2nd International Conference on Frontiers in Engineering Science & Technology
in association with International Journal of Scientific Research in Science,
Engineering and Technology | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
[https://ijsrset.com | doi : https://doi.org/10.32628/IJSRSET]

Drug Recommendation System based on Sentiment Analysis of Drug Reviews using Machine Learning

Aysha Maheen, Shafa Fathima, Sneha, Sushma B H, Mr. Guruprasad G
Department of CSE, Yenepoya Institute of Technology, Karnataka, India

ABSTRACT

The Drug Recommender system for machine learning-based Drug recommender systems, Deep Drug, is proposed. The framework proposed accepts different various heterogeneous inputs from user and Drug entities, and their knowledge to external and implicit feedbacks. In order to ensure the unified deep architecture of the framework, so that it is easier for retrieving and ranking Drugs, it uses suitable machine learning tools to improve the quality of recommendations. The proposed framework has an additional feature which is flexible and modular, and it can be generalized and distributed easily, and hence it turns out to be a rational choice for the recommendation of Drugs for Drug recommender systems. And this can further be extended for other entities.

I. INTRODUCTION

This system is mainly for the secure recommendation purpose and used for the Drug freaks against tedious processes in searching. The first step in this system is to login to check whether the user has been verified or not, the recommendation will not start unless the user logs in and has at least a single rating. In the Drug recommendation it the system application has two entities: users and items. This paper focuses on the Drug recommender systems which are the core usage functionalities of websites and e-commerce applications, i.e. items=Drugs. In order to overcome the drawbacks, such as scalability, sparsity and cold-start problems. Although this framework is intended for Drug recommender systems, it can be easily extended to other domains such as hospital recommendation system. In such Drug recommender systems, users have preferences for certain items, and these preferences must be obtained from the data [8]. And the one main difficulty is in focal point of designing features (e.g. genre in the Drug recommenders) especially for a huge amount of items manually, is intractable. In such issues, the concept of machine learning plays an important role. And as obvious as it is in Artificial Intelligence, Deep Learning, which in the recent emerging of machine learning, there is an approach mainly for recommender systems. In this paper, we propose a novel unified framework which has certain advantages in contrast with the current frameworks. This has future evolved the recommendation system, and in this case a Drug recommendation system.

II. LITERATURE SURVEY

[1] Machine Learning, Nature Machine learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Machine learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

[2] Distributed representations of words and phrases and their compositionality The recently introduced continuous Skip- gram model is an efficient method for learning high- quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships. In this paper we present several extensions that improve both the quality of the vectors and the training speed. By subsampling of the frequent words we obtain significant speedup and also learn more regular word representations. We also describe a simple alternative to the hierarchical softmax called negative sampling. An inherent limitation of word representations is their indifference to word order and their inability to represent idiomatic phrases. For example, the meanings of "Canada" and "Air" cannot be easily combined to obtain "Air Canada". Motivated by this example, we present a simple method for finding phrases in text, and show that learning good vector representations for millions of phrases is possible.

[3] Collaborative Machine Learning for Recommender Systems Collaborative filtering (CF) is a successful approach commonly used by many recommender systems. Conventional CF-based methods use the ratings given to items by users as the sole source of information for learning to make recommendation. However, the ratings are often very sparse in many applications, causing CF-based methods to degrade significantly in their recommendation performance. To address this sparsity problem, auxiliary information such as item content information may be utilized. Collaborative topic regression (CTR) is an appealing recent method taking this approach which tightly couples the two components that learn from two different sources of information. Nevertheless, the latent representation learned by CTR may not be very effective when the auxiliary information is very sparse. To address this problem, we generalize recently advances in deep learning from i.i.d. input to non-i.i.d. (CF- based) input and propose in this paper a hierarchical Bayesian model called collaborative deep learning (CDL), which jointly performs deep representation learning for the content information and collaborative filtering for the ratings (feedback) matrix. Extensive experiments on three real-world datasets from different domains show that CDL can significantly advance the state of the art.

