



A Blockchain and ML based Drug Supply Chain Management and Recommendation System for Smart Pharmaceutical Industry

By

Tushar Shandilya (18135116)

Ankit Sarangi (18135117)

Vijaykumar A. L. (18135119)

Anshul Shrivastava (18135014)

Sumit Kumar Pradhan (18135100)

B.tech in Mechanical Engineering

Under the supervision of

Dr. Cherian Samuel

Mechanical Engineering Department

IIT (BHU), Varanasi

CERTIFICATE

This is to certify that the project titled **A Blockchain and ML based Drug Supply Chain Management and Recommendation System for Smart Pharmaceutical Industry** is a bonafide record of the work done by

Tushar Shandilya (18135116)

Ankit Sarangi (18135117)

Vijaykumar A. L. (18135119)

Anshul Shrivastava (18135014)

Sumit Kumar Pradhan (18135100)

under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Mechanical Engineering** of the **Department of Mechanical engineering Indian Institute of Technology, BHU Varanasi 221005**, India, during the year 2021-2022.

Their work is genuine and has not been submitted for the award of any other degree to the best of my knowledge.

DATE : 16.12.2021

Dr. Cherian Samuel
Associate Professor
Department of
Mechanical Engineering,
IIT (BHU)

DECLARATION

This is to certify that the work which is being hereby presented by us in this project titled “**A Blockchain and ML based Drug Supply Chain Management and Recommendation System for Smart Pharmaceutical Industry**” in partial fulfillment of the award of the Bachelor of Technology submitted at the Department of Mechanical Engineering at Indian Institute of Technology, BHU ,Varanasi, is a genuine account of our work carried out during the period from July to November 2021 under the guidance of **Dr. Cherian Samuel**, Associate Professor, Department of Mechanical Engineering at Indian Institute of Technology , BHU ,Varanasi.

The matter embodied in the project report to the best of our knowledge has not been submitted for the award of any other degree elsewhere.

DATE : 16.12.2021

Tushar Shandilya (18135116)

Ankit Sarangi (18135117)

Vijay A. Lokhande (18135119)

Anshul Shrivastava (18135014)

Sumit Kumar Pradhan (18135100)

Table of contents

Certificate

Declaration

Table of contents

Introduction

Methodology

Dataset and Preprocessing

Disease Prediction

Sentiment Analysis of drug reviews

Drug Recommendation

Results

Sentiment Analysis

Drug Recommendation Analysis

Conclusion and Future scopes

References

Introduction

One of the most commonly found concerns among patients when confronted with any medical condition is which physician to trust. It is a known fact that the health of an individual significantly affects his/her quality of life. A survey in 2013 by the Pew Internet and American Life Project found that 59% of adults have looked online for health topics and with 35% of respondents focusing on diagnosing a medical condition online. There are more people every day caring about the health and medical diagnosis problem but still many who lose their lives due to medical errors. According to the administration's report, more than 200 thousand people in China and over 100 thousand in the USA, die each year due to medication errors. More than 42% medication errors are caused by doctors because they write prescriptions based on their experience which is quite limited. Hence, finding appropriate physicians to diagnose and treat medical conditions is one of the most important decisions a patient must make.

Advancement in Data mining and Recommender Technologies allow us to explore possibilities of potential knowledge from diagnosis history records and reviews and ratings on drugs to help doctors prescribe the correct medication and decrease the medication errors effectively.

The objective of this Data Mining paper is to design and implement a universal Disease Prediction and Drug Recommendation System that applies various Data Mining technologies to the recommendation system. By combining information from different sources we are using various prediction algorithms along with NLP for sentiment analysis and recommendation. Rest of the report talks about Data Gathering, Pre-processing, Methodology, Results and Conclusion of our project.

Methodology

The main goal of our project is to recommend a drug to a patient based on the symptoms she/he has. In accordance with our objective to implement a drug recommender system there are two main subcategories which are to be addressed i.e a disease prediction model and a recommendation model and below is the design pipeline and dataflow of our implementation.

