History of Medicine, 2021, 7(1): 117-132

DOI:10.48047.v7.1.2021.20

Optimizing Drug Recommendations Through Sentiment Analysis and Machine Learning Techniques

Dr. P. Hasitha Reddy[1], Manisha.S[2], B. Sushmitha[2], J. Vignesh[2], G. Vinay Teja[2]

[1,2]Department of Computer Science and Engineering

[1,2] Sree Dattha Group of Institutions, Sheriguda, Telangana.

ABSTRACT

The COVID-19 pandemic has exacerbated the inaccessibility of legitimate medical resources, leading to a critical shortage of specialists and healthcare professionals, inadequate supplies of essential equipment and medications, and increased mortality rates. In response to these challenges, many individuals have resorted to self-medication without proper medical consultation, further deteriorating their health conditions. Machine learning has emerged as a valuable tool in various applications, and its potential for automation has sparked significant interest in research and development.

This paper presents a sentiment and machine learning-based drug recommendation system designed to alleviate the burden on healthcare professionals. The system takes disease names provided by patients and recommends appropriate medications while simultaneously displaying sentiment ratings based on reviews from previous users. A high predicted rating allows patients to trust and consider the recommended drug. The proposed methodology employs several feature extraction techniques, including Term Frequency-Inverse Document Frequency (TF-IDF), Bag of Words, and Word2Vec. These extracted features are then applied to various machine learning algorithms, such as Logistic Regression, Linear Support Vector Classification (SVC), Ridge Classifier, Naïve Bayes, Multilayer Perceptron (MLP), and Stochastic Gradient Descent (SGD) Classifier. Among these models, the MLP classifier utilizing TF-IDF feature extraction demonstrated superior performance compared to the others. For implementation, the study utilized the DRUGREVIEW dataset from the UCI Machine Learning Repository.

Keywords: Drug recommendation, machine learning, multi-layer perception.

1. INTRODUCTION

1.1 Overview

One of the most concerned and searched topics on the internet is about health information. According to the Pew Internet and American Life Project, almost 60% of grownups are looking for enough health information on the web with 35% of respondents concentrating on diagnosing ailments online only. Since many studies show that number of people die due to the medical errors and the semi errors are caused by medical practitioners, who prescribe medicines based on their experiences. As most of their experiences are limited, they often commit mistakes. This study provides a medicine recommendation system for doctors which can be used by them while prescribing medicines. A recommender framework is an ordinary framework that makes the users get a proposal of things which they can utilize for their exact need. Dissimilar to numerous different kinds of frameworks, health recommendation principally relies upon enthusiastic, physical and mental issues of the patients. A medicine recommendation system is similar system that recommend the medicines for a particular disease based on patient reviews. This system is very essential in this fast-growing technological world, which can save lives by helping doctors. In this paper, the proposed medicine recommendation system and its working is depicted,

wherein it uses the current technologies like machine learning, data mining etc. to find out the interesting records hidden in the medical data and reduce the medical errors by the doctors while prescribing medicines. This system consists of following modules such as database module, data preparation, data visualization, recommendation, and model evaluation module . The proposed medication recommender system uses Machine learning N-Gram and Lightgbm algorithms by using data from hospital and the best one is selected for the medicine recommendation system to attain the metrics like good accuracy, scalability, and model efficiency.

1.2 Motivation

Online consultations require the patient to describe their symptoms to the doctor. A spike in virtual medical services has been reported in the wake of the novel coronavirus disease (COVID-19) [1]. Diabetes, hypertension, and heart disease are all associated with an increased risk of virus infections. The availability of health care professionals 24/7, no need for travel, security, privacy, and drug recommendations are all advantages of virtual medical services. The recommender system allows for improvements in medical services in disparate areas [2]. Often, finding a physician in remote areas can be tricky, so recommender systems have been created to help. Health-related recommender systems can make an early diagnosis, predict disease progression, and make appropriate recommendations according to the health status of patients [3,4]. Machine learning (ML) greatly improves the quality of medical recommender systems by providing suggestions that are based on patient needs and feedback [5,6]. By using sentiment analysis and feature engineering, the drug recommender system can dispense medicine according to a specific condition. Emotions, such as attitudes and opinions, are separated and extracted from language through sentiment analysis [7]. By using the recommender system, information overload can be solved, and egovernment and e-learning can be improved [8]. Depending on an individual's health status, these recommender systems prescribe medications, diagnose diseases, and refer them to the relevant health care. An ML-driven recommendation system generates appropriate recommendations using parameters such as blood pressure, gender, cholesterol levels, and blood sugar for diseases such as colds, fevers, and cardiac deaths [9]. The healthcare system built on the Internet of Things (IoT) coupled with an oncology interface has provided nutrition information to individuals [10]. Depending on the patient's medical history, a decision support system can assist a doctor in prescribing a drug. In contrast, the recommendation system suggests the same based on an analysis of previous usage patterns [11]. Four types of recommender systems exist, including content-driven filtering, collaborative filtering, knowledge-driven recommender systems, and hybrid recommender systems [12,13]. Since the drug recommendation framework includes medical terminology, such as infection names, side effects, and synthetic names, only a limited number of papers are available.

1.3 System Analysis

The main concept behind this drug recommendation system is to build a recommendation system that helps the patient by recommending suitable medicine for their disease. This system offers the medicine recommendation by recommending the drug based on the specific condition dependent on the patient reviews.

