Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 6, No. 6, June : 2023

MULTI-LAYER PERCEPTION FOR DRUG RECOMMENDATION SYSTEM BASED ON SENTIMENT ANALYSIS OF DRUG REVIEWS

Sowjanya G, M. Tech, Department of CSE, Swarnandhra College of Engineering & Technology(A), Narsapur, Andhra Pradesh, India. Email: gsowjanya.gs@gmail.com
M. Satyanarayana, Associate Professor, Department of CSE, Swarnandhra College of Engineering & Technology(A), Narsapur, Andhra Pradesh, India. Email: satya.scet@gmail.com
Dr. P. Srinivasulu, Professor & HOD, Department of CSE, Swarnandhra College of Engineering &Technology(A), Narsapur, Andhra Pradesh, India. Email: drspamidi@gmail.com

Abstract
In recent years, the availability of online platforms for sharing drug reviews and experiences has led to an overwhelming amount of unstructured data. Extracting meaningful insights from these reviews is crucial for building effective drug recommendation systems. In this study, we propose a multi-layer perception (MLP) model for a drug recommendation system based on sentiment analysis of drug reviews. The first step in our methodology is dataset preprocessing. We gather a large collection of drug reviews from various online sources. To ensure the quality of the data, we perform text cleaning operations such as removing special characters, punctuation, and stop words. We also handle common challenges like misspellings and abbreviations, employing techniques like stemming and lemmatization to standardize the text. Next, we employ TF-IDF (Term Frequency-Inverse Document Frequency) for feature extraction. TF-IDF assigns weights to each term in the review based on its frequency and rarity in the entire corpus. This approach allows us to capture the importance of each term in the review while considering its relevance in the broader context. The TF-IDF values serve as features for training our MLP classifier. The MLP classifier is designed to learn from the extracted features and classify drug reviews into sentiment categories such as positive, negative, or neutral. We train the MLP model using a labeled dataset of drug reviews, where each review is associated with a sentiment label. The MLP architecture consists of multiple layers of interconnected neurons, enabling it to capture complex relationships between the input features and sentiment labels. During the training process, we optimize the model's parameters using backpropagation and gradient descent. To evaluate the performance of our drug recommendation system, we conduct experiments on a separate test drug review dataset.

Keywords: Multi-Layer Perception, Drug Recommendation System, Sentiment Analysis, Drug Reviews, Dataset Preprocessing, TF-IDF, Feature Extraction.

1. Introduction

Online is one place where individuals go to share their concerns and learn more about various subjects. About 60% of Americans use the internet for health information needs, with 35% of respondents focusing only on online sickness diagnosis, according to the Pew Internet and American Life Project [1]. Since several studies demonstrate that a significant portion of fatal medical errors and semi-mistakes are caused by medical professionals who prescribe medications based on their expertise. They often make errors since most of their knowledge is restricted. This research offers physicians a method for writing prescriptions for medications that they can employ. A popular structure called a "recommender framework" suggests products that customers could utilize to meet their specific needs [2]. The patient's physical, mental, and emotional issues are often considered while providing medical advice, in contrast to the approach used by many other models. A comparable system that offers drugs for a specific condition in response to patient input is the pharmaceutical recommendation system. This technology is essential in today's rapidly changing technological environment since it aids physicians in saving lives [3]. The suggested medication recommendation system is discussed in this

article along with how it functions. It does this by using cutting-edge technologies like as machine learning and data mining to uncover the intriguing records that are buried inside medical data and to reduce the number of medical errors that are committed by physicians while they are prescribing pharmaceuticals. The database module, the data preparation module, the data display module, the recommendation module, and the model evaluation module are the parts that make up this system [4]. The patient is the one who is tasked with supplying the attending physician with a description of their symptoms while participating in an online consultation. Since the discovery of the newly discovered coronavirus infection (COVID-19), there has been a noticeable rise in the usage of online medical services. This is attributed to the fact that online medical services are more convenient [5]. There is a correlation between an increased risk of developing viral infections and several other diseases, including diabetes, hypertension, and heart disease. The availability of certified medical professionals at any time of day or night, removal of the need to travel to a physical place, greater levels of security and privacy, and individualized suggestions for medicine are just some of the perks that come along with utilizing virtual medical services. Changes to medical services is implemented in a variety of settings thanks to the recommender system. Since it is possible to have a tough time finding a doctor in a distant location, recommender systems have been created to provide support to people in this situation [6]. It is possible for recommender systems that are tied to health to give an accurate diagnosis, predict the progression of an ailment, and provide pertinent recommendations. Machine learning (ML) [7] has been shown to considerably enhance the quality of medical recommender systems by enabling the systems to provide suggestions that are based on the needs of patients and the replies that patients offer. Using sentiment analysis and feature engineering, the drug recommendation system can provide a pharmaceutical prescription in line with a certain condition [8]. The process of identifying and extracting feelings from words, including attitudes and points of view, is what is known as sentiment analysis. When a recommender system is used, it is possible to minimize information overload, and it may also be possible for both e-learning and e-government to notice potential improvements. Based on the individual's present health status, these recommender systems may identify illnesses, provide ideas for therapies, and even prescribe medications. When it comes to treating disorders such as colds, fevers, and cardiac deaths, the finest recommendations for treatment is found in a recommendation system that is driven by ML [9]. These suggestions are determined by factors such as the individual's blood pressure, gender, cholesterol levels, and blood sugar levels. People now have access to information on their diets because of the implementation of an oncology interface inside a healthcare system that is based on the Internet of Things (IoT) [10]. When prescribing medicine to a patient, a doctor could benefit from using a decision support system that considers the patient's prior medical issues and helps them choose the proper prescription. On the other hand, the suggestions that are generated by the recommendation system are based on an analysis of the use patterns that have come before. Collaborative filtering, knowledge-driven recommender systems, content-driven recommender systems, and hybrid recommender systems are the four distinct subcategories of recommender systems. Because the pharmaceutical recommendation system employs medical language, such as the names of infections, side effects, and synthetic names, only a select number of articles are available for access.