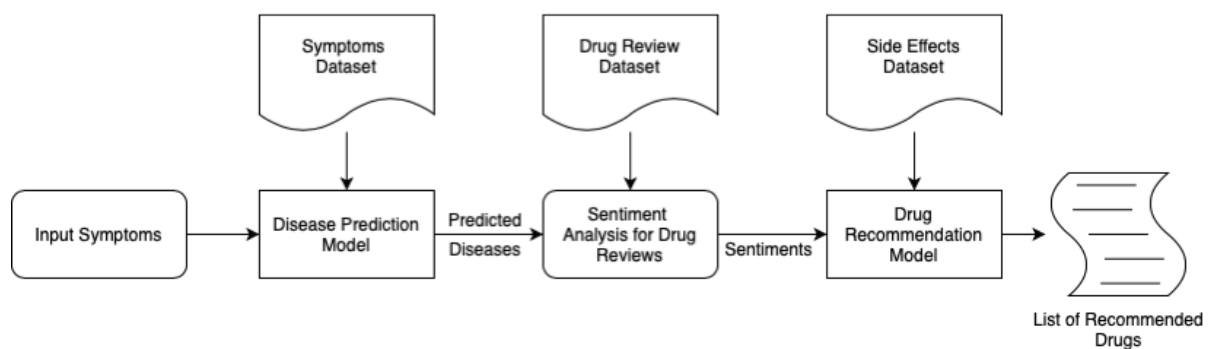


Figure 2: Design implementation pipeline and dataflow of our recommender system. Disease prediction model is a classification model with takes in the symptoms and gives the predicted diseases and the sentiment analysis on the drug reviews and mapped to predicted disease and finally recommended drugs based on the sentiments of the reviews and rank along with possible side effects.

Dataset and Preprocessing

Healthcare information is protected by HIPPA. Sharing of medical records of patients without their knowledge is prohibited. Getting access to government health records and datasets required multiple permissions. Hence, for our project, we are using the datasets that were readily available on the internet and were open to downloads.

Data Gathering

For the accurate recommendation of drugs, we first predict the diseases based on symptoms and then recommend the drugs based on the ratings. For this purpose, we have tried to gather information from two main datasets.

1. Symptoms dataset:

- This dataset is used to take symptoms as input and predict the disease as an output.
- Dataset is obtained from the [Disease-Symptom Knowledge Database](#) which is a knowledge database of disease-symptom associations generated by an automated method based on information in textual discharge summaries of patients at New York-Presbyterian Hospital admitted during 2004.
- This dataset contains 3 columns:
 - Disease
 - Count of Disease Occurrence
 - Symptom

	Disease	Count of Disease Occurrence	Symptom
0	UMLS:C0020538_hypertensive disease	3363.0	UMLS:C0008031_pain chest
1	NaN	NaN	UMLS:C0392680_shortness of breath
2	NaN	NaN	UMLS:C0012833_dizziness
3	NaN	NaN	UMLS:C0004093_asthenia
4	NaN	NaN	UMLS:C0085639_fall

Raw Symptoms dataset

- There are a total of 149 unique diseases in this dataset and 405 symptoms.
- Each disease contains 4-5 symptoms corresponding to it.
- This dataset is sent for pre-processing so that it can be used to train models to classify and predict the disease.

2. Drug Review Dataset:

- This dataset is used in order to take the predicted disease as input and recommend appropriate drugs based on reviews and ratings (Sentiment Analysis).
- The dataset is gathered from the [UCI Machine Learning Repository for Drug Review](#) which provides patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting overall patient satisfaction.
- The Repository had two datasets (Test and Train) which are combined for analysis and visualisation purposes as they had the same number of columns.
- It contains 7 columns and 215063 rows:
ID
 1. Drug name
 2. Condition
 3. Review
 4. Rating
 5. Date
 6. Useful count

	uniqueID	drugName	condition	review	rating	date	usefulCount
0	206461	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9.0	20-May-12	27
1	95260	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8.0	27-Apr-10	192
2	92703	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5.0	14-Dec-09	17
3	138000	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8.0	3-Nov-15	10
4	35696	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9.0	27-Nov-16	37

Raw Drug Review Dataset

- There are 3671 unique Drug names and 916 unique Conditions (Disease) in this dataset along with the rating and reviews corresponding with the drug names.