2. LITERATURE REVIEW

Bartlett et. al [1] compares on real data effective duplicates detection methods for automatic deduplication of files based on names, working with French texts or English texts, and the names of people or places, in Africa or in the West. After conducting a more complete classification of semantic duplicates than the usual classifications, they introduce several methods for detecting duplicates whose average complexity observed is less than O(2n). Through a simple model, they highlight a global efficacy rate, combining precision and recall. We propose a new metric distance between records, as

well as rules for automatic duplicate detection. Analyses made on a database containing real data for an administration in Central Africa, and on a known standard database containing names of restaurants in the USA, have shown better results than those of known methods, with a lesser complexity. Shimada et. al [2] developed a decision support system that helps doctors select appropriate first-line drugs. The system classifies patients' abilities to protect themselves from infectious diseases as a risk level for infection. In an evaluation of the prototype system, the risk level it determined correlated with the decisions of specialists. The system is very effective and convenient for doctors to use.

He et. al [3] presented a novel adaptive synthetic (ADASYN) sampling approach for learning from imbalanced data sets. The essential idea of ADASYN is to use a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data is generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn.

Lei et. al [4] presented a novel approach to polarity classification of short text snippets, which takes into account the way data are naturally distributed into several topics in order to obtain better classification models for polarity. This approach is multi-step, where in the initial step a standard topic classifier is learned from the data and the topic labels, and in the ensuing step several polarity classifiers, one per topic, are learned from the data and the polarity labels. They empirically show that our approach improves classification accuracy over a real-world dataset by over 10%, when compared against a standard single-step approach using the same feature sets. The approach is applicable whenever training material is available for building both topic and polarity learning models. Nikfarjam and Gonzalez et. al [5] presented a new method for using association rules for colloquial text mining. They applied our method on user comments to find mentions of adverse reactions to drugs by extracting frequent patterns. Since we are dealing with highly informal colloquial text, the idea of using extracted patterns might, at first, seem counter-intuitive. However, we indeed found consistencies in the user comments. This evaluation measured the effectiveness of this technique in extracting frequent patterns in this context. However, this method can easily be generalized for other contexts and languages.

Doulaverakis et. al [6] presented a drug recommendation system based on Semantic Web technologies, termed GalenOW. It has been shown that OWL and Semantic Web technologies can provide a good match for drug recommendations as OWL is expressive enough to effectively encapsulate medical knowledge. Rule-based reasoning can model medical decision making and aid experts. A comparison of the semantic-enabled implementation to a traditional business logic implementation was presented. Although the latter has shown better performance in time and memory requirements, semantic technologies provide a better alternative for integrating knowledge in the system than simple rule engines.

Goeuriot et. al [7] presented creation of lexical resources and their adaptation to the medical domain. We first describe the creation of a general lexicon, containing opinion words from the general domain and their polarity. Then they presented the creation of a medical opinion lexicon, based on a corpus of drug reviews. They show that some words have a different polarity in the general domain and in the medical one. Some words considered generally as neutral are opinionated in medical texts. They finally evaluate the lexicons and show with a simple algorithm that using our general lexicon gives better results than other well-known ones on our corpus and that adding the domain lexicon improves them as well.

Keers et. al [8] appraised empirical evidence relating to the causes of medication administration errors (MAEs) in hospital settings. Limited evidence from studies included in this systematic review suggests

that MAEs are influenced by multiple systems factors, but if and how these arise and interconnect to lead to errors remains to be fully determined. Further theoretical focused is needed to investigate the MAE causation pathway, with an emphasis on ensuring interventions designed to minimise MAEs target recognised underlying causes of errors to maximise their impact.

Wittich et. al [9] provides a practicing physicians that focuses on medication error terminology and definitions, incidence, risk factors, avoidance strategies, and disclosure and legal consequences. A medication error is any error that occurs at any point in the medication use process. It has been estimated by the Institute of Medicine that medication errors cause 1 of 131 outpatient and 1 of 854 inpatient deaths. Medication factors (eg, similar sounding names, low therapeutic index), patient factors (eg, poor renal or hepatic function, impaired cognition, polypharmacy), and health care professional factors (eg, use of abbreviations in prescriptions and other communications, cognitive biases) can precipitate medication errors.

Zhang et. al [10] proposed a novel cloud-assisted drug recommendation (CADRE), which can recommend users with top-N related medicines according to symptoms. In CADRE, they first cluster the drugs into several groups according to the functional description information, and design a basic personalized drug recommendation based on user collaborative filtering. Then, considering the shortcomings of collaborative filtering algorithm, such as computing expensive, cold start, and data sparsity, they propose a cloud-assisted approach for enriching end-user Quality of Experience (QoE) of drug recommendation, by modeling and representing the relationship of the user, symptom and medicine via tensor decomposition. Finally, the proposed approach is evaluated with experimental study based on a real dataset crawled from Internet.

Danushka et. al [11] proposed an unsupervised method for learning domain-specific word representations that accurately capture the domain-specific aspects of word semantics. First, we select a subset of frequent words that occur in both domains as \emph{pivots}. Next, they optimize an objective function that enforces two constraints: for both source and target domain documents, pivots that appear in a document must accurately predict the co-occurring non-pivots, and, word representations learnt for pivots must be similar in the two domains. Moreover, they propose a method to perform domain adaptation using the learnt word representations. This proposed method significantly outperforms competitive baselines including the state-of-the-art domain-insensitive word representations, and reports best sentiment classification accuracies for all domain-pairs in a benchmark dataset.

Sarker et. al [12] suggested that interest in the utilization of the vast amounts of available social media data for ADR monitoring is increasing. In terms of sources, both health-related and general social media data have been used for ADR detection—while health-related sources tend to contain higher proportions of relevant data, the volume of data from general social media websites is significantly higher. There is still very limited amount of annotated data publicly available , and, as indicated by the promising results obtained by recent supervised learning approaches, there is a strong need to make such data available to the work community.

