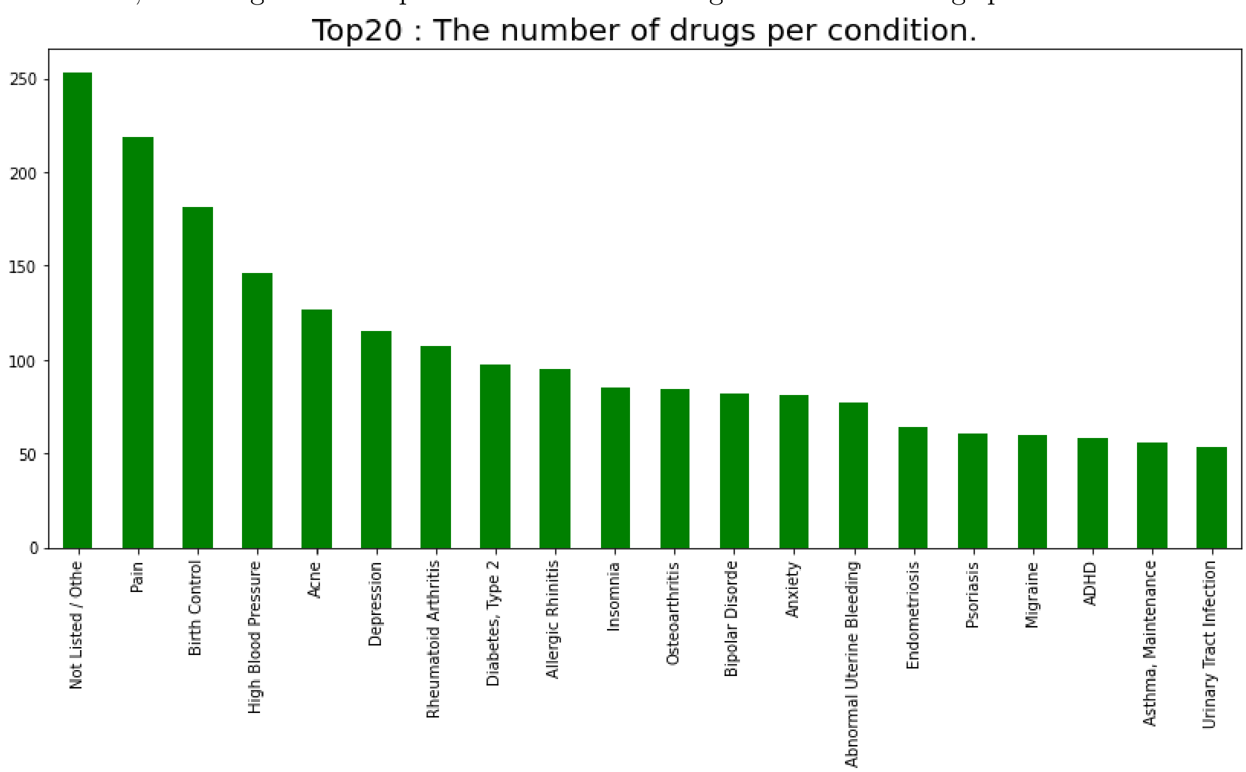


As we can see for top 20 conditions, number of drugs per condition is about 100 for each condition, On the other hand, it should be noted that the phrase ”3¡/span¿ users found this comment helpful” appears in the condition, which seems like an error in the crawling process. I have looked into it to see in more details.

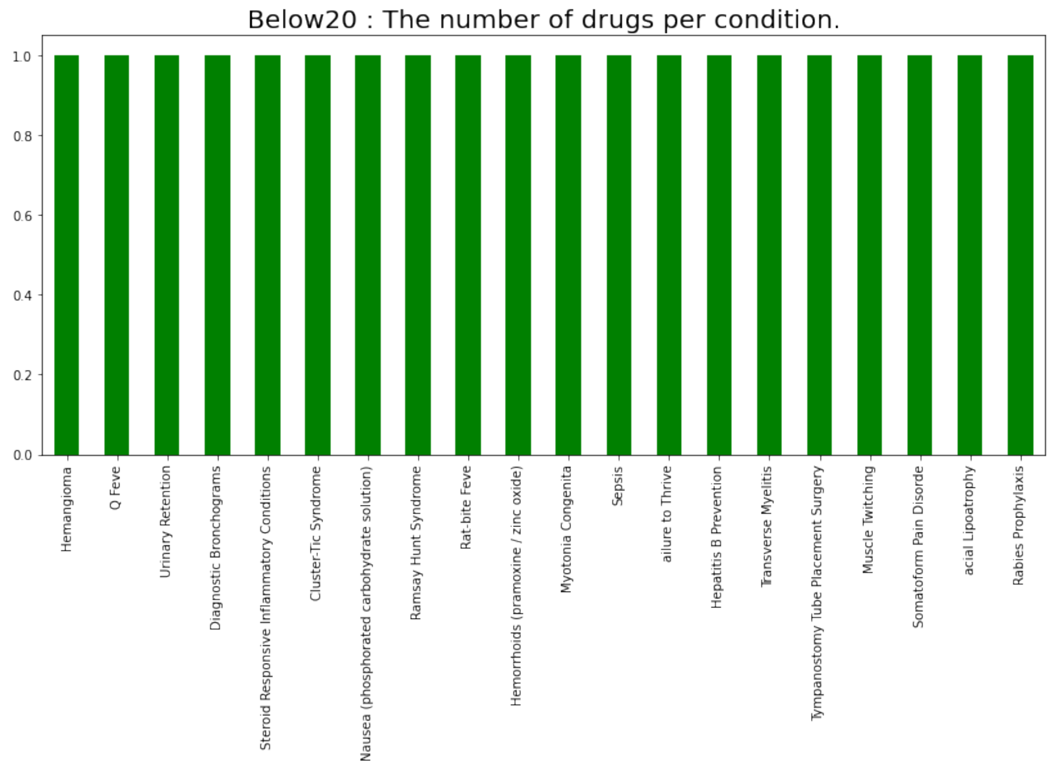
From there I found out there are about 80 conditions with names similar to this. Now we have to ensure the significance of it in our dataset, on further analysis I found, out of total reviews, 2365 reviews are of these types of conditions. Which has the significance of 1.099% in our dataset. Therefore I decided to remove all reviews with these types conditions. After removing them our size of dataset was:-

```
All Data shape: [212698, 7]
```

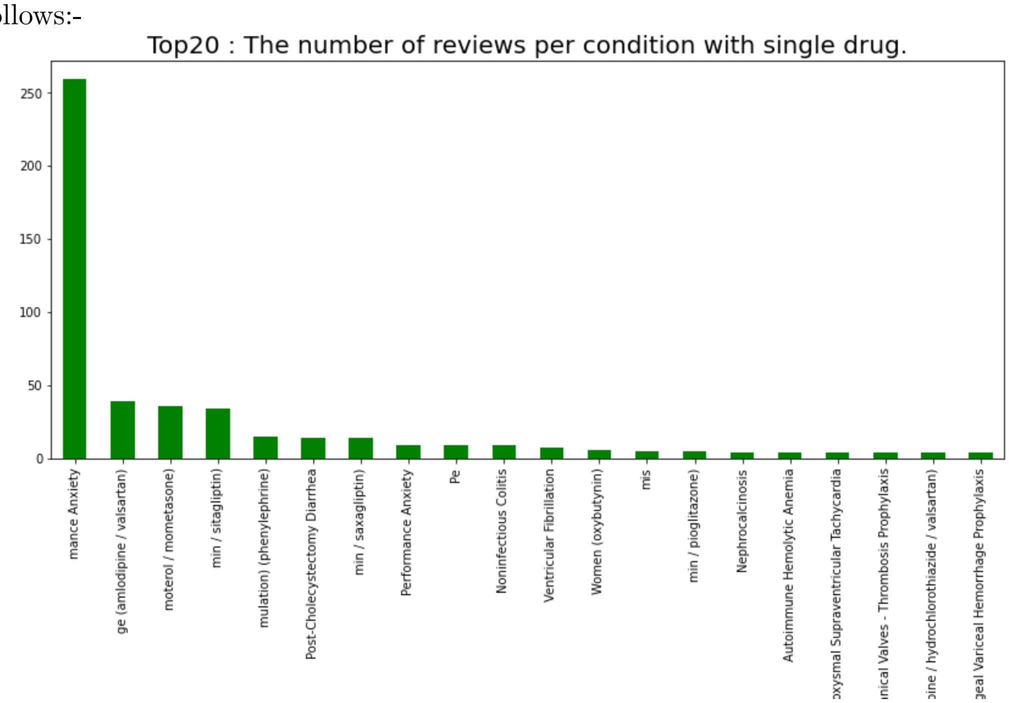
Now, following are the top 20 conditions according to number of drugs per condition:-