- This dataset is then pre-processed and visualised to gain more information for effective drug recommendation.

3. Side Effect Dataset:

- We have successfully included this dataset containing side effects of specific drugs in order to help patients identify the risks involved in the drug that is being recommended.
- This dataset is again gathered from UCI Machine Learning Repository for Side Effects of Drugs along with some raw data gathered from druglib.com.
- This Dataset contains a lot of columns similar to the drug review dataset, but it does not have a lot of rows. Hence only the “Side Effects” column from this dataset will be combined with the other two datasets along with the side effects found from druglib.com

urlDrugName	rating	effectiveness	sideEffects	condition	benefitsReview	sideEffectsReview
enalapril	4	Highly Effective	Mild Side Effects	management of congestive heart failure	slowed the progression of left ventricular dys...	cough, hypotension , proteinuria, impotence , ...
ortho-tri-cyclen	1	Highly Effective	Severe Side Effects	birth prevention	Although this type of birth control has more c...	Heavy Cycle, Cramps, Hot Flashes, Fatigue, Lon...

Raw Dataset containing Side effects of drugs

Data Preprocessing

In this project, we have used and worked upon multiple datasets. All the datasets were obtained in raw format. To preprocess all the datasets, a few common steps and measures were carried out. Those were

- All the datasets were first checked for the count of null and missing values.
- All such values were either handled or dropped from the dataset.
- After that, every column’s unique values were found and their frequencies.
- Using standard libraries, the dataset was visualised and outliers, if any, were found out.
- Any irrelevant information was deleted from the dataset.

1. Symptoms Dataset

- This dataset had to be cleaned in order to gain information from it.
- First, the “Count” column was dropped as the information given in it is irrelevant for this project.

```
data = df.fillna(method='ffill')
```

- Then, the null values in the Disease column were handled using the drop function.
- Cleaning of Disease and Symptom columns was done to get only the name and removed unnecessary data.

	Disease	Symptom
0	hypertensive disease	[pain chest, shortness of breath, dizziness, a...
1	diabetes	[polyuria, polydypsia, shortness of breath, pa...
2	depression mental	[feeling suicidal, suicidal, hallucinations au...
3	depressive disorder	[feeling suicidal, suicidal, hallucinations au...
4	coronary arteriosclerosis	[pain chest, angina pectoris, shortness of bre...

Symptoms dataset after preprocessing

- For classification of the symptoms based on diseases, we have converted it into a new CSV format file which now has symptoms as the columns and diseases as rows. Using one hot encoding, we have mapped every symptom with all the diseases and adding value 1 if it is present for disease and 0 otherwise. Below is a screenshot of the one-hot encoded dataset. This will help us predict the diseases when symptoms are given as input.

Dataset after marking symptoms present for a disease as 1 else 0

	Disease	Heberden's node	Murphy's sign	Stahl's line	abdomen acute	abdominal bloating	abdominal tenderness	abnormal sensation	abnormal hard consistence
0	Alzheimer's disease	0	0	0	0	0	0	0	0
1	HIV	0	0	0	0	0	0	0	0
2	Pneumocystis carinii pneumonia	0	0	0	0	0	0	0	0
3	accident cerebrovascular	0	0	0	0	0	0	0	0
4	acquired immuno-deficiency syndrome	0	0	0	0	0	0	0	0

5 rows x 405 columns

2. Drug Review Dataset

- This dataset contained two sets (Train and Test) which were combined to be able to visualise and analyse the data on a larger dataset. And also because they both had the same columns so could be combined easily.
- The dataset obtained was pretty clean and did not require a lot of pre-processing. Still, a few rows with null values were dropped and columns were renamed.
- The dataset contains a lot of information and visualising was an interesting task.
- Many different graphs were plotted showing results of drugs with most reviews, most popular drugs, most common diseases, etc.