Nikfarjam et. al [13] introduced ADRMine, a machine learning-based concept extraction system that uses conditional random fields (CRFs). ADRMine utilizes a variety of features, including a novel feature for modeling words' semantic similarities. The similarities are modeled by clustering words based on unsupervised, pretrained word representation vectors (embeddings) generated from unlabeled user posts in social media using a deep learning technique. Tekade and Emmanuel et. al [14] used Modeling Based on Probabilistic Approach a more fine-grained aspect level opinion mining. It is interesting to apply the model to find aspects relating to different segmentation of data such as different age groups or other attributes. It is also interesting to work with aspect interpretation as aspects are now

represented by a list of keywords. If a few sentences can be extracted or generated automatically to summarize the keywords, interpretation & understanding will be improved.

Sun et. al [15] aimed at exploiting the rich information in doctor orders and developing data-driven approaches for improving clinical treatments. To this end, they first propose a novel method to measure the similarities between treatment records with consideration of sequential and multifaceted information in doctor orders. Then, they propose an efficient density-based clustering algorithm to summarize large-scale treatment records, and extract a semantic representation of each treatment cluster. Finally, they develop a unified framework to evaluate the discovered treatment regimens, and find the most effective treatment regimen for new patients. In the empirical study, they validate this methods with EMRs of 27,678 patients from 14 hospitals.

3. PROPOSED SYSTEM

A recommender framework is a customary system that proposes an item to the user, dependent on their advantage and necessity. These frameworks employ the customers' surveys to break down their sentiment and suggest a recommendation for their exact need. In the drug recommender system, medicine is offered on a specific condition dependent on patient reviews using sentiment analysis and feature engineering. Sentiment analysis is a progression of strategies, methods, and tools for distinguishing and extracting emotional data, such as opinion and attitudes. On the other hand, Featuring engineering is the process of making more features from the existing ones; it improves the performance of models.

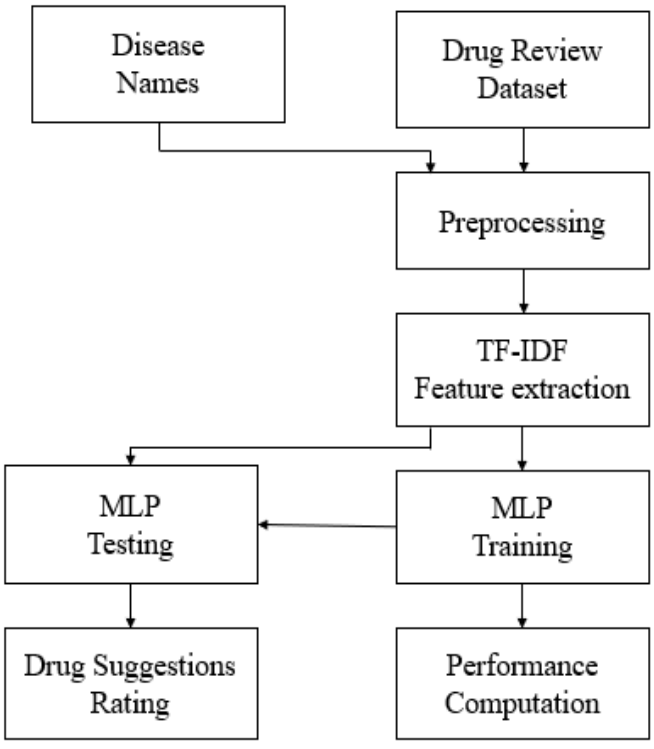

Fig. 1: Block Diagram of Proposed System

3.1 DRUGREVIEW Dataset

3.1.1 Data Set Information

The dataset provides patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting overall patient satisfaction. The data was obtained by crawling online pharmaceutical review sites. The intention was to study

1. sentiment analysis of drug experience over multiple facets, i.e. sentiments learned on specific aspects such as effectiveness and side effects,
2. the transferability of models among domains, i.e. conditions, and
3. the transferability of models among different data sources (see 'Drug Review Dataset (Druglib.com)').

The data is split into a train (75%) a test (25%) partition (see publication) and stored in two .tsv (tab-separated-values) files, respectively. When using this dataset, you agree that you

- only use the data for research purposes
- don't use the data for any commerical purposes
- don't distribute the data to anyone else
- cite us

3.1.2 Attribute Information

- 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

3.2 Preprocessing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data preprocessing task.

3.3 Splitting the Dataset

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

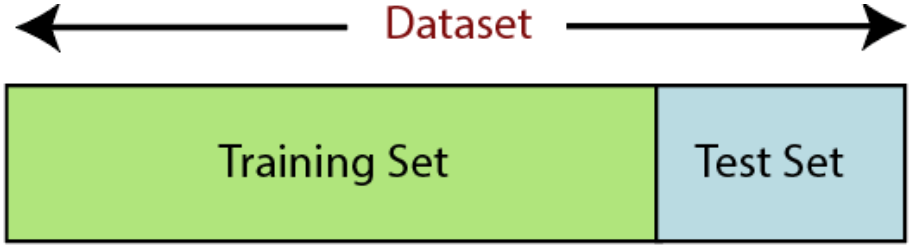

Fig. 2: Splitting the dataset.