2. Literature review

In [11] authors assessed effective duplicate detection techniques using actual data. They worked with French or English texts, names of individuals or places, in Africa or the West. They provided a thorough characterization of semantic duplication, surpassing typical approaches, to find duplicates with an average complexity less than O(2n). They introduced a worldwide effectiveness rate that combined memory and accuracy using a straightforward technique. The study also introduced new record separation metrics and guidelines for automatic duplicate detection. The examination of a database with actual data for a Central African administration and a well-known standard database

with names of restaurants in the USA yielded better conclusions compared to other established approaches with lower complexity. In [12] authors developed a decision support tool for doctors to choose the best first-line medications. The technique estimates an individual's risk of infection based on their capacity to defend against infectious illnesses. The evaluation of the prototype system's risk estimation during a test was consistent with the choices made by experts. The technique is user-friendly for physicians and highly successful. In [13] authors introduced a unique adaptive synthetic (ADASYN) sampling strategy. ADASYN focuses on the challenge level of understanding each minority class example. It generates more synthetic data for minority class examples that are more challenging to learn compared to simpler minority class examples. In [14] authors proposed an innovative method for polarity classification of short text excerpts. The method considers the division of data into multiple categories. It involves training a common topic classifier in the first step, and learning multiple polarity classifiers, one for each subject, in the second step. Experimental results showed that their method significantly improved classification accuracy compared to traditional single-step strategies, achieving an accuracy increase of over 10%. In [15] authors suggested a novel approach using association rules for informal text mining. They applied this technique to user comments to identify common patterns and allusions to drug side effects. Despite the casual nature of the information, they observed meaningful patterns in the user comments. The method proved useful in identifying patterns in this context and can be adapted to different settings and languages.

In [16] authors developed GalenOW, a drug recommendation system using Semantic Web technology. The study demonstrated the suitability of OWL and Semantic Web technologies for pharmaceutical recommendations, as OWL can incorporate medical information and rule-based reasoning can emulate medical decision-making. The benefits of semantic technologies were highlighted, even though traditional methods showed better performance in terms of time and memory requirements. In [17] authors discussed the creation of lexical resources and their use in the medical field. They developed a comprehensive lexicon that included divisive terms from other fields and used a corpus of medication reviews to build a language for medical opinions. The study showed that certain terminologies have distinct polarity in both general and medical domains. The general lexicon outperformed other well-known lexicons in their corpus, and including the domain lexicon further improved results using a simple method. In [18] authors investigated factors contributing to medication administration errors (MAEs) in hospital settings using empirical data. The systematic review indicated that various system characteristics impact MAEs, although it is unclear how these components interact and contribute to mistakes. The study highlighted the need for further theoretical research to understand the causal pathway of MAEs and ensure interventions target root causes to maximize their effectiveness. In [19] authors focused on terminology and vocabulary related to pharmaceutical errors, including occurrence, risk factors, avoidance strategies, disclosure, and legal implications. The study provides useful information for practicing physicians, defining pharmaceutical errors as any mistakes made when taking medication. The Institute of Medicine estimates that 1 out of 131 outpatient deaths and 1 out of 854 fatal inpatient hospitalizations are related to medication errors. In [20] authors developed a state-of-the-art cloud-assisted drug recommendation (CADRE) system that matches users with the most appropriate medications based on their symptoms. CADRE categorizes medications based on functional description data and utilizes user collaborative filtering to generate personalized medication suggestions. The authors recommend using tensor decomposition to enhance the Quality of Experience (QoE) for medication recommendations by defining and displaying the link between the user, symptom, and therapy. This approach aims to improve QoE by addressing the limitations of cold-start, computationally expensive, and data-poor collaborative filtering methods. Experimental research using a practical dataset downloaded from the internet was conducted to assess the proposed strategy.