3. Side Effects Dataset

- This dataset has a lot of information similar to the one in the Drug Review dataset and it was also clean.
- A few null values are handled and irrelevant columns are dropped.
- This dataset is merged with the drug review dataset to map only the side-effects for the specific drugs.

4. Merged Dataset

- The Dataset containing symptoms is merged with the one with reviews for final drug prediction.

<pre>data_drugs['Disease'] = data_drugs['Disease'].str.lower() data_symptoms['Disease'] = data_symptoms['Disease'].str.lower() merged_data = pd.merge(data_drugs, data_symptoms, on='Disease') merged_data.head()</pre>						
	drugName	Disease	review	rating	usefulCount	Symptom
0	Aspirin	transient ischemic attack	"No side effects, easy to take, no more sympt...	10.0	10	['speech slurred', 'dysarthria', 'facial pares...
1	Clopidogrel	transient ischemic attack	"I've been taking this medicine for a lit...	10.0	8	['speech slurred', 'dysarthria', 'facial pares...
2	Clopidogrel	transient ischemic attack	"I took ibuprofen (2 caps at night for severe ...	6.0	13	['speech slurred', 'dysarthria', 'facial pares...
3	Clopidogrel	transient ischemic attack	"After my VAD Stroke I am on plavix. I have a...	5.0	9	['speech slurred', 'dysarthria', 'facial pares...
4	Bayer Children's Aspirin	transient ischemic attack	"No side effects, easy to take, no more sympt...	10.0	10	['speech slurred', 'dysarthria', 'facial pares...

Merged Dataset

- Only the relevant columns are kept in the merged dataset and rest are dropped to reduce the dimensionality.

Disease Prediction

The disease prediction model is a probabilistic model which will give the predictions based on the symptoms. For this purpose, we are using the [Disease-Symptom Knowledge Database](#) which has over 149 unique diseases mapped with 405 symptoms.

For accurately predicting the diseases, we have experimented with multiple approaches in our project. This is because the dataset has only one feature that is the symptoms and hence training any classifier with one data point makes it weak which in turn hampers the prediction rate for different inputs. The solution is to try and create multiple data points for the same disease by taking into account all the symptoms present for that disease, their importance and their occurrence. Since we cannot say for sure which method will be well suited for this kind of dataset, we have implemented 2 different approaches and both of them focus on a different way of prediction.

Approach 1

In this method, the data has been preprocessed, transformed and mapped to a data frame which consists of columns derived from the list of unique symptoms and the list of unique target diseases as labels. The preprocessed symptoms dataset was used to map the newly created disease prediction data frame by setting the values of symptoms present for a particular disease as 1 and 0 if not. And the rows were grouped by to obtain a single row for each unique disease with the corresponding symptoms column mapped to 1. The data was then trained using Decision Tree Classifier Model, Support Vector Machine Classifier Model, using different Symptoms as the training features and Diseases as the labels. And the trained model is used to predict the diseases by passing one, two, and three features to the classifier models.

Approach 2

Even after augmenting data based on the count of disease occurrence, the classifier produced the same accuracy rates of 86%. With 86% accuracy rate, the disease prediction using method 2 was not accurate enough. Hence we tweaked the data in such a way that the classifiers are able to draw patterns and learn accordingly. In this method, the order of importance of each symptom for a disease is preserved. The symptoms dataset was used to map the newly created disease prediction data frame by setting the values of symptoms present for a particular disease as 1 and 0 if not. The number of rows for each disease is multiplied to increase the number of rows. This will create a data frame with an increased and equal number of rows for

each disease. The mapping of 0's and 1's are done in the probabilistic manner. The symptom with the highest importance is mapped to 1 for all the rows of that particular disease. The second most important symptom is mapped to 95% of all the rows and so on. The least important symptom will approximately be mapped to 40% of the total rows of a particular disease. The dataset was then split into training and testing data. The data was then trained using Decision Tree Classifier Model, Support Vector Machine Classifier Model, using different Symptoms as the training features and Diseases as the labels. The trained model is used to predict the diseases by passing one, two, three, and four features to the classifier models.

```
[ ] ext.predict(arr)
array(['hypertensive disease'], dtype=object)

[ ] mnb.predict(arr)
array(['hypertensive disease'], dtype='<U36')

[ ] svm.predict(arr)
array(['hypertensive disease'], dtype=object)
```