Following are the bottom 20 conditions according to number of drugs per condition:-



Now as we can see here, each condition has only single drugs available, for these cases it becomes difficult to recommend medicines as there is only one medicine available from to recommend. Here I decided to take a look at number of reviews per condition which has only one drug, which was as follows:-



From here we can see few of the conditions have quite a lot of reviews, which shows the confidence in these drugs from the patients. So we will be removing dataset of conditions with only one drug and less than 20 reviews. After it, our size of data is as follows:-

```
All Data shape: [212421, 7]
```

Now, let's take a look at reviews. This is the second review in our dataset from index.

```
'"My son is halfway through his fourth week of Intuniv. We became concerned when he
    began this last week, when he started taking the highest dose he will be on. For
    two days, he could hardly get out of bed, was very cranky, and slept for nearly 8
    hours on a drive home from school vacation (very unusual for him.) I called his
    doctor on Monday morning and she said to stick it out a few days. See how he did
    at school, and with getting up in the morning. The last two days have been problem
    free. He is MUCH more agreeable than ever. He is less emotional (a good thing),
    less cranky. He is remembering all the things he should. Overall his behavior is
    better. \r\nWe have tried many different medications and so far this is the most
    effective."'
```

Here, first noticeable parts are HTML strings like \r and \n, also there are parenthesis and capital letters. All these are unnecessary details for understanding the semantic of the review. Therefore we will be removing all of punctuations. Also on further analysis I found that in many reviews for single quotes (') it has been replaced with a string as "'" so we will also be replacing them with single quotes too. After all this, earlier review became this:-

```
my son is halfway through his fourth week of intuniv we became concerned when he began
    this last week when he started taking the highest dose he will be on for two days
    he could hardly get out of bed was very cranky and slept for nearly 8 hours on a
    drive home from school vacation very unusual for him i called his doctor on monday
    morning and she said to stick it out a few days see how he did at school and with
    getting up in the morning the last two days have been problem free he is much more
    agreeable than ever he is less emotional a good thing less cranky he is
```

```
remembering all the things he should overall his behavior is better we have tried
many different medications and so far this is the most effective
```

Now, I looked on word frequency over all reviews, and this was the result:

```
i                 979069
and               566929
the               535550
to                425698
my                383957
                    ...
342am                  1
insertioni             1
quasimoto              1
remediesyeah           1
escitaploprgram        1
Length: 87317, dtype: int64
```

From here what we can see is that some of the words like "i", "my", "and" etc. have very high frequency, and that is most probably because they are also very common words in English Language itself, these types of words are called stop words, which does not provide much of a significance to the actual sentiment of the text. We will be removing these stop words except negative words like "not", "non", "no", "doesn't" etc. Also before that we will be performing **Lemmatization** in this process, we convert words to their simplest form for ex- "have" → "be", "ran" → "run", "eats" → "eat", "ate" → "eat" etc.

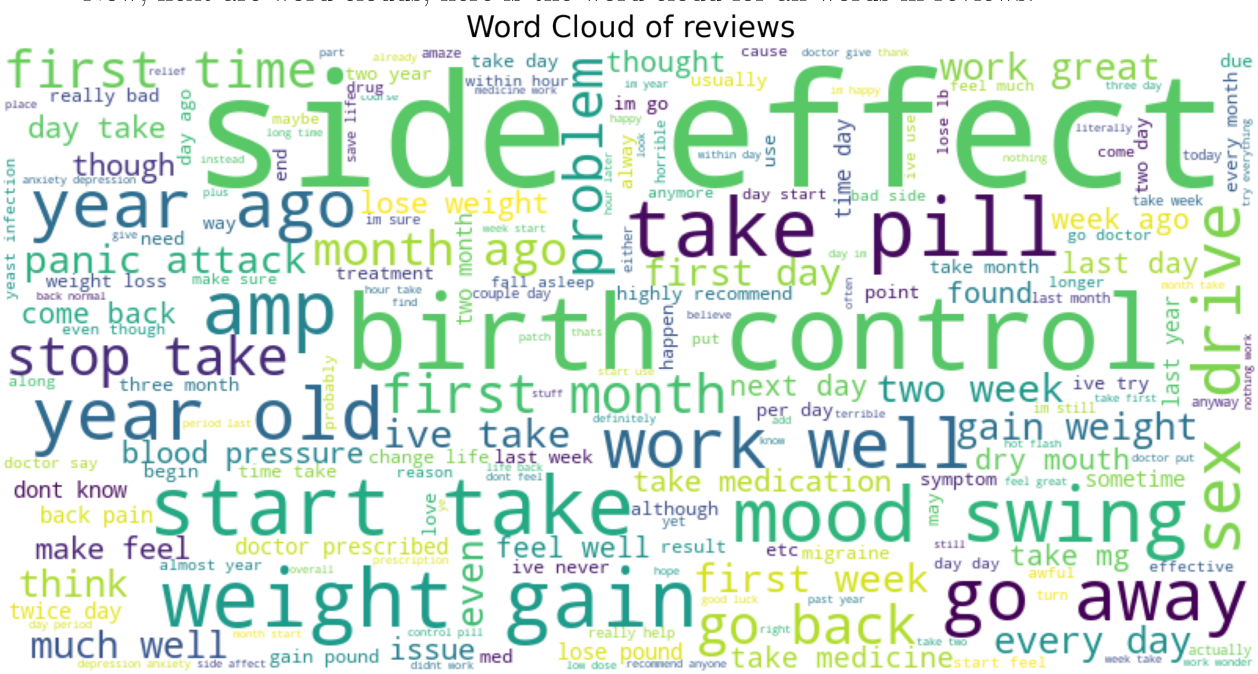
After both of these steps we also removed numbers from our text, as those also does not provide significant amount of effect on the sentiment of the medical review. So, after all these steps our earlier sentence is converted to the following:-

```
son halfway fourth week intuniv become concerned begin last week start take high dose
    two day could hardly get bed cranky slept nearly hour drive home school vacation
```

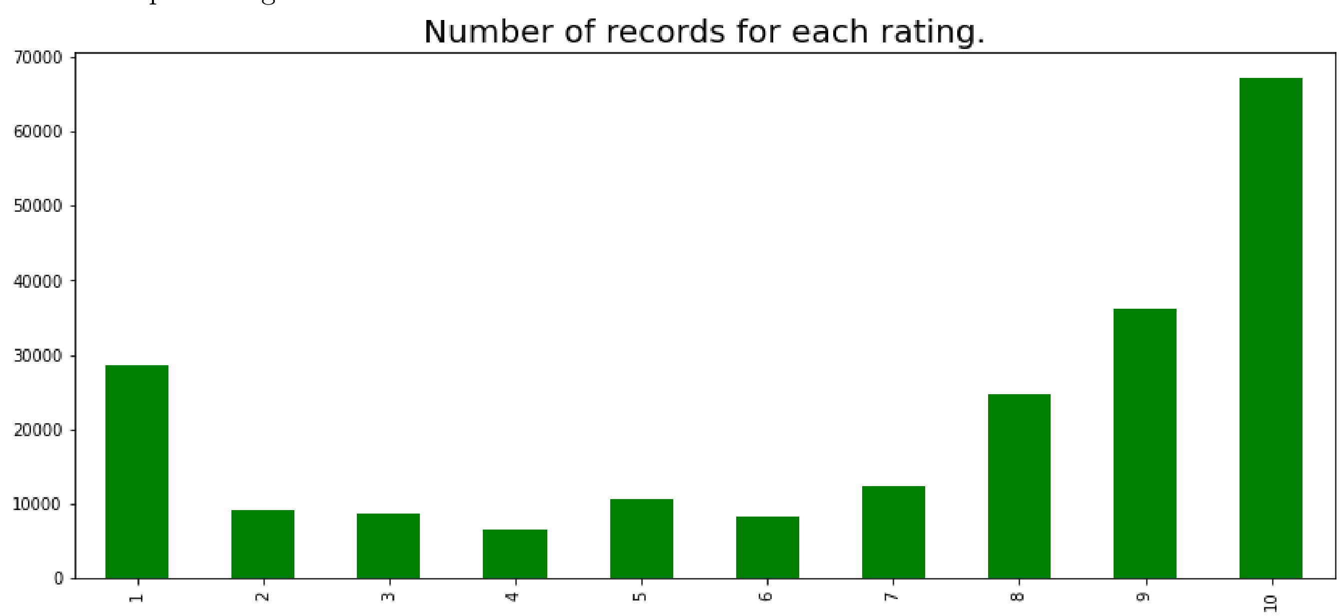
unusual call doctor monday morning say stick day see school get morning last two day problem free much agreeable ever less emotional good thing less cranky remember thing overall behavior well try many different medication far effective