3. Proposed Methodology

In our study, we conducted a detailed analysis of the performance of our MLP model for a drug recommendation system based on sentiment analysis of drug reviews. Figure 1 shows the proposed block diagram. We gathered a large collection of drug reviews from online sources and performed dataset preprocessing to ensure data quality. This involved text cleaning operations, such as removing special characters, punctuation, and stop words. We also handled challenges like misspellings and abbreviations by employing techniques like stemming and lemmatization to standardize the text. Then, we employed the TF-IDF technique for feature extraction. TF-IDF assigns weights to each term in the drug reviews based on its frequency and rarity in the entire corpus. This approach allows us to capture the importance of each term in the review while considering its relevance in the broader context. The TF-IDF values served as features for training our MLP classifier. We trained the MLP model using a labeled dataset of drug reviews, where each review was associated with a sentiment label (positive, negative, or neutral). The MLP architecture consisted of multiple layers of interconnected neurons, enabling it to capture complex relationships between the input features and sentiment labels. During the training process, we optimized the model's parameters using backpropagation and gradient descent. To evaluate the performance of our drug recommendation system, we conducted experiments on a separate test dataset consisting of drug reviews. We measured standard evaluation metrics such as accuracy, precision, recall, and F1-score to assess the effectiveness of our MLP model in classifying drug reviews based on sentiment. These metrics provide insights into the model's ability to correctly classify reviews and its overall performance. We compared our MLP-based approach with other state-of-the-art sentiment analysis techniques to highlight its strengths and limitations. By comparing the performance metrics, we can determine how our model fares in comparison to existing methods and identify any improvements or areas of further research. Based on the evaluation metrics and comparisons, we interpreted the results of our analysis. This involved assessing the accuracy and effectiveness of our drug recommendation system in classifying drug reviews and providing meaningful insights.

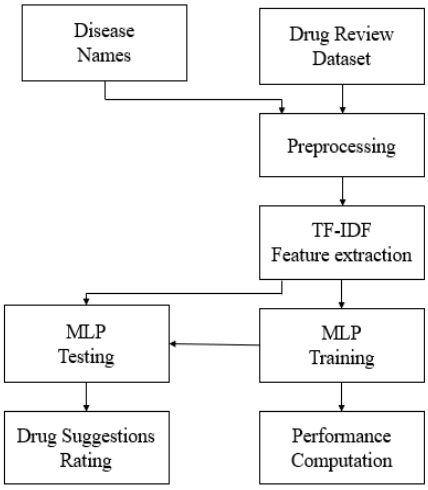

Fig. 1: Block diagram of the suggested system.

3.1 Pre-processing

The process of cleaning and formatting raw data so that it is used in a machine learning model is referred to as "data pre-processing." This phase, which is both the first and most important step, kicks off the process of developing a model for machine learning, which is the process. We do not always come across data that is both clean and organized when we are utilizing machine learning to develop a project since it is not always the case. This is since it does not always hold true. In addition, before carrying out any action that includes data, it is required to thoroughly clean the data and arrange it in the right manner. This is a prerequisite for carrying out any activity that involves data. Before moving

on to any other steps that include data, this one must be completed first. As a direct result of this, we make use of the labor involved in data preparation for the purpose of achieving this goal.

Missing data management: Dealing with any datasets that are missing data is the task that will be performed at the subsequent step of the technique for data preparation. Our machine learning model is susceptible to experiencing a large level of difficulty as a direct consequence of the absence of any of the data included within our dataset. Because of this, it is of the utmost importance to address the missing values that are included within the dataset. The most common approaches of dealing with missing data include the following:

- Dealing with null values often involves removing the individual row, since this is the first option available. In this way, all that must be done to complete the task at hand is to delete the specific row or column that is composed of null data. Moreover, this strategy involves removing data.
- Using the method of calculating the mean: We will first identify the mean of the column or row in which there is a missing value, calculate that mean, and then use it in place of the missing value. This approach allows us to do all these things. This strategy is excellent for the components of the product that require quantitative data, such as age, income, year, and a host of other similar categories.

Encoding Categorical data: Categorical data is a kind of data that refers to information that is partitioned into separate categories. For instance, our dataset contains two categorical variables: the location of the item and whether it was bought. Since the machine learning model relies only on mathematical equations and numerical data, the construction of the model might be complicated by the presence of a categorical variable in our dataset. Consequently, it is essential to convert these category variables into numerical form.

Feature Scaling: The last step of the "data pre-processing" phase in machine learning is referred to as "feature scaling." In the process of scaling features as shown in Figure 2, all our variables are normalized such that they fall within the same range and are measured using the same scale. This guarantees that no one variable has more weight than the rest in the equation. If we do not scale the variable, then our machine learning model will have some kind of difficulty as a result. If we do not scale the model, which is based on the Euclidean distance between two points, then we will run into a difficulty. Machine learning models are built on this concept. If we calculate any two values using age and salary, then the values related to salary will take precedence over the age-related computations, which will lead to an inaccurate result. Therefore, we need to carry out feature scaling for machine learning to eliminate this problem. If we calculate any two values using age and salary, then the values related to salary will take precedence over the age-related computations, which will lead to an inaccurate result. Therefore, we need to carry out feature scaling for machine learning to eliminate this problem.