Accurate drug predictions for four input symptoms

Sentiment Analysis of drug reviews

Upon obtaining the most probable diseases the next task is to map the list of drugs that can be prescribed for this particular disease using the Merged Dataset which has been created by preprocessing the symptoms and the drug review dataset. Once we obtain the list of possible drugs the next task is to be able to recommend the best drug for the patient. In accordance we adopted a NLP based approach to analyse the sentiments using a neural net model in order to obtain positive or negative sentiment predictions for the reviews.

Approach

In this approach, we will be adopting sentiment analysis using the Natural Language Processing on the drug review data set to understand the trend in the positive and negative reviews given by the patients. The main rationale behind doing this sentiment analysis was the rating given and the review stated was seen to be inconsistent so likely if the review stated it was good and no side effects it still had a rating below 5 which made it hard to just make use of the rating based on some threshold and as NLP captures the essence of a sentence to predict a sentiment hence it proves to be more reliable. The NLP model was designed based on the word2vec method and a neural network classifier was built to classify the reviews. The model was designed by extracting the probabilistic values of relational occurrence using TfidfVectorizer that convertes the words in the sentences into a vector of probabilities. Upon obtaining the vectors a neural network model using sequential method was trained on this data to obtain the sentiment predictions. The model totally consisted of 3+1 layers with each layer having batch normalization and relu activation and the last fully connected layer has softmax activation function and binary cross entropy loss was used with adam optimizer. The neural network model implementation was completely done using keras API.

Algorithm: Sentiment Analysis; epochs: 25, l_rate:1e-1, batch_size = 100

Input

X: Reviews

Y: Ratings

Output

Drugs, Reviews, Disease and Sentiments steps:

1. encode the rating with threshold of 7 into two classes (>7 and <7) - input: y

2. get Tfidf vectors for the vectors - input: X
3. define sequential model using keras API
4. for epochs in steps per epoch.
 - train(fit) the model
 - update gradients
5. get predictions for all the reviews and merge the data to original dataset

Model: "sequential"		
Layer (type)	Output Shape	Param #
=====		
dense (Dense)	(None, 25)	315550
batch_normalization (BatchNo	(None, 25)	100
activation (Activation)	(None, 25)	0
dropout (Dropout)	(None, 25)	0
dense_1 (Dense)	(None, 25)	650
batch_normalization_1 (Batch	(None, 25)	100
activation_1 (Activation)	(None, 25)	0
dropout_1 (Dropout)	(None, 25)	0
dense_2 (Dense)	(None, 25)	650
batch_normalization_2 (Batch	(None, 25)	100
activation_2 (Activation)	(None, 25)	0
dense_3 (Dense)	(None, 2)	52
=====		
Total params: 317,202		
Trainable params: 317,052		
Non-trainable params: 150		

Neural Network model summary

Drug Recommendation

After the disease has been predicted, we need to recommend a drug for that disease. For recommendation of the drug, we have used the [UCI Machine Learning Repository for Drug Review](#) dataset which has the disease along with its multiple available drugs, their reviews, ratings and useful count.

This dataset, after preprocessing, is merged with the [Disease-Symptom Knowledge Database](#) on the basis of common diseases as shown in the section for preprocessing of merged dataset.

There are 6 features in this merged dataset out of which 3 are the ones that will help us to recommend the best drug. These 3 features and their importance is explained below:

- **Feature 1 : Review**

The review column is one of the most important columns for drug recommendation as it is a direct feedback from the users after using that particular drug. Hence, by doing Sentiment Analysis, we have taken into consideration only the reviews with a positive sentiment.