---

Now, next are word clouds, here is the word cloud for all words in reviews:

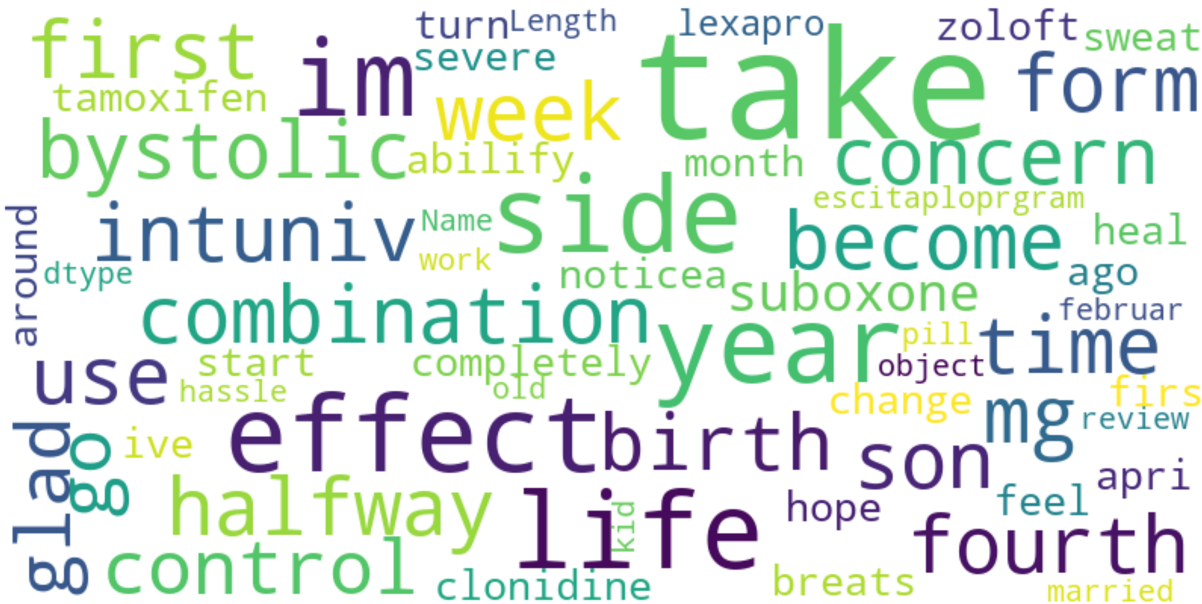


Now, to split our data in positive and negative reviews, we will be looking into number of records per rating value:



From here as we can see most of the reviews are greater than 8 also, so to equalize and to also be able to recommend confidently we will be considering reviews with rating greater than 7 to be positive and below than that to be negative.

After that, now we will look at the word cloud of positive and negative dataset, following is for positive:



And here are of negative:

4.2 Creating and Comparing Models

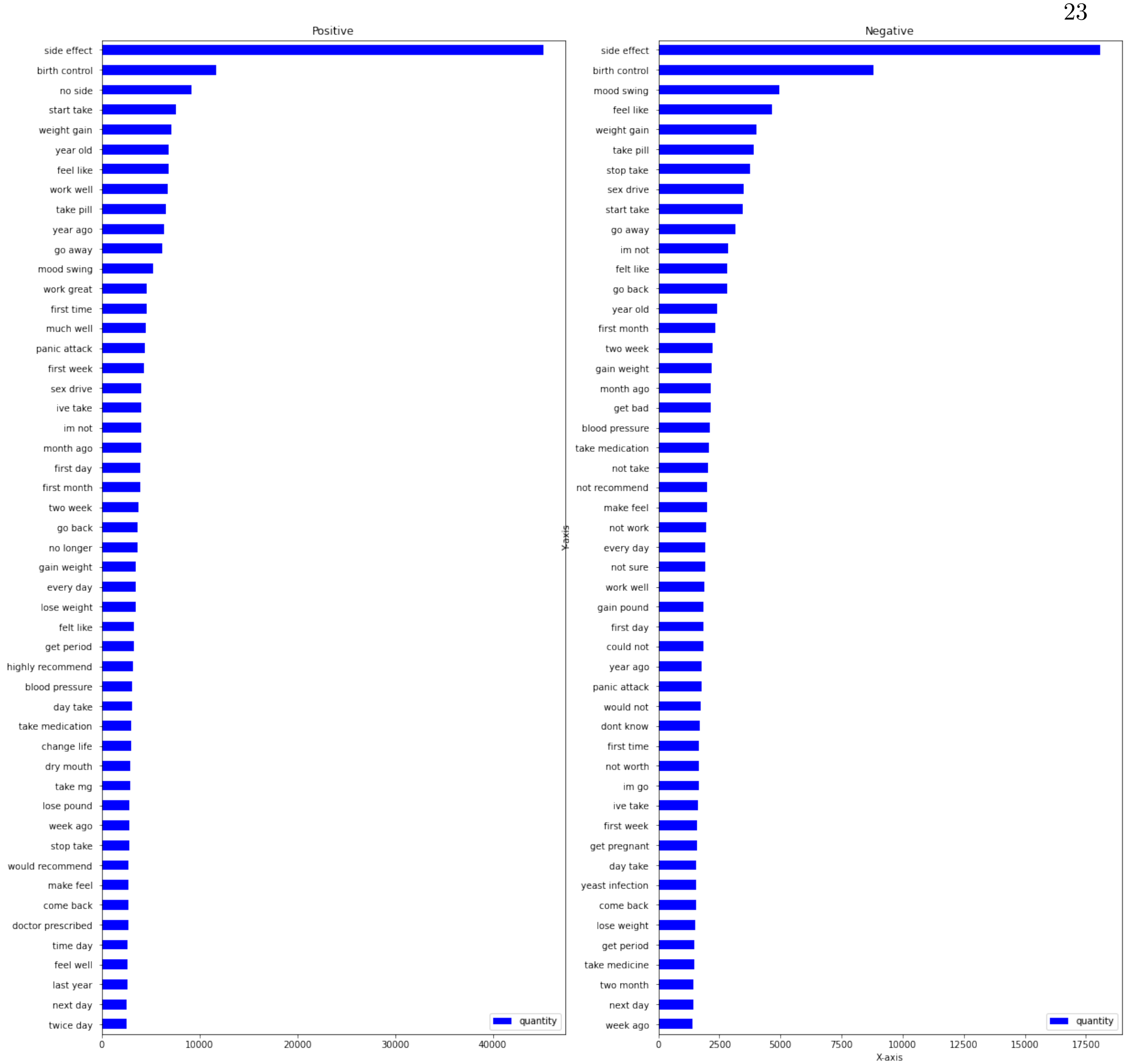
Now let's start on our models:

4.2.1 N-Gram Model

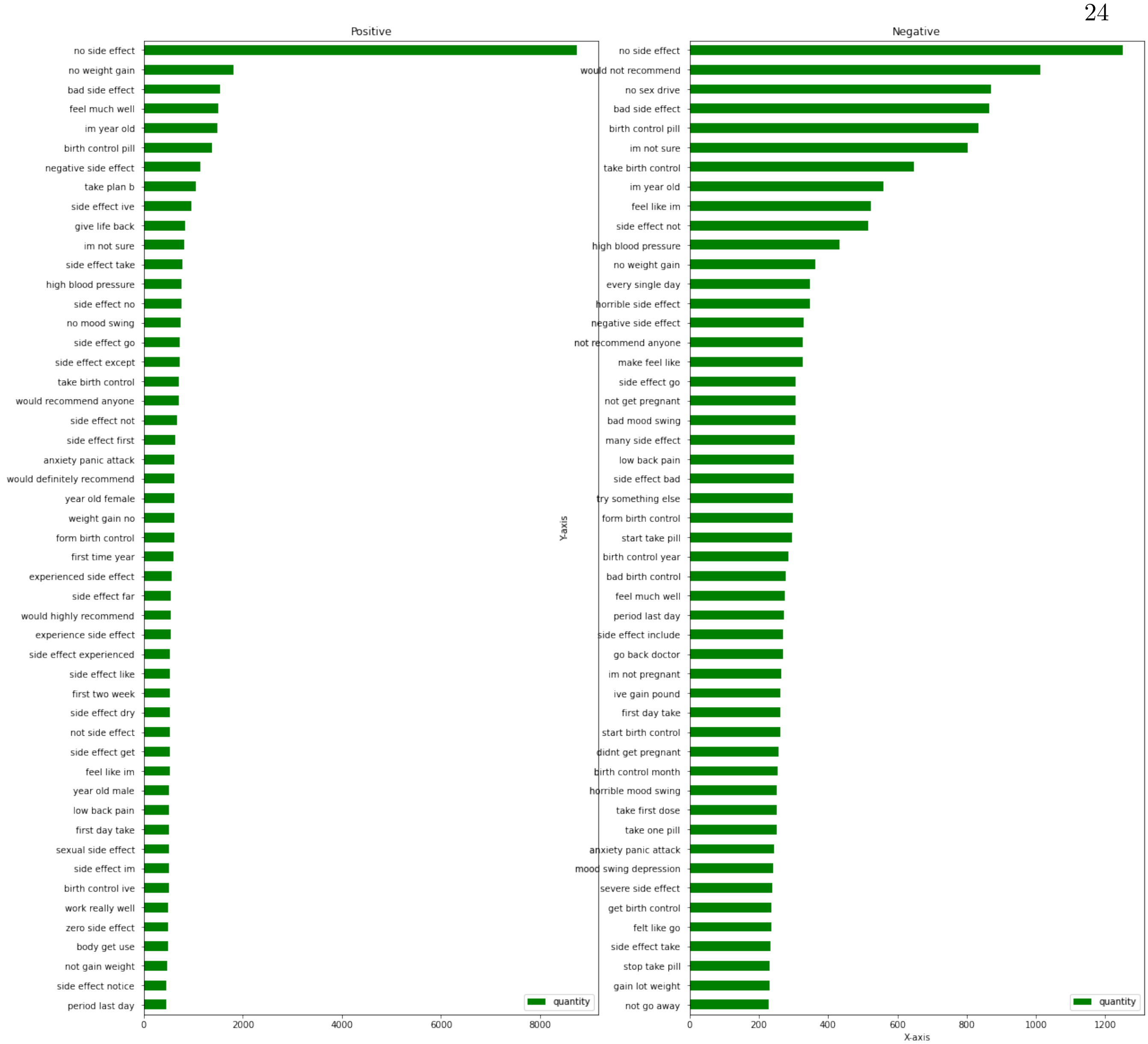
Our first model will be N-Gram Model, here for that we will be first needing to find the optimum value of "n" here:

Let's first see for n = 1:

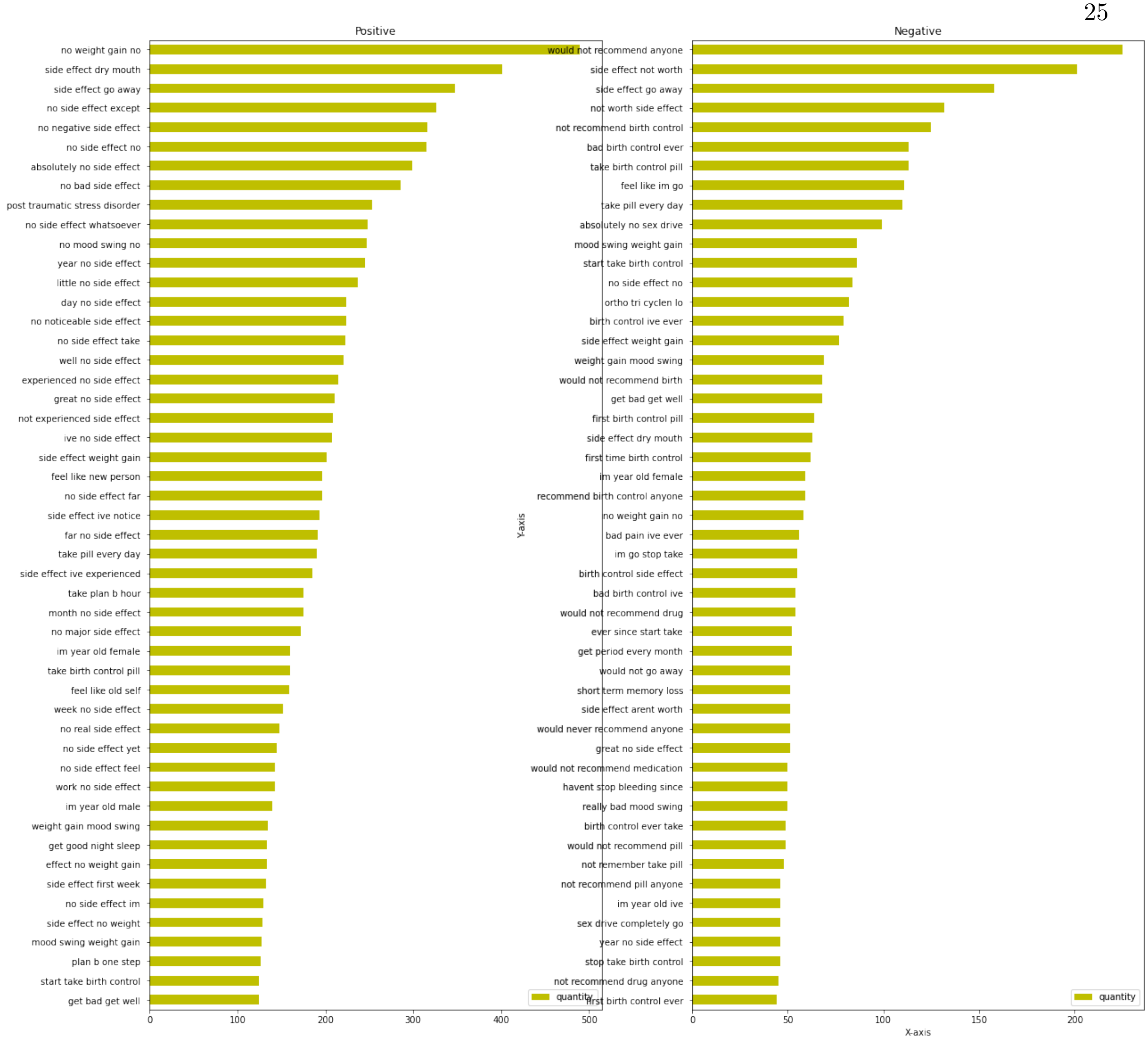
Here, in 1-gram as we can see in 1-GRam Model, here top words are exactly same in both just in different order, which tells us that this does not classify emotion well. Now let's see for 2-Gram:



Here, as we can see it is also not that good in classifying emotions, Let's move to 3-Gram:

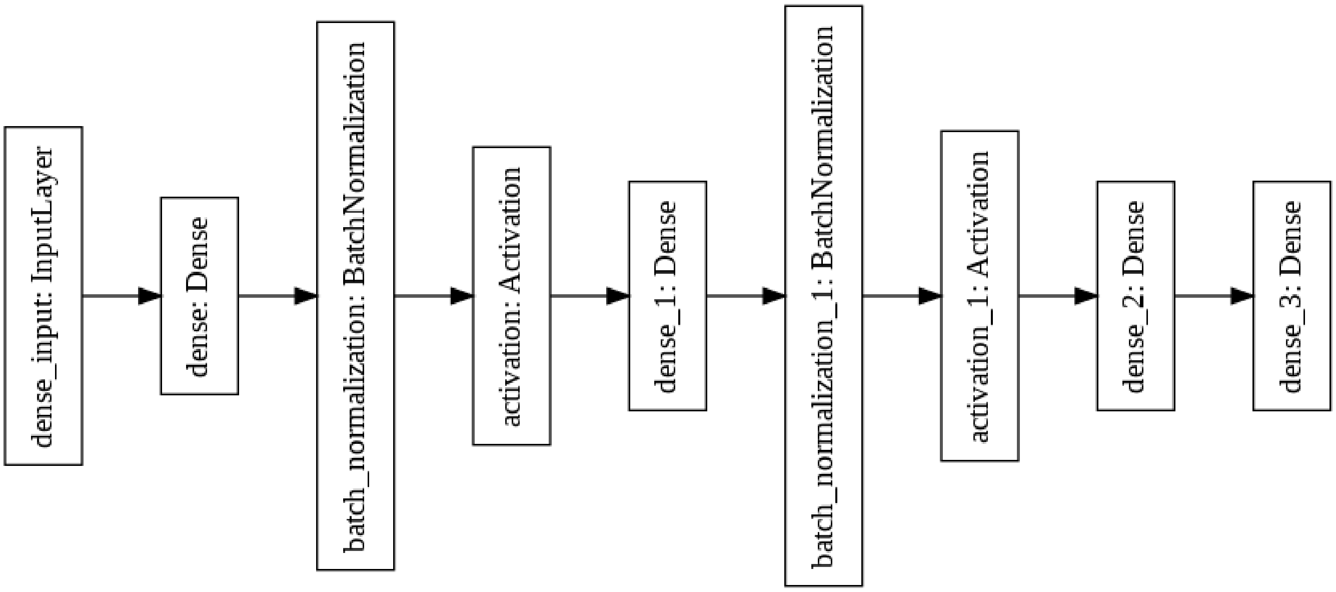


This is not that bad, but it is still far from perfect, let's move to 4-Gram:

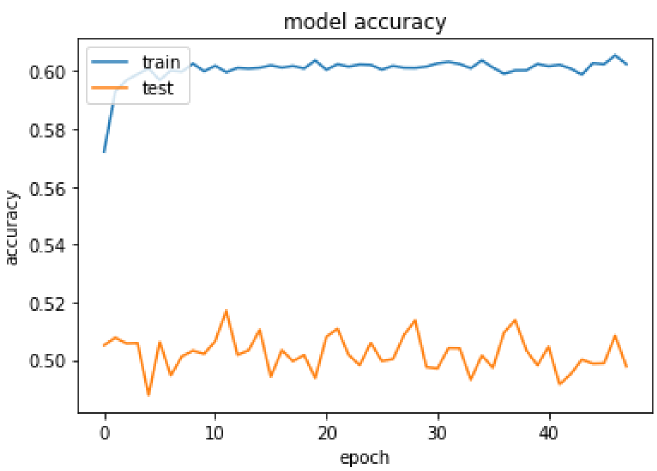


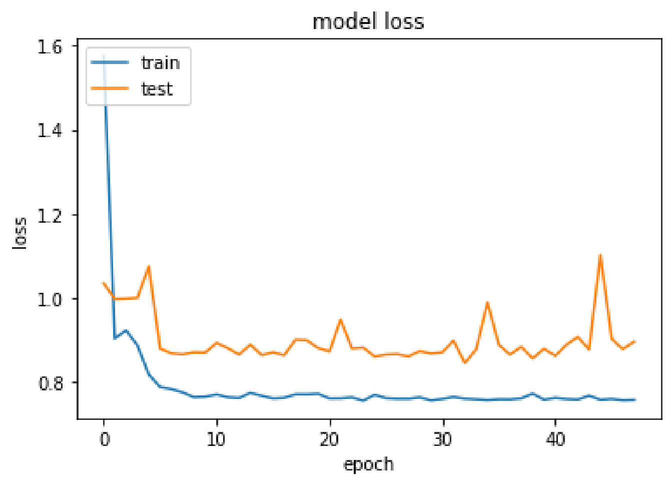
Here, this looks great, as we can see it classifies data very accurately, there we will be creating a 4-Gram model for this project.

Now for the inputs, we will be sending the bag of words of type of input to our model, which basically tells the frequency of each word from dictionary in that particular input/review. After lots of optimizations this was the model architecture I finalized:



After training we got the following results:

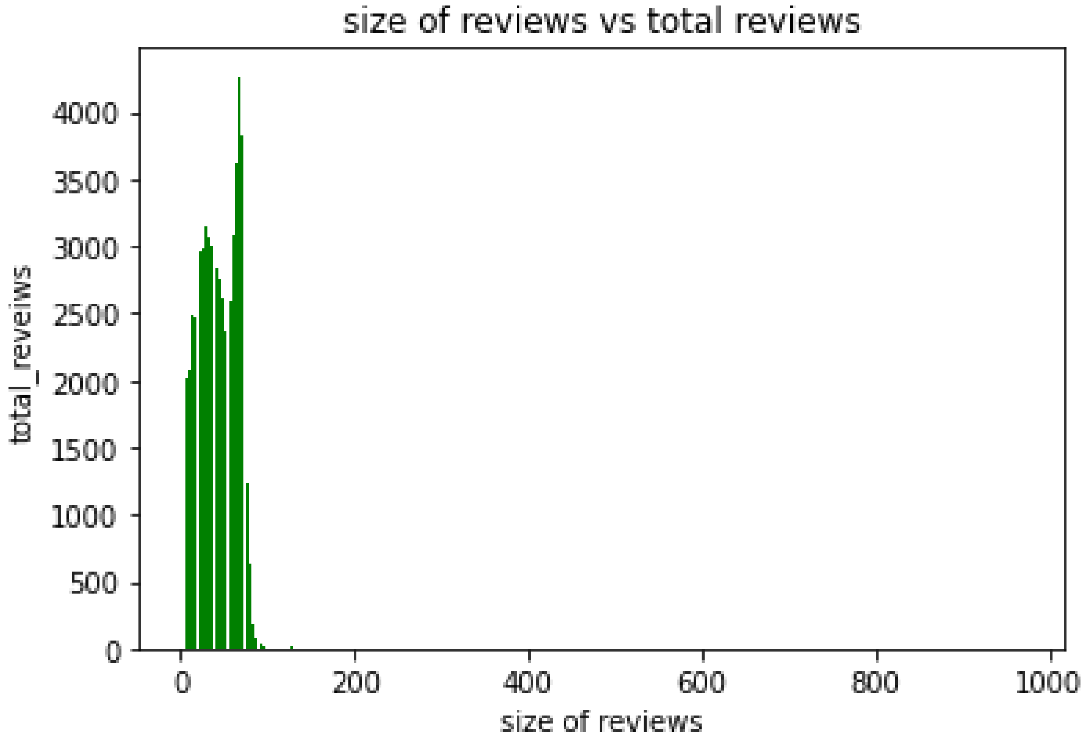





```
2366/2366 [==========================] - 28s 12ms/step - loss: 0.7660 - accuracy: 0.6004
262/262  [==========================] - 2s 8ms/step - loss: 0.8918 - accuracy: 0.5024
```

4.2.2 BiLSTM

Now, we will create the BiLSTM model, but before that we will be needing to set the size of input for our model, for this, first we will be checking the size of reviews, which is as follows:



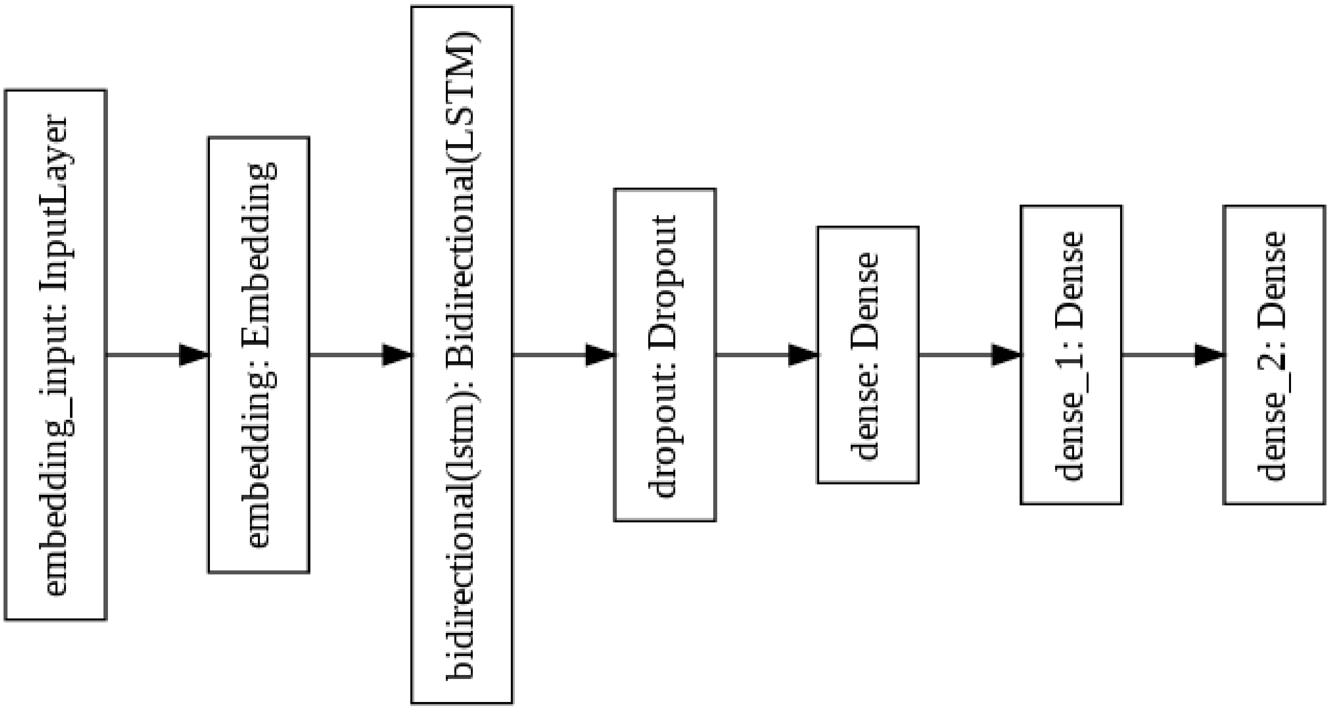
From here as we can see most of the reviews are of size under 100, I also checked the percentage of reviews with size under 100 words, it is as follows:-

```
Total Reviews under 100 words: 99.83664515278621%
```