[4] Neural Collaborative Filtering In recent years, deep neural networks have yielded immense success on speech recognition, computer vision and natural language processing. However, the exploration of deep neural networks on recommender systems has received relatively less scrutiny. In this work, we strive to develop techniques based on neural networks to tackle the key problem in recommendation -- - collaborative filtering - -- on the basis of implicit feedback. Although some recent work has employed Machine learning for recommendation, they primarily used it to model auxiliary information, such as textual descriptions of items

and acoustic features of musics. When it comes to model the key factor in collaborative filtering --- the interaction between user and item features, they still resorted to matrix factorization and applied an inner product on the latent features of users and items. By replacing the inner product with a neural architecture that can learn an arbitrary function from data, we present a general framework named NCF, short for Neural network-based Collaborative Filtering. NCF is generic and can express and generalize matrix factorization under its framework. To supercharge NCF modelling with non-linearities, we propose to leverage a multi-layer perceptron to learn the user-item interaction function. Extensive experiments on two real- world datasets show significant improvements of our proposed NCF framework over the state-of-the-art methods. Empirical evidence shows that using deeper layers of neural networks offers better recommendation performance

III. METHODOLOGY

The dataset used in this research is Drug Review Dataset (Drugs.com) taken from the UCI ML repository [4]. This dataset contains six attributes, name of drug used (text), review (text) of a patient, condition (text) of a patient, useful count (numerical) which suggest the number of individuals who found the review helpful, date (date) of review entry, and a 10- star patient rating (numerical) determining overall patient contentment. It contains a total of 215063 instances. Fig. 1 shows the proposed model used to build a medicine recommender system. It contains four stages, specifically, Data preparation, classification, evaluation, and Recommendation.

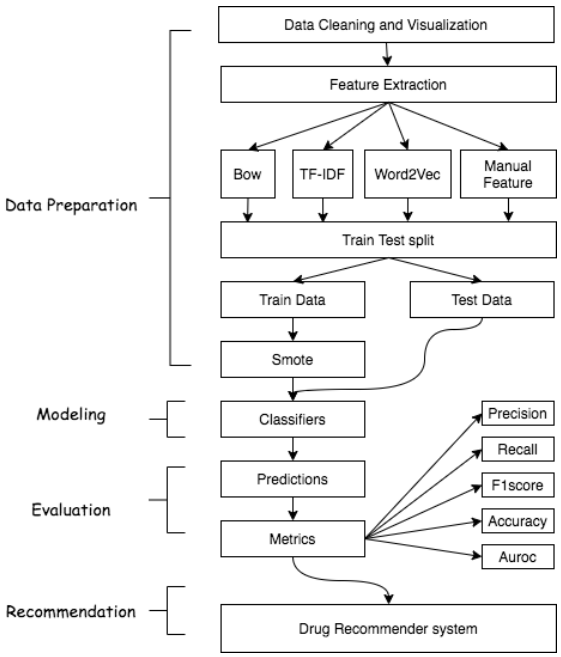

Fig. 1. Flowchart of the proposed model

A. Data Cleaning and Visualisation Applied standard Data preparation techniques like checking null values, duplicate rows, removing unnecessary values, and text from rows in this research. Subsequently, removed all 1200 null values rows in the conditions column, as shown in Fig. 2. We make sure that a unique id should be unique to remove duplicacy.

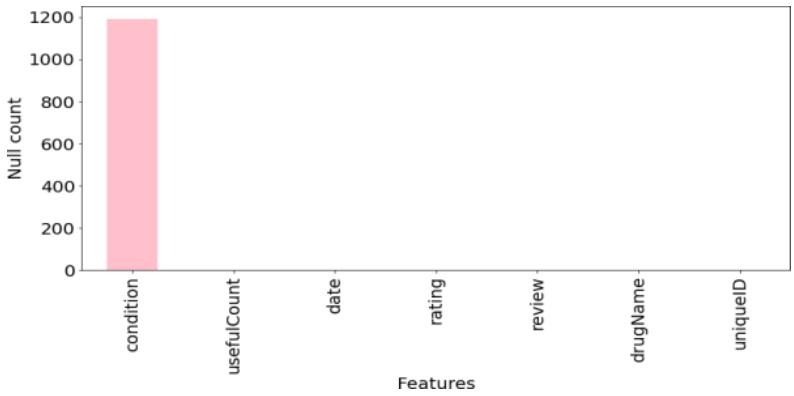

Fig. 2. Bar plot of the number of null values versus attributes

Fig. 3 shows the top 20 conditions that have a maximum number of drugs available. One thing to notice in this figure is that there are two green-colored columns, which shows the conditions that have no meaning. The removal of all these sorts of conditions from final dataset makes the total row count equals to 212141.