- **Feature 2 & 3 : Rating and Useful Count**

Every drug has a rating associated with it from 1-10 and a useful count which tells us how many users have given that particular rating. This information can be used to find a weighted average rating or a probabilistic value for each drug. Both the approaches are explained below:

Approach : Weighted Average Approach

Weighted average or mean is similar to an ordinary arithmetic mean, except that instead of each of the data points contributing equally to the final average, some data points contribute more than others.

Formula for weighted average:

$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i},$$

which expands to:

$$\bar{x} = \frac{w_1 x_1 + w_2 x_2 + \dots + w_n x_n}{w_1 + w_2 + \dots + w_n}.$$

Formula for weighted average

where, data elements with a high weight contribute more to the weighted mean than do elements with a low weight.

In our case, x is the rating and w is the useful count. In this way, we get a weighted average rating which gives more importance to the useful count i.e. the number of people who have actually given that rating.

This is the function created for calculating weighted average:

```
def wavg(group, avg_name, weight_name):
    d = group[avg_name]
    w = group[weight_name]
    try:
        return (d * w).sum() / w.sum()
    except ZeroDivisionError:
        return d.mean()
```

```
data_wavg = data.groupby(["Drug"], as_index=False).apply(wavg, "Rating", "UsefulCount")
```

After calculating , this weighted average column is merged with the dataset which replaces the rating column as shown:

	Disease	Drug	Review	Review_Sentiment	Rating_Wavg	UsefulCount
0	transient ischemic attack	Aspirin	"No side effects, easy to take, no more sympt...	Positive	10.000000	10
1	transient ischemic attack	Bayer Children's Aspirin	"No side effects, easy to take, no more sympt...	Positive	10.000000	10
2	transient ischemic attack	Clopidogrel	"I've been taking this medicine for a lit...	Negative	6.766667	8
3	transient ischemic attack	Clopidogrel	"I took ibuprofen (2 caps at night for severe ...	Negative	6.766667	13
4	transient ischemic attack	Clopidogrel	"After my VAD Stroke I am on plavix. I have a...	Positive	6.766667	9

After weighted average calculation

- This merged dataset now has the disease and drug name along with the review and its sentiment and the average rating with useful count.
- For recommending drugs to the user, we have only considered drugs which have a positive sentiment in their reviews. Hence, filtering out unwanted drugs.
- Also, we have taken a total of all the useful counts present for a particular drug to sort the drug that has to be recommended with the highest useful count first and then the highest average rating.
- For this, we have used the `groupby.sum()` method as shown below:

```
# taking predicted disease as input and recommending drug based on highest weighted average of ratings
groupedByCount = merged_wavg.groupby(['Disease', 'Drug', 'Rating_Wavg'])['UsefulCount'].sum().reset_index()
```

- This approach allows us to use all the information present in the dataset according to its importance and hence as an output we get the best possible drug recommendation.

Results

Approach 1

For disease prediction using method 1, we have used two models namely, Decision Tree Classifier and Support Vector Machine. Decision Tree Classifier produces 89.93% accuracy and SVM produces 87.58% accuracy. Disease predictions are wrong as there was insufficient data for the model to learn.

Approach 2

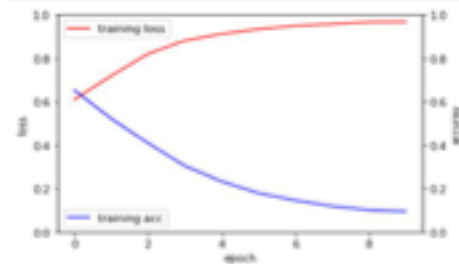
For disease prediction using method 2, we have used two models namely, Decision Tree Classifier and Support Vector Machine. Decision tree classifier provides 86.95% accuracy for a max depth of 120. And Support Vector Machine classifier provides 86.96%. The disease predictions were incorrect for one and two symptoms.

Models	Approach 1	Approach 2
Decision Tree Classifier	89.93%	86.95%
Support Vector Machine	89.93%	86.99%

Accuracy values of each approach.