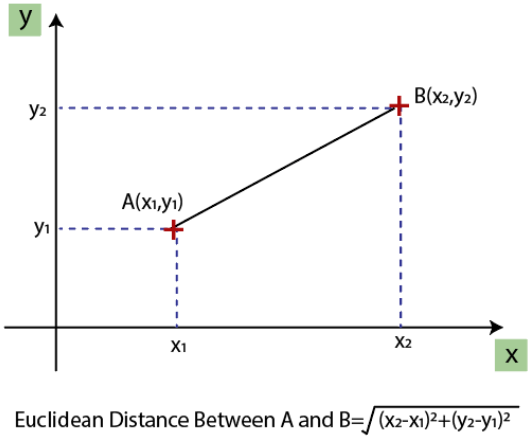

Fig. 2: Feature scaling.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 6, No. 6, June : 2023

3.4 TF-IDF Feature extraction

When it comes to information retrieval, one of the most important tactics that can be applied is to convey the relevance of a given word or phrase in connection to a certain document. This is done in several different ways. Consider the following example as a point of reference: We have what's known as a Bag of Words (BOW), and we need to parse the contents of it to get the information we need. In this scenario, we may implement the technique. On the other hand, this increase is usually counterbalanced by the frequency with which the phrase occurs in the corpus, which helps to explain for the fact that terms are used more frequently in general. However, this growth is proportionate to the number of times a word appears in the document. However, this growth is proportionate to the number of times a word appears in the document. The value of the TF-IDF increases in proportion to the number of times a word appears in the document; however, this growth is proportional to the number of times a word appears in the document.

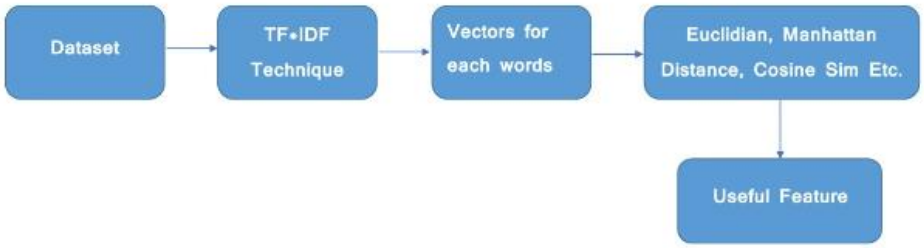

Fig. 3: TF-IDF block diagram.

The first method is known as the phrase frequency technique as shown in Figure 3. It is possible to calculate the term frequency of a piece of writing by comparing the total number of occurrences of a certain term within the document to the total number of words included within the document. This will give you an estimate of the term frequency. A percentage is used to represent the frequency of a term. The amount of information that a word provides is measured using the inverse document frequency, which also considers how often the term occurs in documents. It assesses how significant a particular word is in relation to the meaning of the whole piece of work. IDF demonstrates if a certain phrase is used often or seldom across all the publications. Calculating TF-IDF is as simple as multiplying TF by IDF. Raw data are not immediately transformed into usable features using TF-IDF in any way. To begin, it transforms raw strings or datasets into vectors, with a separate vector being assigned to every word. Then, to get the feature, we are going to put into practice a particular approach such as the Cosine Similarity algorithm, which works with vectors, and so on.

Term Frequency (TF): Let's say we have a collection of English text papers, and we want to determine which document is the most pertinent to the question "Data Science is awesome!" by ranking them in order of relevance. Eliminating papers from the search that do not include the three terms "Data is," "Science," and "awesome" is a straightforward method for getting started, but this will still leave a significant number of pages. To differentiate them even further, by keeping track of the number of occurrences of each word that we come across in each document, we can calculate the frequency with which they are used. The number of occurrences of a certain word inside the confines of a given text is referred to as that text's frequency, and the phrase frequency of a document refers to that number. The frequency with which a certain word or phrase occurs in a piece of writing is directly proportional to the importance that the phrase or word carries when it appears in the text.

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

Document Frequency: In this way, it is fairly like TF in that it evaluates the significance of each document within the context of the whole corpus. However, there is one major difference between the two. The only thing that differentiates the two is that DF is the count of occurrences of term t throughout all the documents in set N, while TF is a frequency counter for a word t in document d. This is the only thing that distinguishes the two. This is the sole difference between the two. To put it another way, DF is the number of documents that include the word in some capacity. If the word

appears in the text at least once, then we count it as one occurrence; however, it is not necessary for us to know how many times the phrase is present.

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

Inverse Document Frequency (IDF): When calculating TF, each phrase is given the same amount of weight as the others. On the other hand, it is common knowledge that some phrases, such as "is," "of," and "that," may occur rather often yet are of relatively little significance. Because of this, we need to give words that occur often in the document set less weight in Favor of terms that appear seldom and provide more weight to phrases that appear frequently. Calculating the inverse document frequency factor (IDF) is one way to achieve this goal. The IDF works by decreasing the amount of weight that is given to phrases that occur very often in the document collection and increasing the amount of weight that is given to terms that appear seldom. The informativeness of a term is quantified by looking at how often it appears in documents; the inverse of that number is referred to as the information document frequency, or IDF. IDF will assign a very low value to the words that appear the most often, such as stop words (this is because stop words such as "is" are present in practically all the texts, and N/df will assign a very low value to that word). When we do the calculation for IDF, the value assigned to the words that occur the most often, such as stop words, will be very low. This provides us with a weightage in relative terms, which is exactly what we have been searching for from the beginning.

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

If there is a huge corpus, say 100,000,000, the IDF value skyrockets; thus, to prevent the impact, we take the log of the idf. Now there are a few more issues with the IDF. When a word that is not in the vocab is encountered within the duration of the query, the df value will be 0. Since we are unable to do division by zero, we will round off the value by adding one to the denominator.