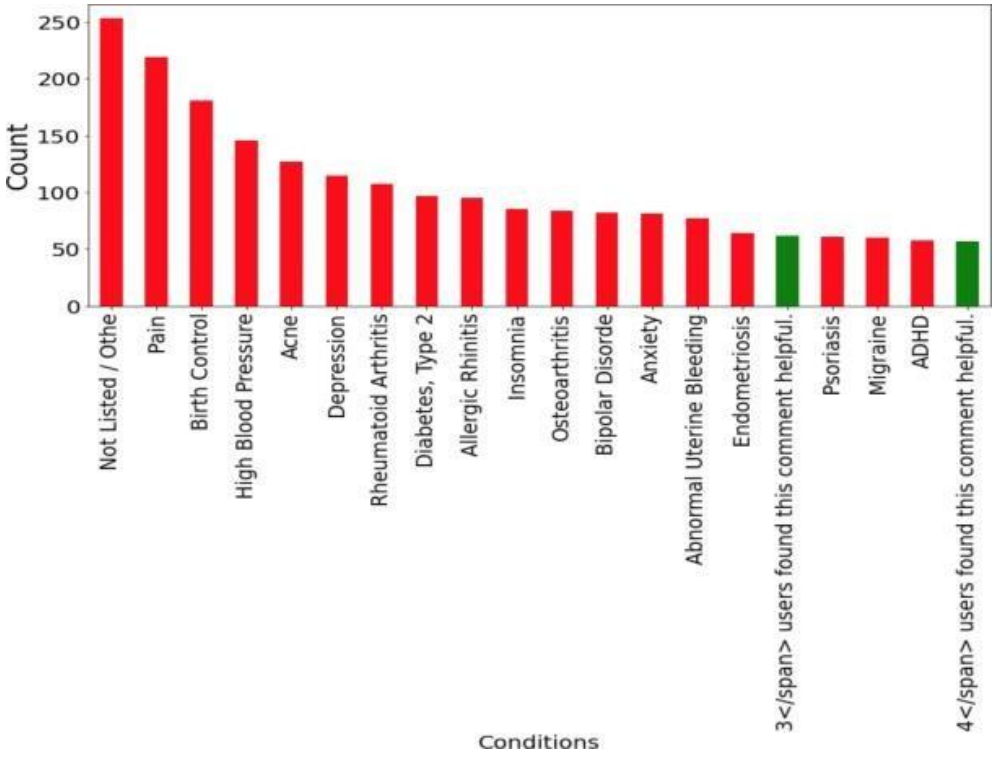

Fig. 3. Bar plot of Top 20 conditions that has a maximum number of drugs available

Fig. 4 shows the visualization of value counts of the 10-star rating system. The rating beneath or equivalent to five featured with cyan tone otherwise blue tone. The vast majority pick four qualities; 10, 9, 1, 8, and 10 are more than twice the same number. It shows that the positive level is higher than the negative, and people's responses are polar. The condition and drug column were joined with review text because the condition and medication words also have predic- tive power. Before proceeding to the feature extraction part, it is critical to clean up the review text before vectorization. This process is also known as text preprocessing. We first cleaned the reviews after removing HTML tags, punctuations, quotes, URLs, etc. The cleaned reviews were lowercased to avoid duplication, and tokenization was performed for converting the texts into small pieces called tokens. Additionally, stopwords.