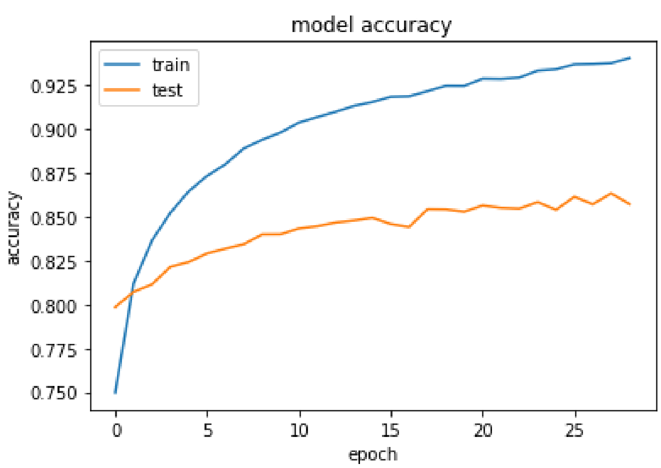
Now, next part will be tokenization and padding of reviews, for example our earlier review becomes like this now:

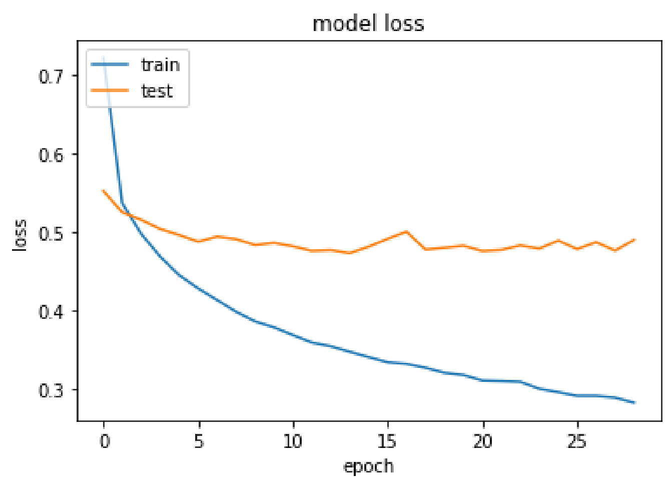
```
[   0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
    0,    0,    0,    0,    1,  275,  445,   23,   48, 1246,    1,
  957,  136,    4,   55,  317,   98,    5,   55,  204,  326,  175,
   86,  157,  120,   15,   49,  375,   10,    1,   23,    3,  114,
   49,    2,    1,  372,  301,  280,  241,  354,  668,   25,  198,
  154]
```

And after lots of optimizations, this is the model design which I found was working the best:



And after training following were the results obtained:



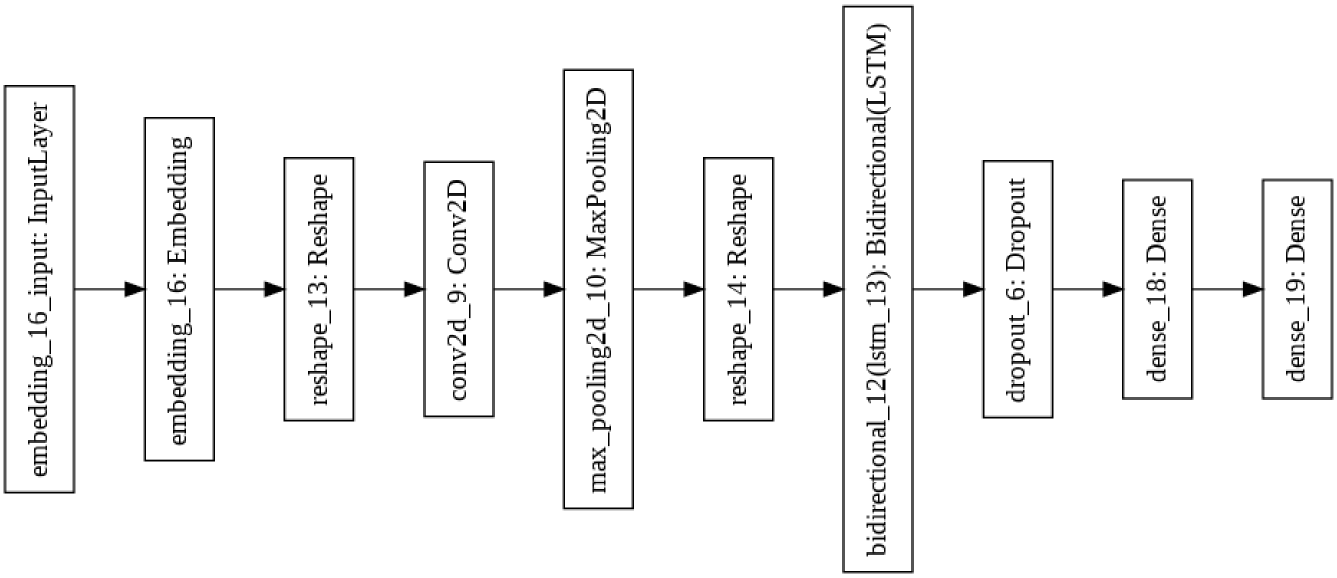


```
4734/4734 [==========================] - 41s 9ms/step - loss: 0.2453 - accuracy: 0.9507
526/526  [==========================] - 5s 9ms/step - loss: 0.4896 - accuracy: 0.8570
```

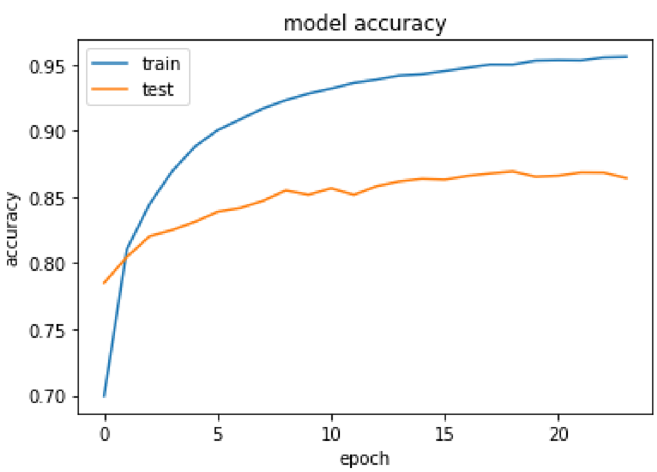
4.2.3 CNN-BiLSTM

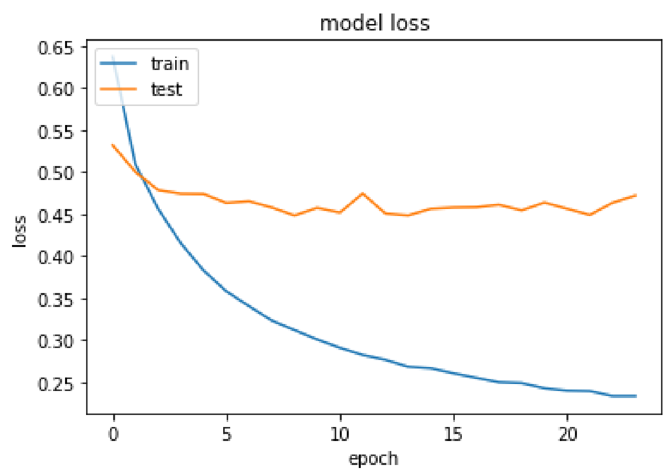
Now, we will be creating a CNN-BiLSTM model, for this model text preprocessing steps are all same as for the BiLSTM one.