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

The TF-IDF has recently achieved the level of accuracy required to accurately evaluate the significance of a word in relation to a particular piece of writing that is included within a collection or corpus. When I first tried, I failed miserably. There are a wide variety of TF-IDF iterations illustrated in this image; nevertheless, for the time being, let's focus on this most fundamental iteration.

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

3.3 MLP Classifier

The Perceptron is now more often known as an algorithm; yet, when it was first developed, its primary function was that of an image recognition computer. It derives its name from the fact that it can perceive, see, and identify pictures in the same way that humans do. Particularly, there has been a lot of interest in the concept of a machine that would be capable of conceptualizing inputs impinging directly from the physical environment, such as light, sound, temperature, etc. The "phenomenal world" that we are all familiar with, without the need for the intervention of a human agent to digest and code the necessary information. This concept has garnered a lot of attention in recent years. There has been a great deal of interest in the idea of a device that would be capable to conceptualizing the inputs that would be impacting the outputs. This has been the primary focus of most of the research that has been conducted in this area. The neuron, a fundamental component of computational systems, was essential to Rosenblatt's perceptron machine. Each neuron, much as in the earlier models, consists of a cell that is provided with several inputs and weights in sequential order. The inputs are aggregated into a weighted total in Rosenblatt's model, and the neuron only fires and delivers an output if the combined weighted total is larger than a present threshold. This is the primary distinction between Rosenblatt's model and other models. Figure 4 shows threshold logic (right) and the perceptron neuron model (left). The activation function is denoted by the symbol T, which stands for the threshold. The neuron will create the value 1 as its output if the total of the weights of the inputs is higher than zero; otherwise, it will produce the value 0, while in all other cases, it will produce the value 0.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 6, No. 6, June : 2023

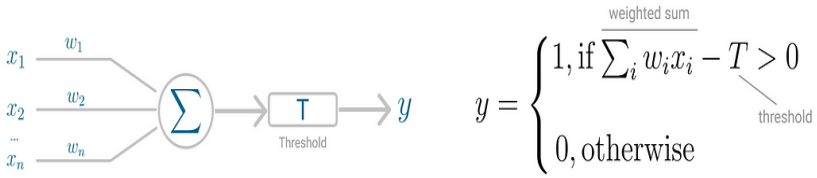

Fig. 4: Threshold logic (right) and the perceptron neuron model (left).

Binary Classification Using a Perceptron: The perceptron is employed as a binary classification model thanks to its discrete output, which is controlled by the activation function. This results in the production of a linear decision boundary. This is possible because of the perceptron's ability to construct a linear choice border. Because of the perceptron's capacity to construct a linear choice boundary, this is really feasible. This feat is made possible because of the perceptron's ability to establish a decision boundary. It does this by determining the separation hyperplane that reduces the distance between points that are unsure about their classification and the decision boundary, and it does so by finding that hyperplane. The following is a definition of the perceptron loss function:

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

misclassified observations

The optimization function known as stochastic gradient descent (SGD) is what the perceptron employs to cut down on the magnitude of this distance. If the data can be linearly isolated from one another, then it is certain that the SGD method will converge in a constrained number of steps. This is a certain assurance. The activation function, which is responsible for determining whether a neuron will fire, is the last component that the Perceptron requires to be fully functional. It is responsible for deciding whether a neuron will fire. When one looks at the function only from the perspective of its shape, it becomes clear why the sigmoid function was chosen to be included into the first Perceptron models. The sigmoid function is the mathematical representation of a non-linear function. It accepts any actual input and returns a number that, depending on the conditions, is either 0 or 1. Even if the neuron can interpret negative numbers when they are supplied as input, it is still able to create an output value of either 0 or 1, depending on which one it is told to do so.

Figure 5 shows the perceptron neuron model (left) and activation function (right). Here, Rectified Linear Unit (ReLU) as the activation function for the neuron. This is because the ReLU is a linear unit that has been rectified. The reason for this is since the ReLU is a linear activation function. Since the ReLU is a linear activation function, this is the result. It is scale-invariant, which means that its characteristics are not influenced in any way by the amount of the input, making it more efficient to compute, and it permits more optimization via the use of SGD. Because of these advantages, the usage of ReLU has increased, causing it to become more popular. The neuron takes in the inputs, and then it chooses a random starting weight set for the connections it makes. The value of the output is decided by the activation function, which is known as ReLU. This occurs after the inputs have been combined by means of a weighted sum addition. The SGD method is employed by the Perceptron so that it can find, or you might say learn, the combination of weights that results in the smallest distance feasible between the misclassified points and the decision border. This is accomplished via the use of the SGD methodology. This is accomplished via the utilization of the SGD method. This is accomplished by using the SGD algorithm. The dataset is divided along a linear hyperplane as soon as the SGD algorithm reaches a point of convergence, and the two halves are then compared. Although it was claimed that the Perceptron could represent any logic or circuit, the most significant criticism directed at it was that it was unable to represent the XOR gate, which is also known as the exclusive OR gate and is a gate that only returns 1 if the inputs are unique. This was the gate that was the most important failure of the Perceptron. This was shown around a decade later, and it highlights the fact that the

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 6, No. 6, June : 2023

Perceptron model, which is comprised of just a single neuron, is unable to be applied to non-linear data sets.