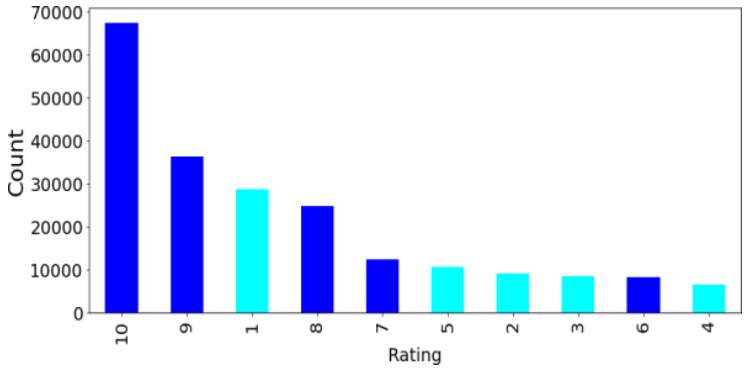

Fig. 4. Bar plot of count of rating values versus 10 rating number

for example, “a, to, all, we, with, etc.,” were removed from the corpus. The tokens were gotten back to their foundations by performing lemmatization on all tokens. For sentiment analysis, labeled every single review as positive and negative based on its user rating. If the user rating range between 6 to 10, then the review is positive else negative.

B. Feature Extraction After text preprocessing, a proper set up of the data required to build classifiers for sentiment analysis. Machine learning algorithms can’t work with text straightforwardly; it should be changed over into numerical format. In particular, vectors of numbers. A well known and straightforward strategy for feature extraction with text information used in this research is the bag of words (Bow) [16], TF-IDF [17], Word2Vec [18]. Also used some feature engineering techniques to extract features manually from the review column to create another model called manual feature aside from Bow, TF-IDF, and Word2Vec.

1) Bow: Bag of words [16] is an algorithm used in natural language processing responsible for counting the number of times of all the tokens in review or document. A term or token can be called one word (unigram), or any subjective number of words, n-grams. In this study, (1,2) n-gram range is chosen. Fig. 5 outlines how unigrams, digrams, and trigrams framed from a sentence. The Bow model experience a significant drawback, as it considers all the terms without contemplating how a few terms are exceptionally successive in the corpus, which in turn build a large matrix that is computationally expensive to train.

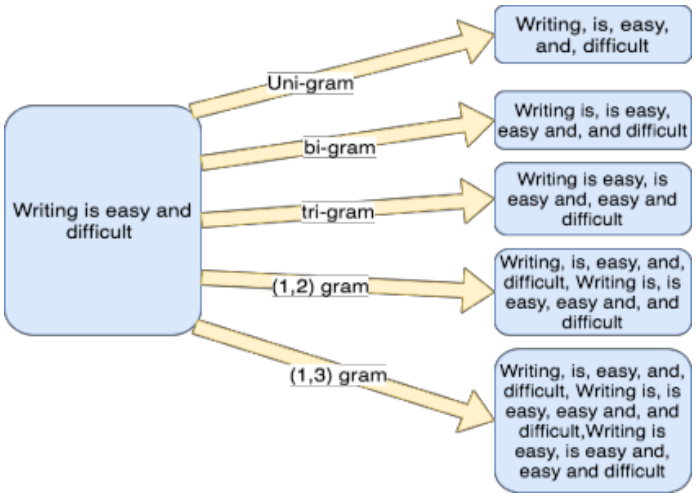

Fig. 5. Comparison of various types of grams framed from a sentence

2) TF-IDF: TF-IDF [17] is a popular weighting strategy in which words are offered with weight not count. The principle was to give low importance to the terms that often appear in the dataset, which implies TF-IDF estimates relevance, not a recurrence. Term frequency (TF) can be called the likelihood of locating a word in a document. tf (t, d) = log(1 + freq(t, d)) (1)

Inverse document frequency (IDF) is the opposite of the number of times a specific term showed up in the whole corpus. It catches how a specific term is document specific. idf (t, d) = log(count(dϵD : tϵd)) (2)

TF-IDF is the multiplication of TF with IDF, suggesting how vital and relevant a word is in the document. tfidf (t, d, D) = tf (t, d).idf (t, D) (3) Like Bow, the selected n-gram range for TF-IDF in this work is (1,2).