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

For splitting the dataset, we will use the below lines of code:

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)
```

3.4 TF-IDF Feature extraction

TF-IDF which stands for Term Frequency – Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document. Let's take an example, we have a string or Bag of Words (BOW) and we have to extract information from it, then we can use this approach.

The tf-idf value increases in proportion to the number of times a word appears in the document but is often offset by the frequency of the word in the corpus, which helps to adjust with respect to the fact that some words appear more frequently in general. TF-IDF use two statistical methods, first is Term Frequency and the other is Inverse Document Frequency. Term frequency refers to the total number of times a given term t appears in the document doc against (per) the total number of all words in the document and The inverse document frequency measure of how much information the word provides. It measures the weight of a given word in the entire document. IDF show how common or rare a given word is across all documents. TF-IDF can be computed as tf * idf

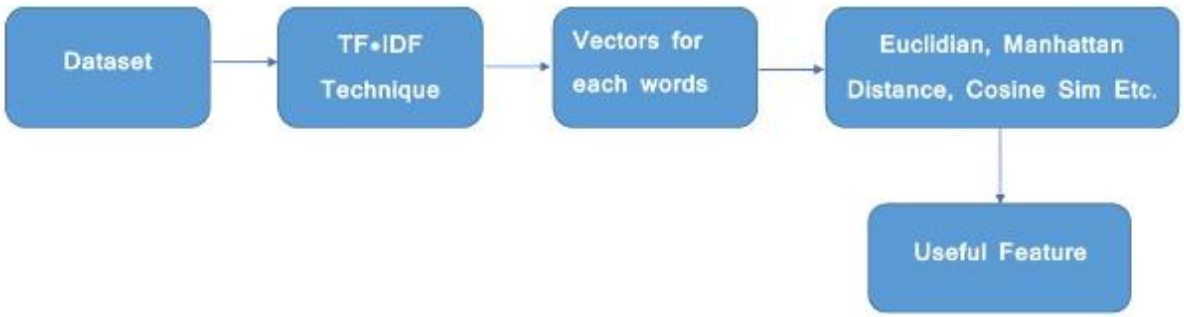

Fig. 3 TF-IDF block diagram.

TF-IDF do not convert directly raw data into useful features. Firstly, it converts raw strings or dataset into vectors and each word has its own vector. Then we'll use a particular technique for retrieving the feature like Cosine Similarity which works on vectors, etc.

Terminology:

t — term (word)

d — document (set of words)

N — count of corpus

corpus — the total document set

Term Frequency (TF): Suppose we have a set of English text documents and wish to rank which document is most relevant to the query, "Data Science is awesome!" A simple way to start out is by eliminating documents that do not contain all three words "Data" is", "Science", and "awesome", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document; the number of times a term occurs in a document is called its term frequency. The weight of a term that occurs in a document is simply proportional to the term frequency.

$$tf(t,d) = count\ of\ t\ in\ d\ /\ number\ of\ words\ in\ d$$

Document Frequency: This measures the importance of document in whole set of corpus, this is very similar to TF. The only difference is that TF is frequency counter for a term t in document d, whereas DF is the count of occurrences of term t in the document set N. In other words, DF is the number of documents in which the word is present. We consider one occurrence if the term consists in the document at least once, we do not need to know the number of times the term is present.

$$df(t) = occurrence\ of\ t\ in\ documents$$

Inverse Document Frequency (IDF): While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by computing IDF, an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely. **The** IDF is the inverse of the document frequency which measures the informativeness of term t. When we calculate IDF, it will be very low for the most occurring words such as stop words (because stop words such as "is" is present in almost all of the documents, and N/df will give a very low value to that word). This finally gives what we want, a relative weightage.

$$idf(t) = N/df$$

Now there are few other problems with the IDF, in case of a large corpus,say 100,000,000 , the IDF value explodes , to avoid the effect we take the log of idf . During the query time, when a word which is not in vocab occurs, the df will be 0. As we cannot divide by 0, we smoothen the value by adding 1 to the denominator.

$$idf(t) = log(N/(df + 1))$$

The TF-IDF now is at the right measure to evaluate how important a word is to a document in a collection or corpus. Here are many different variations of TF-IDF but for now let us concentrate on this basic version.