Sentiment Analysis

For evaluating the NLP model we made use of loss, accuracy, f1 score, recall and



Evaluating model performance with scoring metrics

```
In [9]: scores = model.evaluate(x_test, y_test, verbose=0)
y_pred = model.predict(x_test)
y_pred = np.argmax(y_pred, axis=1)
y_true = np.argmax(y_test, axis=1)
score = precision_recall_fscore_support(y_true, y_pred, average="macro")
print("Evaluation metrics for Sentiment Analysis: ", "\n", "Loss: ", scores[0], "\n",
      "Accuracy: ", scores[1], "\n", "F1_Score: ", score[2], "\n", "Precision: ", score[0], "\n",
      "Recall: ", score[1])

Evaluation metrics for Sentiment Analysis:
loss: 0.8925471483015773
Accuracy: 0.670437715
F1_Score: 0.6676674342456834
Precision: 0.683251574187989
Recall: 0.6746314581395809
```

precision metrics and the following results were obtained.

Evaluation metrics for sentiment analysis

It can be noted that the training accuracy reached over 90% but not CV accuracy

some of the reviews and sentiments using d2v embedding

	reviews	actual label	\
0	"I am taking Diamox this is the 3rd time I hav...	positive	
1	"Works good for me."	negative	
2	"I've had glaucoma since I was 9, and I've...	positive	
3	"I have Iritis, so to get rid of the inflamati...	negative	
4	"It helped lower my pressure but made my eyes ...	negative	

	predicted label
0	negative
1	positive
2	positive
3	negative
4	negative

one reason could be due to inaccurate labelling of actual labels. Below are the few predicted sentiments.

Predicted labels of reviews

In the above figure it can be observed that in review at index 1 says "Works good for me" has a negative review but the predicted label classifies it accurately as a positive review hence giving it a strong reason to use NLP based predicted values than setting threshold for ratings.

Drug Recommendation Analysis

As discussed above, the final recommendation was done using the Weighted Average approach.

Few things considered while recommending the drug:

There are multiple drugs for a single disease. Hence, by using sentiment analysis we were able to filter out the negative and neutral reviews leaving us with only positive ones.

Approach : Weighted average Approach

The results obtained from this approach gave us the correct and highly rated drug for the predicted disease. The disease that we predicted is osteoporosis based on the given and we are recommending top 3 drugs from our dataset as an output. It can be seen below:

	Disease	Drug	Rating_Wavg	UsefulCount
246	osteoporosis	Calcium / vitamin d	9.0	15
248	osteoporosis	Caltrate 600+D	9.0	15
245	osteoporosis	Calci-Chew	8.0	5

```
print("Recommended drugs for this disease are:\n ", predicted_drug["Drug"].unique())  
Recommended drugs for this disease are:  
['Calcium / vitamin d' 'Caltrate 600+D' 'Calci-Chew']
```

Output of weighted average approach for drug recommendation.

Conclusion and Future scopes

Drug Recommendation systems are a prevailing technology in today's online services and with the rise of requirements of these services there is more and more need to automate the processes. Below are the key conclusions from our project.

- Successfully built a drug recommendation system that predicts diseases and recommends drugs along with possible side effects based on user symptoms input.
- We designed three models for this project implementation. A disease prediction model, Sentiment Analysis model and a recommendation model.
- Experimented with different approaches for each of three models.
- Each of the three models gave good accuracies contributing to the overall reliability of the drug recommendation model.

One key future scope can definitely be improving the accuracies of the prediction and recommender model using deep neural networks by using larger data and demonstration of the project using the ChaBot as a UI between the code that we've written and its results for better understanding. . Also with regard to chatbot UI it can be integrated to websites that cater towards online medical services.

References

- [1] Druglib.com - Drug Information, Research, Clinical Trials, News.
<http://www.druglib.com/>
- [2] <https://arxiv.org/pdf/2104.01113.pdf>
- [3] <https://www.ijstr.org/final-print/feb2020/Medicine-Recommendation-System-Based-On-Patient-Reviews.pdf>
- [4] Youjun Bao ; Xiaohong Jiang. (2016). An intelligent medicine recommender system framework