3) Word2Vec: Even though TF and TF-IDF are famous vec- torization methods used in different natural language preparing tasks [27], they disregard the semantic and syntactic like- nesses between words. For instance, in both TF and TF- IDF extraction methods, the words lovely and delightful are called two unique words in both TF and TF-IDF vectorization techniques although they are almost equivalents. Word2Vec [18] is a model used to produce word embedding. Word- embeddings reproduced from gigantic corpora utilizing various deep learning models [19]. Word2Vec takes an enormous corpus of text as an input and outputs a vector space, generally composed of hundred dimensions. The fundamental thought was to take the semantic meaning of words and arrange vectors of words in vector space with the ultimate objective that words that share similar sense in the dataset are found close to one another in vectors space.

4) Manual Features: Feature engineering is a popular con- cept which helps to increase the accuracy of the model. We used fifteen features, which include usefulcount, the condition column which is label encoded using label encoder function from Scikit library, day, month, year features were developed from date column using DateTime function using pandas. Textblob toolkit [20] was used to extract the cleaned and uncleaned reviews polarity and added as features along with a total of 8 features generated from each of the text reviews as shown in Table I. C. Train Test Split We created four datasets using Bow, TF-IDF, Word2Vec, and manual features. . These four datasets were split into 75% of training and 25% of testing. While splitting the data, we set an equal random state to ensure the same set of random numbers generated for the train test split of all four generated datasets.

C. Classifiers Distinct machine-learning classification algorithms were used to build a classifier to predict the sentiment. Logistic Regression, Multinomial Naive Bayes, Stochastic gradient descent, Linear support vector classifier, Perceptron, and Ridge classifier experimented with the Bow, TF-IDF model since they are very sparse matrix and applying tree-based classifiers would be very time-consuming. Applied Decision tree, Ran- domForest, LGBM, and CatBoost classifier on Word2Vec and manual features model. A significant problem with this dataset is around 210K reviews, which takes substantial computational power. We selected those machine learning classification al- gorithms only that reduces the training time and give faster predictions.

D. Metrics The predicted sentiment were measured using five metrics, namely, precision (Prec), recall (Rec), f1score (F1), accuracy (Acc.) and AUC score [23]. Let the letter be: Tp = True positive or occurrences

where model predicted the positive sentiment truly, Tn = True negative or occurrences where model predicted the negative class truly, Fp = False positive or occurrences where model predicted the positive class falsely, Fn = False negative or occurrences where model predicted the negative class falsely, Precision, recall, accuracy, and f1score shown in equations given below,

Precision = Tp + Fp (4)

Tp Recall = Tp + Fn Tp + Tn (5)

Accuracy = Tp + Tn + Fp + Fn (6)

F 1score = 2. Precision + Recall (7)

Area under curve (Auc) score helps distinguish a classifier's capacity to compare classes and utilized as a review of the region operating curve (roc) curve. Roc curve visualizes the relationship between true positive rate (Tpr) and false positive rate (Fpr) across various thresholds. G. Drug Recommender system After assessing the metrics, all four best-predicted results were picked and joined together to produce the combined prediction. The merged results were then multiplied with normalized useful count to generate an overall score of drug of a particular condition. The higher the score, the better is the drug. The motivation behind the standardization of the useful count was looking at the distribution of useful count in Fig. 7; one may analyze that the contrast among the least and most extreme is around 1300, considerable. Moreover, the deviation is enormous, which is 36. The purpose behind is that the more medications individuals search for, the more individuals read the survey regardless of their review is positive or negative, which makes the useful count high. So the accuracy achieved by perceptron (91%) using bag of words model. There was a close competition between LinearSVC, perceptron, and ridge classifier, with only a 1% difference. However, LinearSVC was picked as the best algorithm since the AUC score is 90.7%, which is greater than all other algorithms.