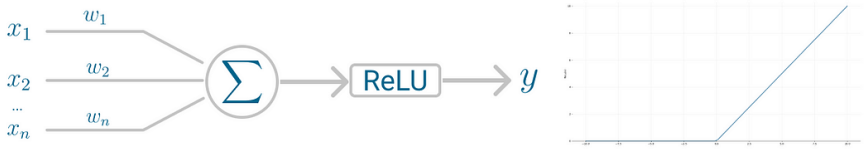

Fig. 5: Perceptron neuron model (left) and activation function (right).

The MLP was developed in response to this limitation, which served as the drive for its development as shown in Figure 6. It is a neural network in which there is no linear development between the mapping of inputs and outputs. A multilayer perceptron, commonly referred to as an MLP, is constructed of not just layers for input and output, but also one or more hidden layers. Each of these hidden layers is made up of several neurons that are placed one on top of the other. And even though every neuron in a perceptron is required to have an activation function that enforces a threshold, such as a ReLU or sigmoid, neurons in a multilayer perceptron are permitted to make use of any activation function of their choosing. In contrast to this, the scenario in a perceptron requires each neuron to have an activation function that maintains a threshold. In a perceptron, each neuron must meet this requirement.

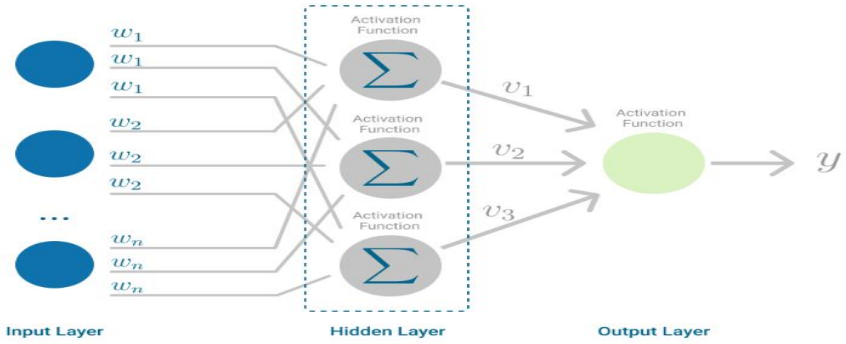

Fig. 6: Architecture of MLP.

MLP is believed to be a feedforward algorithm since the inputs are added to the initial weights in a weighted sum, and then the result is applied to the activation function. This process is identical to the one used in the Perceptron. This is because the Perceptron came before the MLP, which explains why this is the case. However, a significant difference is that each linear combination is passed on to the succeeding layer. This is one of the fundamental differences. Each layer transmits to the next layer both the results of its own computation and its own individual interpretation of the data. These are both handed on by each layer in turn. This reaches all the way up to the output layer and travels through each of the hidden layers on its way there. If the algorithm were to simply calculate the weighted sums in each neuron, send its findings to the output layer, and then stop, it would not be able to learn the weights that minimize the cost function because it would not be able to select which weights are ideal. Therefore, it would not be able to learn the weights that minimize the cost function. This would prevent it from being able to learn the weights that would result in the cost function being reduced to its lowest possible value. There would be no true learning taking place if the algorithm were just given a single opportunity to iterate. Backpropagation is something that kicks into gear at this point in the process and continues throughout.

Backpropagation is the name of the learning process that is carried out by the MLP as shown in Figure 7, and this strategy is what enables the MLP to repeatedly adjust the weights that are already present in the network with the goal of bringing the cost function down to a lower value. There is a single, very tight need that must be satisfied before backpropagation can operate as it should. Differentiability is required for both the threshold function and the function of a neuron that combines inputs and weights, such as the weighted sum and the ReLU functions, respectively. In order for these functions

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 6, No. 6, June : 2023

to be considered valid, they need to have a finite derivative. This is because gradient descent is the optimization function that is used in MLP most of the time.

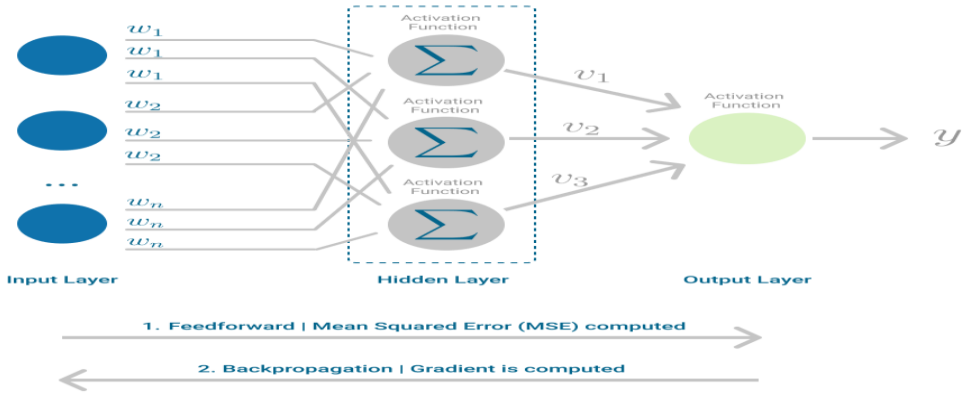

Fig. 7: MLP, highlighting the feedforward and backpropagation steps.