Here is the model that I finalized for this method, after lots of optimizations:



And after training following were the results obtained:






```
4734/4734 [==========================] - 61s 13ms/step - loss: 0.1894 - accuracy: 0.9686
526/526  [==========================] - 7s 12ms/step - loss: 0.4720 - accuracy: 0.8642
```

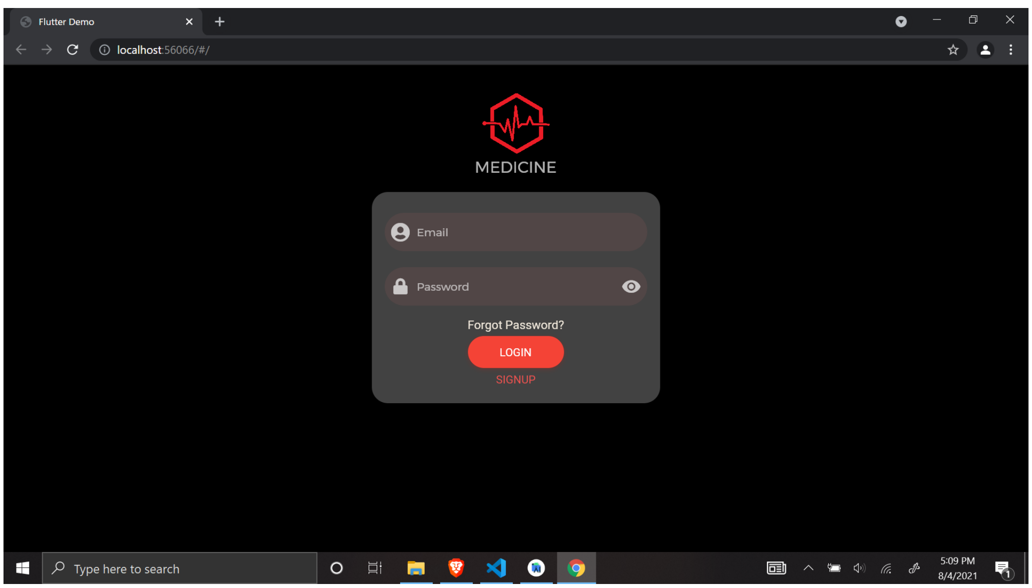
4.3 Conclusion

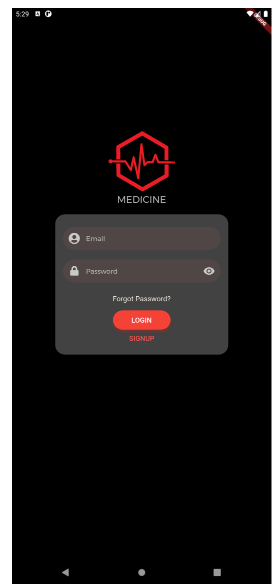
From Here as we can see CNN-BiLSTM works the best for the sentiment analysis. Therefore I have used CNN-BiLSTM model as my finalized model for sentiment analysis to provide a sentiment score and used it for generating rankings.

4.4 Mobile & Web App

After I have completed our ML Model and Ranking Generator, I created both Mobile and Web apps here are some screenshots for the same:

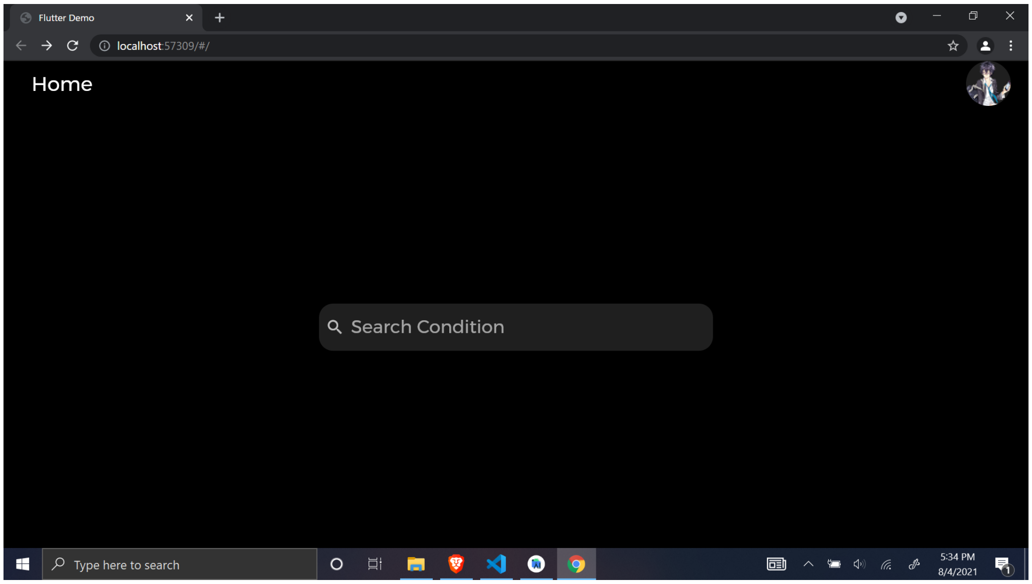
4.4.1 Login Screen

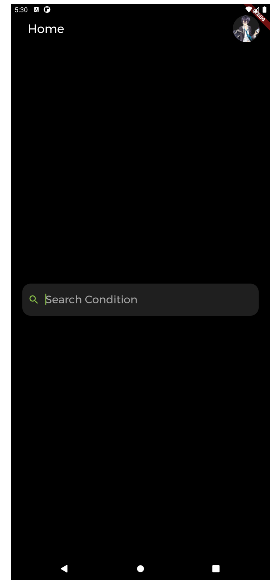




In this screen, user can login to their account and also sign-up for their new account. This screen has cool **animations**, while actually using the app.

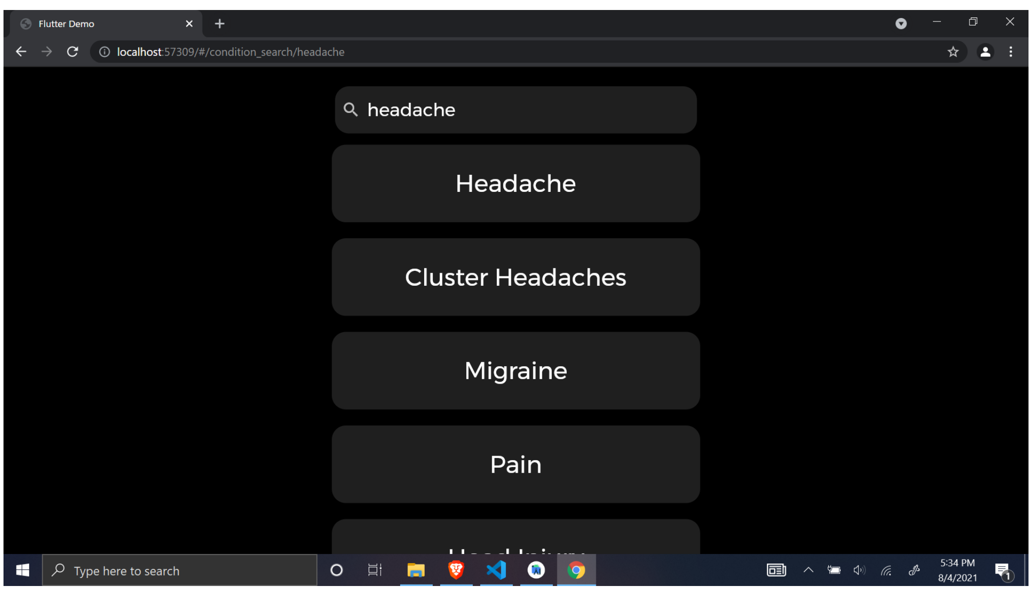
4.4.2 Home Screen

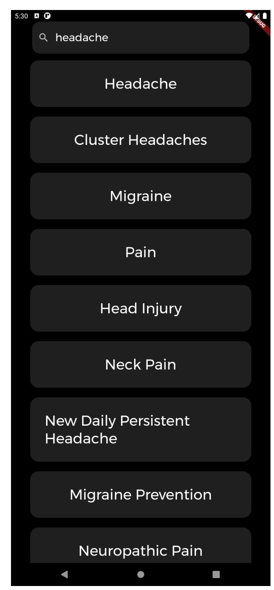




In this screen, user can see his profile-pic or auto-generated robot pic from gravatar, at the top-right corner of the screen. User can search the condition for which they want medicine recommendations.