$$tf - idf(t,d) = tf(t,d) * log(N/(df + 1))$$

Implementing TF-IDF: To make TF-IDF from scratch in python, let's imagine those two sentences from different document:

first_sentence : "Data Science is the sexiest job of the 21st century".

second_sentence : "machine learning is the key for data science".

First step we have to create the TF function to calculate total word frequency for all documents.

3.5 Multilayer perceptron (MLP)

3.5.1 Perceptron

Although today the Perceptron is widely recognized as an algorithm, it was initially intended as an image recognition machine. It gets its name from performing the human-like function of perception, seeing, and recognizing images.

In particular, interest has been centered on the idea of a machine which would be capable of conceptualizing inputs impinging directly from the physical environment of light, sound, temperature, etc. — the "phenomenal world" with which we are all familiar — rather than requiring the intervention of a human agent to digest and code the necessary information. Rosenblatt's perceptron machine relied on a basic unit of computation, the neuron. Just like in previous models, each neuron has a cell that receives a series of pairs of inputs and weights. The major difference in Rosenblatt's model is that inputs are combined in a weighted sum and, if the weighted sum exceeds a predefined threshold, the neuron fires and produces an output.

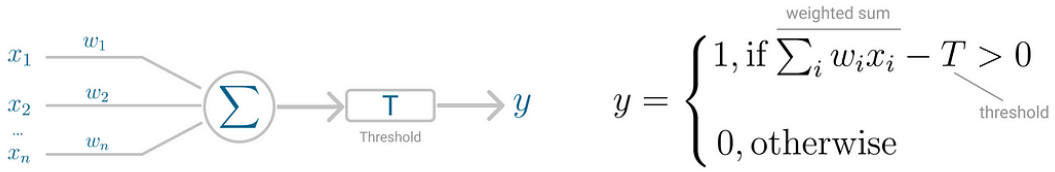

Fig. 4: Perceptron neuron model (left) and threshold logic (right).

Threshold *T* represents the activation function. If the weighted sum of the inputs is greater than zero the neuron outputs the value 1, otherwise the output value is zero.

Perceptron for Binary Classification

With this discrete output, controlled by the activation function, the perceptron can be used as a binary classification model, defining a linear decision boundary.

It finds the separating hyperplane that minimizes the distance between misclassified points and the decision boundary. The perceptron loss function is defined as below:

$$\underbrace{D(w,c)}_{\text{distance}} = -\sum_{i \in \mathrm{M}} \overbrace{y_i}^{\text{output}} (x_i w_i + c)$$

misclassified observations

To minimize this distance, perceptron uses stochastic gradient descent (SGD) as the optimization function. If the data is linearly separable, it is guaranteed that SGD will converge in a finite number of steps. The last piece that Perceptron needs is the activation function, the function that determines if the neuron will fire or not. Initial Perceptron models used sigmoid function, and just by looking at its shape, it makes a lot of sense! The sigmoid function maps any real input to a value that is either 0 or 1 and encodes a non-linear function. The neuron can receive negative numbers as input, and it will still be able to produce an output that is either 0 or 1.

But, if you look at Deep Learning papers and algorithms from the last decade, you'll see the most of them use the Rectified Linear Unit (ReLU) as the neuron's activation function. The reason why ReLU became more adopted is that it allows better optimization using SGD, more efficient computation and is scale-invariant, meaning, its characteristics are not affected by the scale of the input.

The neuron receives inputs and picks an initial set of weights random. These are combined in weighted sum and then ReLU, the activation function, determines the value of the output.

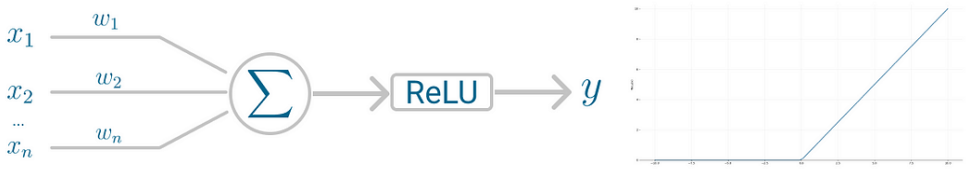

Fig. 5: Perceptron neuron model (left) and activation function (right).