TABLE IV TF-IDF

Model	Class	Prec	Rec	F1	Acc.	AUC
LogisticR egression	negative positive	0.79 0.89	0.74 0.92	0.76 0.90	0.86	0.826
Perceptro n	negative positive	0.89 0.93	0.83 0.96	0.86 0.94	0.92	0.895
RidgeCla ssifier	negative positive	0.89 0.93	0.84 0.95	0.86 0.95	0.92	0.897
Multinom ialNB	negative positive	0.85 0.93	0.83 0.94	0.84 0.93	0.90	0.883
SGDClas sifier	negative positive	0.76 0.83	0.57 0.92	0.65 0.88	0.82	0.745
LinearSV C	negative positive	0.89 0.94	0.86 0.96	0.87 0.95	0.93	0.907

The performance metrics of various classification methods on Word2Vec can be analyzed using Table V. The best accuracy is 91% by the LGBM model. Random forest and catboost classifier provide comparable sort of results whereas decision tree classifier performed poorly. Analyzing the region operating curve score, we can easily manifest that the LGBM has the highest AUC score of 88.3%.

TABLE WORD2VEC

Model	Class	Prec	Rec	F1	Acc.	AUC
Decision Tree Classifier	negative positive	0.61 0.86	0.69 0.81	0.65 0.84	0.78	0.751
Random Forest Classifier	negative positive	0.86 0.91	0.77 0.95	0.81 0.93	0.89	0.858
LGBM Classifier	negative positive	0.86 0.93	0.82 0.94	0.84 0.93	0.91	0.883
Cat Boost Classifier	negative positive	0.81 0.91	0.79 0.92	0.80 0.92	0.88	0.855

Table VI displays the performance metrics of four different classification algorithms on manually created features on user reviews. Compared to all other text classification methods, the results are not pretty impressive. However, the random forest achieved a good accuracy score of 88%.

TABLE VI MANUAL FEATURE SELECTION

Model	Class	Prec	Rec	F1	Acc.	AUC
DecisionTree Classifier	negative positive	0.65 0.88	0.75 0.83	0.69 0.85	0.80	0.816
RandomForest Classifier	negative positive	0.79 0.92	0.81 0.91	0.80 0.91	0.88	0.857
LGBM Classifier	negative positive	0.74 0.89	0.74 0.89	0.74 0.89	0.85	0.787
CatBoost Classifier	negative positive	0.72 0.88	0.73 0.88	0.73 0.88	0.84	0.804

After evaluating all the models, the prediction results of Perceptron (Bow), LinearSVC (TF-IDF), LGBM (Word2Vec), and RandomForest (Manual Features) was added to give combined model predictions. The main intention is to make sure that the recommended top drugs should be classified correctly by all four models. If one model predicts it wrong, then the drug's overall score will go down. These combined predictions were then

multiplied with normalized useful count to get an overall score of each drug. This was done to check that enough people reviewed that drug. The overall score is divided by t he total number of drugs per condition to get a mean score, which is the final score. Fig. 8 shows the top four drugs recommended by our model on top five conditions namely, Acne, Birth Control, High Blood Pressure, Pain and Depression.

condition	drugName	Score
Acne	Retin-A	0.069334
Acne	Atralin	0.088545
Acne	Magnesium hydroxide	0.088545
Acne	Retin A Micro	0.097399
Birth Control	Mono-Linyah	0.005448
Birth Control	Gildess Fe 1.5 / 30	0.005987
Birth Control	Ortho Micronor	0.006149
Birth Control	Lybrel	0.027766
High Blood Pressure	Adalat CC	0.303191
High Blood Pressure	Zestril	0.305851
High Blood Pressure	Toprol-XL	0.362589
High Blood Pressure	Labetalol	0.367021
Pain	Neurontin	0.158466
Pain	Nortriptyline	0.171771
Pain	Pamelor	0.231829
Pain	Elavil	0.304513
Depression	Remeron	0.124601
Depression	Sinequan	0.146486
Depression	Provigil	0.240185
Depression	Methylin ER	0.328604

Fig. 8. Recommendation of top four drugs on top five condition

IV. DISCUSSION

The results procured from each of the four methods are good, yet that doesn't show that the recommender framework is ready for real-life applications. It still need improvements. Predicted results show that the difference between the positive and negative class metrics indicates that the training data should be appropriately balanced using algorithms like Smote, Adasyn [24], SmoteTomek [25], etc. Proper hyperparameter optimization is also required for classification algorithms to improve the accuracy of the model. In the recommendation framework, we simply just added the best-predicted result of each method. For better results and understanding, require a proper ensembling of different predicted results. This paper intends to show only the methodology that one can use to extract sentiment from the data and perform classification to build a recommender system.