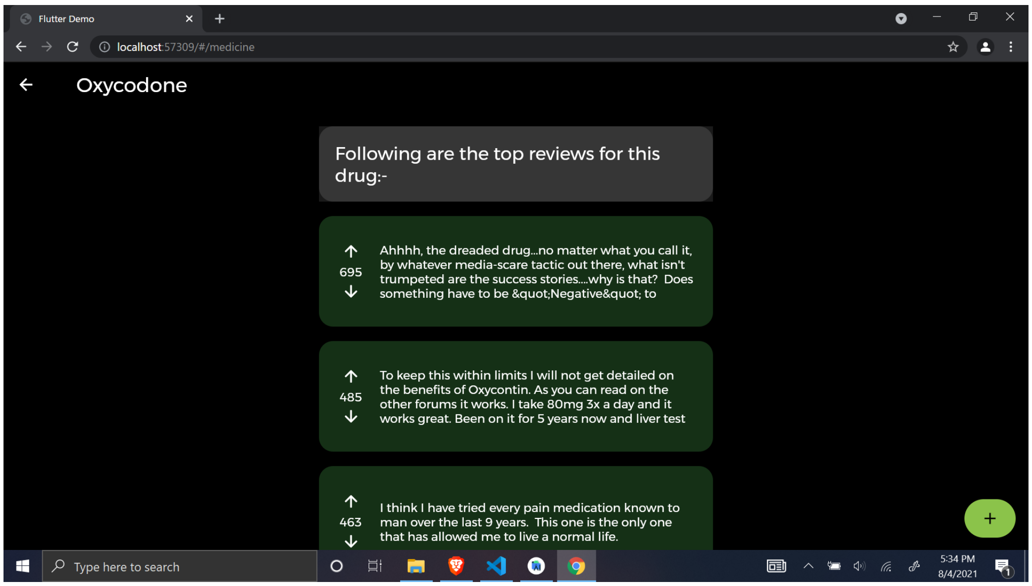
4.4.3 Search Results Screen

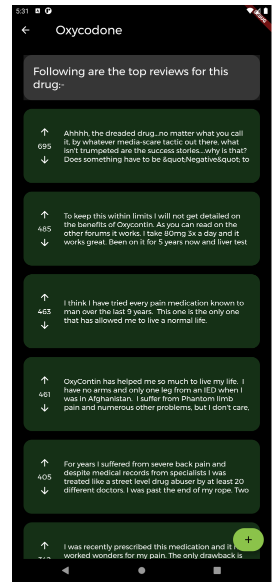




In this screen, user will see the search box from last screen to be animated from center to the top of the screen and the results will appear below it. First thing to notice on this page is that we have searched for headache and we are still getting results like "Pain", "Migraine", "Head Injury" this shows that the model knows the meaning behind the conditions and can provide a smart suggestion based on the semantic of the condition and the query.

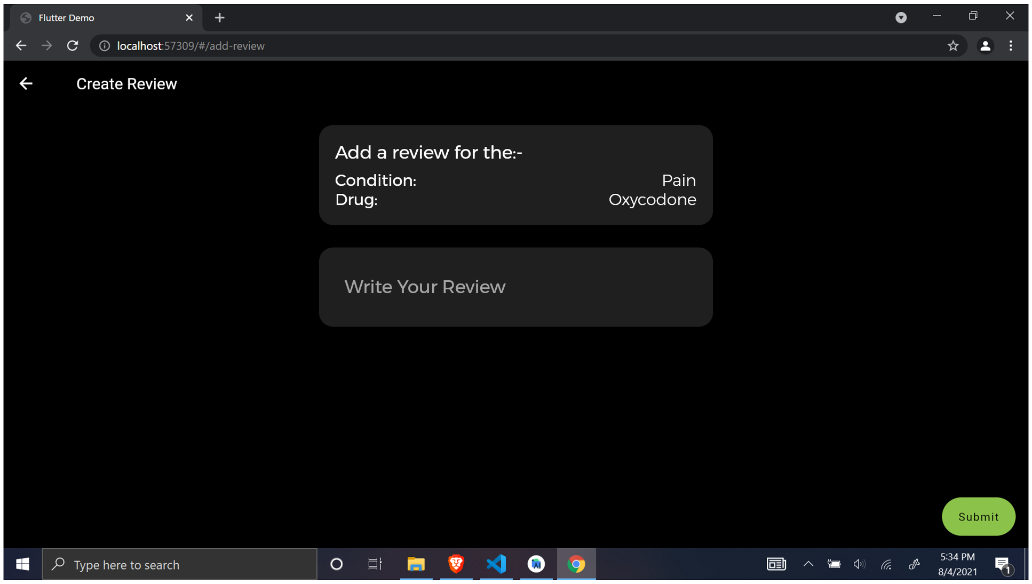
4.4.4 Reviews Screen

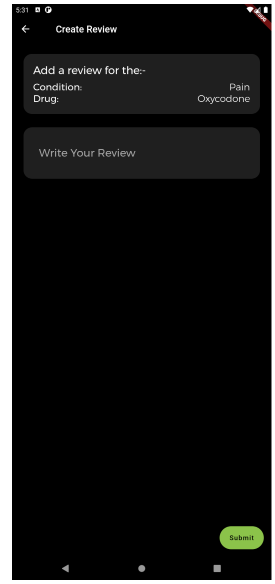




In this screen, we can see the top positive and top negative reviews for the medicine and condition selected Each positively marked review is greenish in color while all negatively marked reviews are marked red in color. User can upvote and downvote the reviews too, according to if he/she founds it helpful or wrong respectively. On clicking on the bottom down corner user will be redirected to the Add Review Screen to add his own review for this particular condition and medicine.

4.4.5 Add Review Screen





In this screen, User can submit his own review for the particular medicine and condition. Note:- New Reviews and Upvotes/Downvotes are not instantly added to the permanent database and considered by the ML model. ML model auto fetch new reviews and upvotes/downvotes for medicines and conditions, and re-generate ranking for the medicines for each condition.

Chapter 5

Discussions and conclusion

Getting good recommendation for medicines is very essential and will save many lives in future. Once the project is completed we will have a mobile application and a web application on which we could search for a condition and get the recommended medicines for it. And also we should be able to add our own review for any medicine and condition, which will further help to improve the model predictions.

5.1 Limitations

- Current model can only predict medicines for a single condition.
- In this system we are only using reviews and useful count for our final predictions, there can be many other factors included in the project for further enhancement.
- There are many different approaches which could have been used for sentiment analysis, in this project we have only used the most popular ones. Also it is very common to create our own hybrid model and new approach for a specific machine learning problem, while here we have only selected the best of the popular approaches.

5.2 Future scope

- In future we would like to add predictions based on multiple conditions too.

- Also we can add the doctor recommendation for a specific conditions patient might be feeling, and also add contact details for each doctor, also with instant chat option too.

- If more time is available more different approaches should also be tried for sentiment analysis model, and also try to create our own approach and model for this particular problem if possible.

- One could add more factors for the final predictions of rankings too, like adding a deep learning model which would take sentiment score, useful count, blood pressure, oxygen(Spo2) value etc. and compare two medicines for the condition.

Bibliography

[1] Rung-Ching Chen, Yun-Hou Huang, Cho-Tsan Bau, and Shyi-Ming Chen. A recommendation system based on domain ontology and swrl for anti-diabetic drugs selection. Expert Systems with Applications, 39(4):3995–4006, 2012.

[2] John D. Cook. A bayesion view of amazon resellers, 2011.

[3] M. Ghiassi, J. Skinner, and D. Zimbra. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. Expert Systems with Applications, 40(16):6266–6282, 2013.

[4] Harshith. Text preprocessing in natural language processing, 2019.

[5] Renu Khandelwal. Word embedding for natural language processing, 2019.

[6] Jessica Li. Uci ml drug review dataset.

[7] Evan Miller. Formulas for bayesion testing.

[8] A-nanya Promkot, Somjit Arch-int, and Ngamnij Arch-int. The personalized traditional medicine recommendation system using ontology and rule inference approach. In 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), pages 96–104. IEEE, 2019.

[9] Maryem Rhanoui, Mounia Mikram, Siham Yousfi, and Soukaina Barzali. A cnn-bilstm model for document-level sentiment analysis. Machine Learning and Knowledge Extraction, 1(3):832–847, 2019.

[10] Tensorflow. Universal sentence encoder.

[11] Guixian Xu, Yueting Meng, Xiaoyu Qiu, Ziheng Yu, and Xu Wu. Sentiment analysis of comment texts based on bilstm. Ieee Access, 7:51522–51532, 2019.