Perceptron uses SGD to find, or you might say learn, the set of weight that minimizes the distance between the misclassified points and the decision boundary. Once SGD converges, the dataset is separated into two regions by a linear hyperplane. Although it was said the Perceptron could represent any circuit and logic, the biggest criticism was that it couldn't represent the XOR gate, exclusive OR, where the gate only returns 1 if the inputs are different. This was proved almost a decade later and highlights the fact that Perceptron, with only one neuron, can't be applied to non-linear data.

3.5.2 MLP

The MLP was developed to tackle this limitation. It is a neural network where the mapping between inputs and output is non-linear. A MLP has input and output layers, and one or more hidden layers with many neurons stacked together. And while in the Perceptron the neuron must have an activation function that imposes a threshold, like ReLU or sigmoid, neurons in a MLP can use any arbitrary activation function.

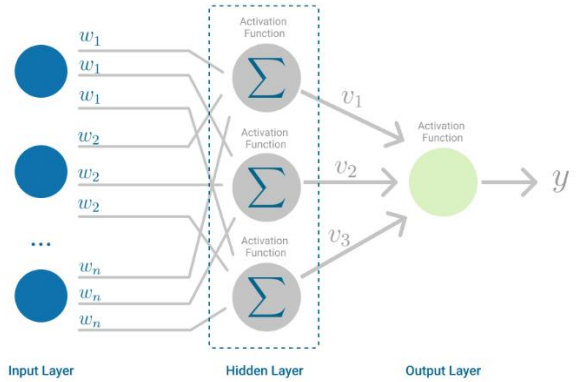

Fig. 6: Architecture of MLP.

MLP falls under the category of feedforward algorithms, because inputs are combined with the initial weights in a weighted sum and subjected to the activation function, just like in the Perceptron. But the difference is that each linear combination is propagated to the next layer. Each layer is feeding the next one with the result of their computation, their internal representation of the data. This goes all the way through the hidden layers to the output layer.

If the algorithm only computed the weighted sums in each neuron, propagated results to the output layer, and stopped there, it wouldn't be able to learn the weights that minimize the cost function. If the algorithm only computed one iteration, there would be no actual learning. This is where Backpropagation comes into play.

Backpropagation

Backpropagation is the learning mechanism that allows the MLP to iteratively adjust the weights in the network, with the goal of minimizing the cost function. There is one hard requirement for backpropagation to work properly.

The function that combines inputs and weights in a neuron, for instance the weighted sum, and the threshold function, for instance ReLU, must be differentiable. These functions must have a bounded derivative because Gradient Descent is typically the optimization function used in MLP.

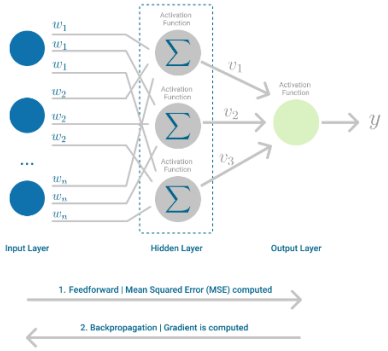

Fig. 7: MLP, highlighting the Feedforward and Backpropagation steps.

In each iteration, after the weighted sums are forwarded through all layers, the gradient of the Mean Squared Error is computed across all input and output pairs. Then, to propagate it back, the weights of the first hidden layer are updated with the value of the gradient. That's how the weights are propagated back to the starting point of the neural network. One iteration of Gradient Descent is defined as follows:

$$\underbrace{\Delta_w(t)}_{\substack{\text{Gradient}\\ \text{Current Iteration}}} = \overbrace{-}^{\text{Bias}}\varepsilon\frac{\overbrace{dE}^{\text{Error}}}{\underbrace{dw_{(t)}}_{\text{Weight vector}}} + \overbrace{\alpha}^{\text{Learning Rate}}\underbrace{\Delta_{w(t-1)}}_{\substack{\text{Gradient}\\ \text{Previous Iteration}}}$$

This process keeps going until gradient for each input-output pair has converged, meaning the newly computed gradient hasn't changed more than a specified convergence threshold, compared to the previous iteration.

4. RESULTS AND DISCUSSION

4.1 Modules

To implement this project, we have designed following modules

1) Upload Drug Review Dataset: using this module we will upload dataset to application
2) Read & Preprocess Dataset: using this module we will read all reviews, drug name and ratings from dataset and form a features array.
3) TF-IDF Features Extraction: features array will be input to TF-IDF algorithm which will find average frequency of each word and then replace that word with frequency value and form a vector. If word does not appear in sentence then 0 will be put. All reviews will be considered as input features to machine learning algorithm and RATINGS and Drug Name will be consider as class label.