V. EXPERIMENTAL RESULTS

In this section we will be discussing the results of our implementation and display the snapshots of the application that has been developed. How each module that we discussed in the implementation will be

represented and how the expected results are obtained. The app that has been developed can be shown with a screenshot and how the interactions happen. But the working of the model cannot be displayed in this report.

VI. CONCLUSION

The results procured from each of the four methods are good, yet that doesn't show that the recommender framework is ready for real-life applications. It still need improvements. Predicted results show that the difference between the positive and negative class metrics indicates that the training data should be appropriately balanced using algorithms like Smote, Adasyn [24], SmoteTomek [25], etc. Proper hyperparameter optimization is also required for classification algorithms to improve the accuracy of the model. In the recommendation framework, we simply just added the best-predicted result of each method. For better results and understanding, require a proper ensembling of different predicted results. This paper intends to show only the methodology that one can use to extract sentiment from the data and perform classification to build a recommender system.

VII. ACKNOWLEDGEMENT

The successful completion of any work would be incomplete without a mention of the people who made i t possible, whose constant guidance and encouragement served as a beacon light and crowned my efforts with success. I owe my gratitude to many people who helped and supported me during my Internship/Professional Practice report "Drug Recommendation System based on Sentiment Analysis of Drug Reviews using Machine Learning". My deepest thanks to my project guide Prof. Guruprasad G, Assistant Professor, Dept. of Computer Science & Engineering for his constant support and encouragement and providing with the necessary advice and help. I am highly indebted to him for taking keen interest in my work, monitoring and providing guidance throughout the course. I also thank Mr.Manjunath Raikar, Project Co-ordinator, Assistant Professor, Dept. of Computer Science & Engineering for her constant encouragement and support extended throughout. I sincerely express my gratitude to Prof. Pandu Naik, H.O.D., Dept. Of Computer Science & Engineering for his constant support and guidance for the successful completion of this Internship Report. I take immense pleasure in thanking my beloved Principal Dr. R. G. D'Souza for his constant support. I also thank my lectures who were ready with a positive comment to help me all the time, whether it was an off-hand comment to encourage me or a constructive piece of criticism. At last, but not the least I want to thank my classmates and friends who appreciated my work and motivated me.

VIII. REFERENCES

[1]. Wittich CM, Burkle CM, Lanier WL. Medication errors: an overview for clinicians. Mayo Clin Proc. 2014 Aug;89(8):1116-25.

[2]. CHEN, M. R., & WANG, H. F. (2013). The reason and prevention of hospital medication errors. Practical Journal of Clinical Medicine, 4.

[3]. T. N. Tekade and M. Emmanuel, ”Probabilistic aspect mining approach for interpretation and evaluation of drug reviews,” 2016 International Conference on Signal Processing, Communication, Power and Embed - ded System (SCOPES), Paralakhemundi, 2016, pp. 1471 -1476, doi: 10.1109/SCOPES.2016.7955684.

[4]. Fox, Susannah, and Maeve Duggan. ”Health online 2013. 2013.” URL: http://pewinternet.org/Reports/2013/Health -online.aspx

[5]. Bartlett JG, Dowell SF, Mandell LA, File TM Jr, Musher DM, Fine MJ. Practice guidelines for the management of community-acquired pneumonia in adults. Infectious Diseases Society of America. Clin Infect Dis. 2000 Aug;31(2):347-82. doi: 10.1086/313954. Epub 2000 Sep 7. PMID: 10987697; PMCID: PMC7109923.

[6]. Fox, Susannah & Duggan, Maeve. (2012). Health Online 2013. Pew Research Internet Project Report