After the weighted sums have been spread throughout all the layers in each iteration, the gradient of the Mean Squared Error is calculated across all the input and output pairs. This step is repeated until the desired level of accuracy is achieved. This occurs after the iteration is complete. This is done to reduce the likelihood of making a mistake. Following this step, the value of the gradient is then added to the weights of the first hidden layer. This is done so that it is propagated backwards. This is the process through which the weights are sent all the way back to the beginning of the neural network. This procedure will continue until the gradient for each input-output pair has converged, which occurs when the newly calculated gradient hasn't changed by more than a convergence threshold since the previous time it was iterated. Convergence is defined as the situation in which the newly computed gradient hasn't changed by more than the threshold since the last time it was iterated.

4. Results and discussion

This section gives the detailed analysis of results and discussion, which includes dataset description and performance evaluation.

4.1 DRUGREVIEW Dataset

The information contains not only patient testimonials about medications and the illnesses to which they are associated, but also a patient rating out of ten stars that denotes the level of overall patient approval. The information was gathered by searching through several pharmaceutical review websites that are available on the internet. The goal was to do research:

- Sentiment analysis of drug experience across multiple facets, which includes sentiments learned on aspects like effectiveness and side effects.
- The transferability of models among domains, which includes conditions; and
- The transferability of models across different data sources (see 'Drug Review Dataset (Druglib.com)').

A train partition that contains 75% of the data and a test partition that contains 25% of the data are constructed (see publication) and stored in two separate tab-separated values (.tsv) files, respectively. Both partitions are saved separately. You understand and agree that you are solely responsible for your actions when using this dataset.

- drugName (a categorical term): the name of the drug
- condition (a categorical term): the name of the ailment
- review (a textual term): a patient review
- rating (a numeric term): a patient rating of 10 stars
- date (date): the date on which the review was first submitted.
- usefulCount (numerical): the total number of people who considered the review to be helpful.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 6, No. 6, June : 2023

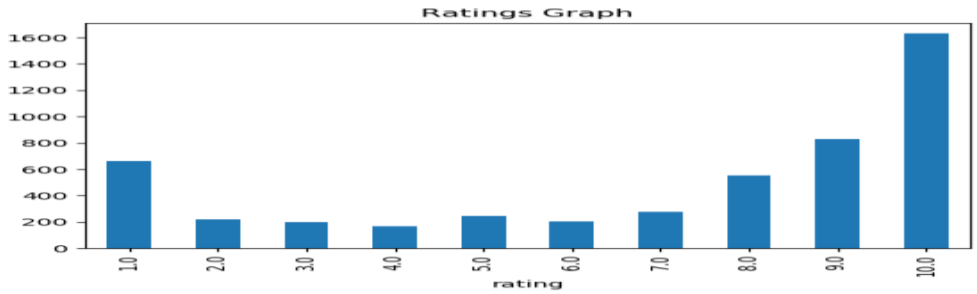

Fig. 8: Drugs ratings graph.

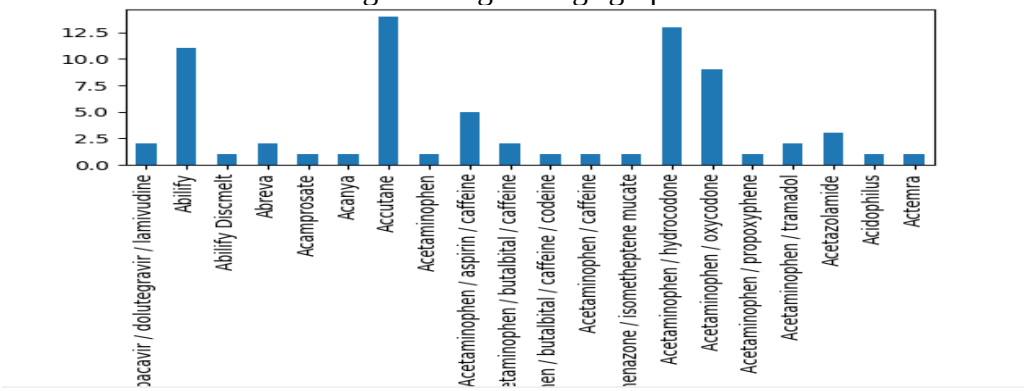

Fig. 9: Drug names dataset.

Figure 8 shows the dataset description. On the graph, the ratings are shown along the x-axis, and the total number of records that got that rating is displayed along the y-axis. Figure 9 shows the drug names of dataset. It is clear from the screenshot that all stop words and other symbols have been eliminated from the reviews, and the graph now just displays the top 20 drugs that were found in the dataset. The x-axis of the graph above displays the name of the drug, while the y-axis displays the count of how many times it was used. Table 1 compares the performance evaluation of various drug recommendation systems. Here, proposed MLP resulted in improved performance as compared to other models like Existing Logistic regression, Existing SVC, Existing Ridge classifier, Existing Multimodal naive bayes, and Existing SGDC. Figure 10 shows the performance comparison graph of Table 1. Figure 11 shows the drug recommendations from test data.