4) Train Machine Learning Algorithms: using this module we will input TF-IDF features to all machine learning algorithms and then trained a model and this model will be applied on test data to calculate prediction accuracy of the algorithm.

5) Comparison Graph: using this module we will plot accuracy graph of each algorithm

6) Recommend Drug from Test Data: using this module we will upload disease name test data and then ML will predict drug name and ratings.

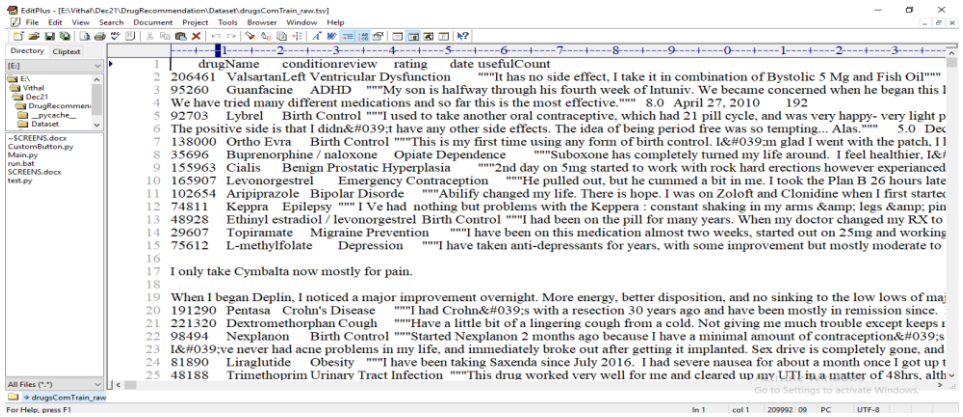

Fig. 8: Sample dataset.

In above screen first row represents dataset column names such as drug name, condition, review and rating and remaining rows contains dataset values and we will used above REVIEWS and RATINGS to trained machine learning models. Below is the test data which contains only disease name and machine learning will predict Drug name and ratings.

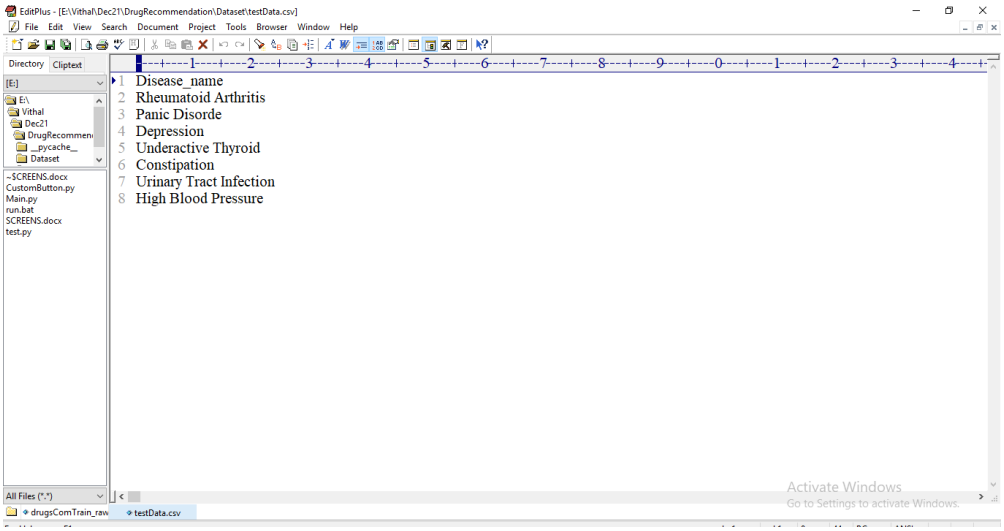

Fig. 9: Dataset properties.

In above test data we have only disease name and machine learning will predict ratings and drug names.

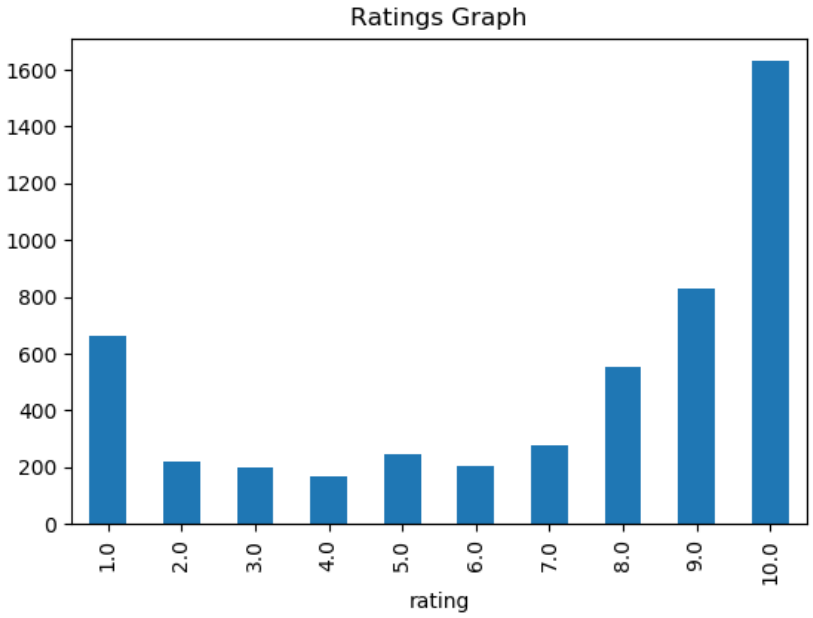

Fig. 10: Drugs ratings graph.

In above graph we can see dataset loaded and in graph x-axis represents ratings and y-axis represents total number of records which got that rating. Now close above graph and then click on 'Read & Pre-process Dataset' button to read all dataset values and then pre-process to remove stop words and special symbols and then form a features array.

In below screen we can see from all reviews stop words and special symbols are removed and in graph I am displaying TOP 20 medicines exist in dataset. In above graph x-axis represents drug name and y-axis represents its count.

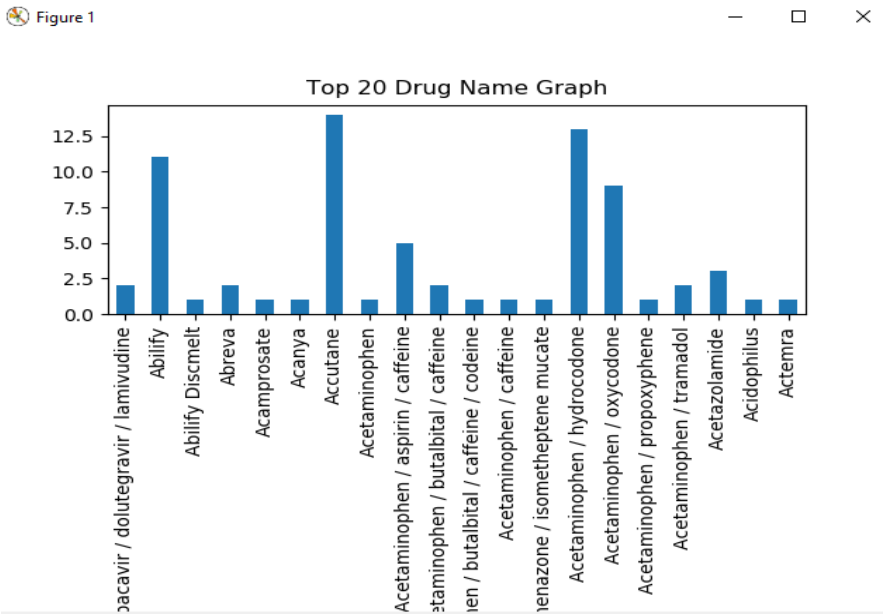

Fig. 11: Drug names dataset.

Table 1. Performance comparison.

Method	Precision	Recall	F1-Score	Accuracy
Existing Logistic regression	80.54	79.30	79.27	76
Existing SVC	70.51	71.18	70,46	67.80
Existing Ridge classifier	66.786	37.72	42.78	55.1
Existing Multimodal navie bayes	41.32	47.98	43.14	47.19
Existing SGDC	41.324	47.18	43.44	47.49
Proposed MLP	99.96	99.72	99.84	99.9

In above table for each algorithm we calculate accuracy, precision, recall and FSCORE and in all algorithms MLP has got high performance.

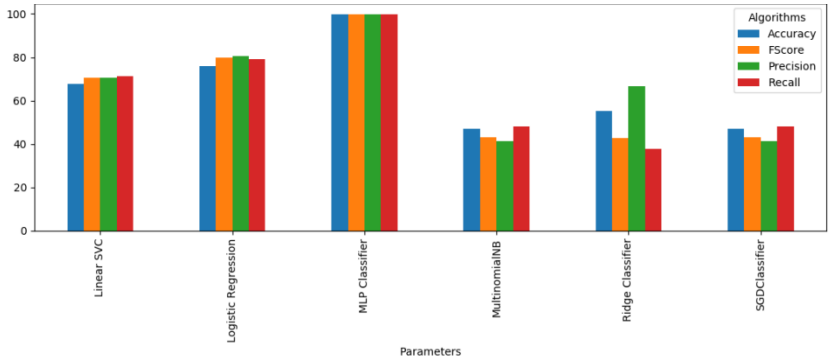

Fig. 12: Performance comparison graph.

In above graph x-axis represents algorithm name and y-axis represents accuracy, precision recall and FSCORE where each different colour bar will represent one metric and in above graph we can see MLP got high performance.

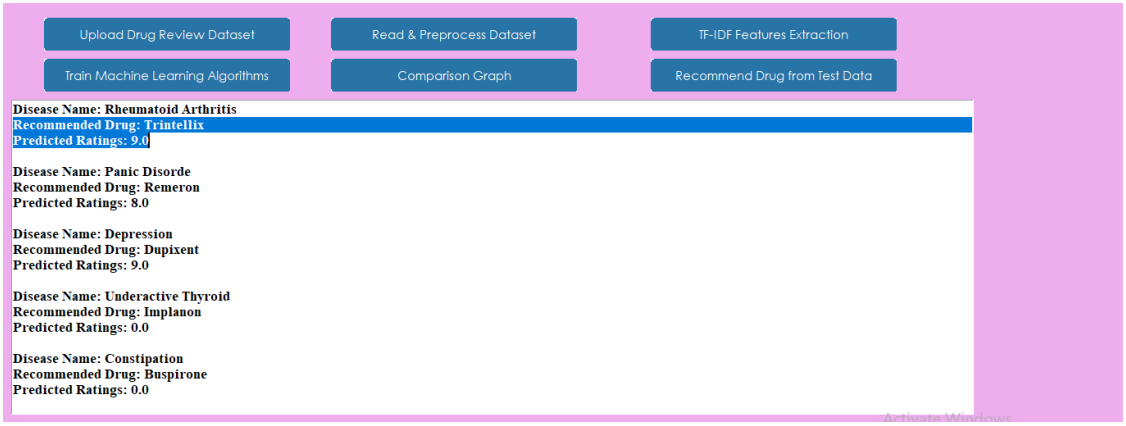

Fig. 13: Drug recommendations from test data.

In above screen for each disease name application has predicted recommended drug name and ratings.

5. CONCLUSION AND FUTURE SCOPE

Optimizing Drug Recommendations Through Sentiment Analysis and Machine Learning Techniques

Reviews are becoming an integral part of our daily lives; whether go for shopping, purchase something online or go to some restaurant, we first check the reviews to make the right decisions. Motivated by this, in this research sentiment analysis of drug reviews was studied to build a recommender system using different types of machine learning classifiers, such as Logistic Regression, MLP, Multinomial Naive Bayes, Ridge classifier, Stochastic gradient descent, LinearSVC, applied on TF-IDF features. We evaluated them using five different metrics, precision, recall, f1score, accuracy, and AUC score, which reveal that the MLP on TF-IDF outperforms all other models with 99% accuracy.

Future work

Future work involves comparison of different oversampling techniques, using different values of n-grams, and optimization of algorithms to improve the performance of the recommender system. Emergencies such as pandemics, floods, or cyclones can be helped by the medical recommender system. In the era of deep learning, recommender systems produce more accurate, quick, and reliable clinical predictions with minimal costs. As a result, these systems maintain better performance, integrity, and privacy of patient data in the decision-making process and provide precise information at any time.

REFERENCES

[1] J. Ramos. “Using tf-idf to determine word relevance in document queries”, in Proceedings of the first instructional conference on machinelearning, vol. 242, pp. 133–142, Piscataway, NJ, 2003

[2] K. Shimada, H. Takada, S. Mitsuyama, H. Ban, H. Matsuo, H. Otake, H. Kunishima, K. Kanemitsu and M. Kaku. “Drug-recommendation system for patients with infectious diseases”. AMIA Annu Symp Proc. 2005;2005:1112. PMID: 16779399; PMCID: PMC1560833.

[3] H. He, Y. Bai, E. A. Garcia and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning”, 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 1322-1328, doi: 10.1109/IJCNN.2008.4633969.

[4] X. Lei, G. Anna Lisa, M. James and J. Iria. 2009. “Improving Patient Opinion Mining Through Multi-step Classification. In Proceedings of the 12th International Conference on Text, Speech and Dialogue (TSD ’09)”. Springer-Verlag, Berlin, Heidelberg, 70–76. https://doi.org/10.1007/978-3-642-04208-9_13.

[5] A. Nikfarjam and G. H. Gonzalez. “Pattern mining for extraction of mentions of Adverse Drug Reactions from user comments”. AMIA Annu Symp Proc. 2011;2011:1019-26. Epub 2011 Oct 22. PMID: 22195162; PMCID: PMC3243273.

[6] C. Doulaverakis, G. Nikolaidis and A. Kleontas. “GalenOWL: Ontology-based drug recommendations discovery”. J Biomed Semant 3, 14 (2012). https://doi.org/10.1186/2041-1480-3-14.

[7] L. Goeuriot, J. C. Na, W. Y. M. Kyaing, C. Khoo, Y. K. Chang, Y. L. Theng, and J. Kim. 2012. “Sentiment Lexicons for Healthrelated Opinion Mining. In Proceedings of the 2Nd ACM SIGHIT International Health Informatics Symposium (IHI ’12)”. ACM, New York, NY, USA, 219–226. https://doi.org/10.1145/2110363.2110390.

[8] R. N. Keers, S. D. Williams, J. Cooke and D. M. Ashcroft. “Causes of medication administration errors in hospitals: a systematic review of quantitative and qualitative evidence”. Drug Saf. 2013 Nov;36(11):1045-67. doi: 10.1007/s40264-013-0090-2. PMID: 23975331; PMCID: PMC3824584.

[9] C. M. Wittich, C. M. Burkle and W. L. Lanier. “Medication errors: an overview for clinicians. Mayo Clin Proc. 2014 Aug;89(8):1116-25”. doi: 10.1016/j.mayocp.2014.05.007. Epub 2014 Jun 27. PMID: 24981217.

[10] Y. Zhang, D. Zhang and M. M. Hassan. “CADRE: Cloud-Assisted Drug REcommendation Service for Online Pharmacies”. Mobile Netw Appl 20, 348–355 (2015). https://doi.org/10.1007/s11036-014-0537-4.

[11] B. Danushka, M. Takanori and K. Kenichi. “Unsupervised Cross-Domain Word Representation Learning”, 2015;arXiv:1505.07184

[12] A. Sarker, R. Ginn, A. Nikfarjam, K. O’Connor, K. Smith, J. Swetha, U. Tejaswi, G. Gonzalez. “Utilizing social media data for pharmacovigilance: A review, Journal of Biomedical Informatics”, Vol. 54, 2015, pp. 202-212, no. 1532-0464, https://doi.org/10.1016/j.jbi.2015.02.004.

[13] A. Nikfarjam, A. Sarker, K. O'Connor, R. Ginn and G. Gonzalez. “Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features”, J Am Med Inform Assoc. 2015 May;22(3):671-81. doi: 10.1093/jamia/ocu041. Epub 2015 Mar 9. PMID: 25755127; PMCID: PMC4457113.

[14] 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 Embedded System (SCOPES), 2016, pp. 1471-1476, doi: 10.1109/SCOPES.2016.7955684.

[15] L. Sun, C. Liu, C. Guo, H. Xiong, and Y. Xie. 2016. “Data-driven Automatic Treatment Regimen Development and Recommendation”. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16). Association for Computing Machinery, New York, NY, USA, 1865–1874. DOI:https://doi.org/10.1145/2939672.2939866.