Table. 1: Performance evaluation.

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 naive 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

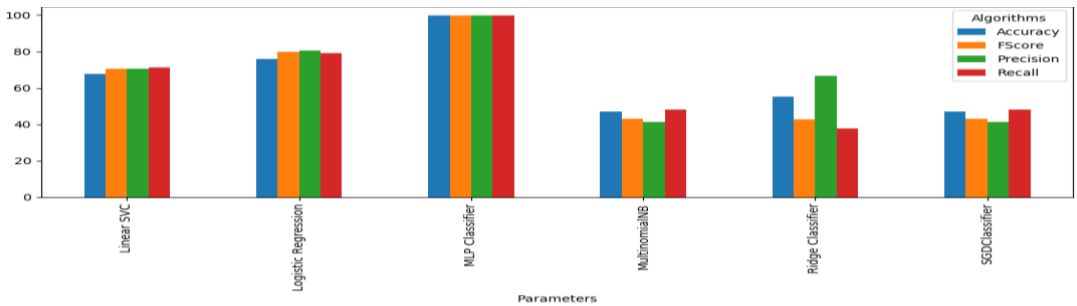

Fig. 10: Performance comparison graph.

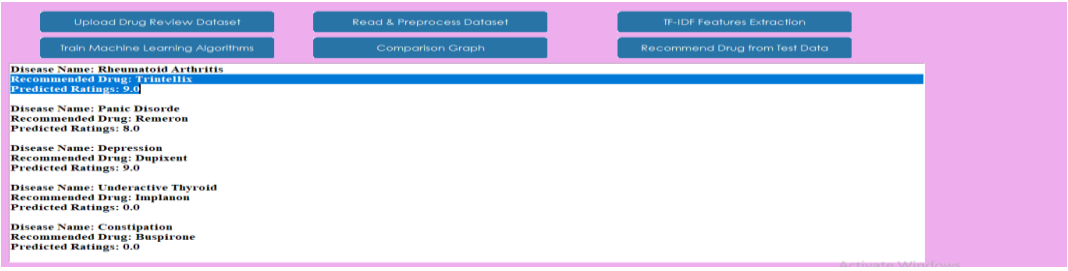

Fig. 11: Drug recommendations from test data.

5. Conclusion

In conclusion, this work proposed a MLP model for a drug recommendation system based on sentiment analysis of drug reviews. Our methodology involves dataset preprocessing, TF-IDF feature extraction, and training an MLP classifier to classify drug reviews into sentiment categories. Through dataset preprocessing, we ensured the quality of the data by performing text cleaning operations and handling challenges like misspellings and abbreviations. The TF-IDF technique allowed us to extract important features from the drug reviews by assigning weights based on term frequency and rarity. These TF-IDF values served as input features for training our MLP classifier. The MLP classifier, with its multiple layers of interconnected neurons, was capable of capturing complex relationships between the input features and sentiment labels. We optimized the model's parameters using backpropagation and gradient descent during the training process. To evaluate the performance of our drug recommendation system, we conducted experiments on a separate test dataset of drug reviews. The system's effectiveness was assessed using standard evaluation metrics such as accuracy, precision, recall, and F1-score. However, it is important to note that further research and refinement are necessary to improve the accuracy and robustness of the MLP model. Additionally, incorporating additional factors such as user demographics and medical history could enhance the precision of the drug recommendation system.

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.

[16] J. Li, H. Xu, X. He, J. Deng and X. Sun, "Tweet modeling with LSTM recurrent neural networks for hashtag recommendation", 2016 International Joint Conference on Neural Networks (IJCNN), 2016, pp. 1570-1577, doi: 10.1109/IJCNN.2016.7727385.

[17] I. Korkontzelos, A. Nikfarjam, M. Shardlow, A. Sarker, S. Ananiadou, G. Gonzalez. "Analysis of the effect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts", Journal of Biomedical Informatics, vol. 62, 2016, pp. 148-158, no. 1532-0464, https://doi.org/10.1016/j.jbi.2016.06.007.

[18] V. Gopalakrishnan and C. Ramaswamy. "Patient opinion mining to analyze drugs satisfaction using supervised learning, Journal of Applied Research and Technology", vol. 15, issue 4, 2017, pp. 311-319, no. 1665-6423, https://doi.org/10.1016/j.jart.2017.02.005.

[19] S. Alireza, K. Fatemeh and R. Nasrin. "Preventing the medication errors in hospitals: A qualitative study", International Journal of Africa Nursing Sciences, vol. 13, 2020, 100235, no. 2214-1391, https://doi.org/10.1016/j.ijans.2020.100235.

[20] G. Gurdin, J. A. Vargas, L. G. Maffey, A. L. Olex , N. A. Lewinski, S. T. McInnes. "Analysis of Inter-Domain and Cross-Domain Drug Review Polarity Classification". AMIA Jt Summits Transl Sci Proc. 2020 May 30;2020:201-210. PMID: 32477639; PMCID: PMC7233089.