Drug Prediction Based on Customer Reviews

Tina Torabinejad
Department of Computer Science
California State University, Fullerton
Fullerton, United States
ttorabinejad@csu.fullerton.edu

Roya Zeinali
Department of Computer Science
California State University, Fullerton
Fullerton, United States
zeinali.roya@csu.fullerton.edu

Kanika Sood
Department of Computer Science
California State University, Fullerton
Fullerton, United States
kasood@fullerton.edu

***Abstract*—All humans are prone to experience sickness at least once in their lifetime. Regardless of how sick humans get, there is a need to take medication most of the time. Often, doctors choose the appropriate medication from available drugs depending on the patient's situation and the side effects of the medication. Given that we find 215,063 data points in which customers rate their satisfaction with various medications, we create two different models and use various machine learning techniques to determine drugs with fewer side effects and higher satisfaction rates among consumers. We want this method to generate a smoother relationship between physicians, patients, and medication options. We do not intend to generate a model that recommends medication to patients. Instead, we hope to create a tool that doctors can use to choose the best available option for their patients based on other patients' previous experiences with the particular medication. This paper examines different preprocessing techniques, such as sentiment analysis and feature encoding, to clean and prepare data for applying different machine learning techniques. In addition, more than eight distinct machine learning techniques are used to train and test our data. In this paper, two different groups of conditions are used. One group focuses on a single condition, and the other examines the nine most distributed conditions. Then, we compare two models and realize that the model, which is focused on a single condition, gives us more accurate results than the nine most distributed conditions.**

***Keywords*— classification, classification report, data preprocessing, drug choice suggestion, machine learning**

I. Introduction

Medicine is essential in enhancing health and quality of life. It is significant for doctors and patients to suggest and use medication with fewer side effects and faster disease resolution. The safety and effectiveness of pharmaceutical products are determined using clinical trials [4]. This method is effective but requires standardized settings that the Department of Health Care and many other government organizations must approve. In addition, it requires a specific testing protocol, given that most drugs are tested on animals and humans. Also, it is time-consuming and requires many resources. As a result, some methods, such as data-driven Clinical Decision Support Systems (CDSS), can help with clinical decision-making tasks [5]. CDSS can help health professionals in many aspects, including diagnosis, treatment, and side effects of medications [5]. Machine learning techniques can be of great use in this case as they can help combine large amounts of data gathered from clinical trials and give healthcare professionals an overall view of how effective a treatment method or medication can be for patients. Technology can help improve decisions being made regarding medication; however, some challenges still need to be resolved gradually to be able to use the data for making predictions. Even though available machine learning techniques help predict different things based on big data, some critical factors must be considered.

For instance, data preprocessing and cleaning may be challenging. Many problems can be predicted and solved, but figuring out what might be wrong can be time-consuming and expensive. In addition, many features can affect the prediction process [1]. If features are irrelevant to the target value (the event that scientists aim to predict), or if there are few relevant features, the accuracy of the model would decrease, which means that the model is not representing the real-world problem, and that is what machine learning scientists want to ignore. Lastly, some frameworks can be used to make the prediction easier [2].

Nowadays, technology plays a crucial role in collecting data from customers. Recent studies suggest that younger generations prefer to express their opinion and thoughts regarding medication and clinician-to-patient communication via social media [6]. Social media and websites such as Drugs.com, DrugsLib.com, WebMB.com, etc., are famous for patients to share their concerns, experiences, and drug side effects with others. Recent research has shown that patients' opinions are valuable and essential for medical professionals and pharmaceutical companies -especially if they want to understand better their customer's thoughts about a particular medication [3].

New drugs are developed daily as researchers learn more about the human body. As a result, there are more options to choose from when picking the proper medication. As everyone knows, given the chemical composition of a particular drug, some individuals' bodies may be more sensitive to a particular medication than others. On the other hand, there is more than one brand for a particular drug. Often, having more than one brand to choose from makes it difficult for physicians to prescribe the medication and uncomfortable for patients to choose from available options. For example, anyone with a common cold who wants to buy conjunction relief medication from any pharmacy may have experienced the problem of dealing with more than three brands of medications on the store shelf. Frequently, whichever brand that person chooses, there is always a guilty feeling that the other brand might be more effective than the one he or she is using. This paper's primary purpose is to analyze drug review data and use different machine learning techniques to identify which drug is more effective and popular than the others. This prediction is based on customers' reviews, conditions,

979-8-3503-4110-2/23/$31.00 ©2023 IEEE

ratings, dates, and how other customers find the comments and ratings applicable based on their experience. Different techniques, such as sentiment analysis of review, have been used to make such a model, which will be discussed later in this paper.

The primary motivation of this paper is to create a bridge between medicine and customer satisfaction. It is important to note that most of the medications are prescribed by doctors, and this paper is not proposing any suggestion regarding which medication is better for patients; however, when it comes to different pharmaceutical companies (a.k.a. brand) that generate the medication, this paper can be helpful. For instance, more than twelve different brands provide birth control medication. The purpose of these medications is the same, and they are all prescribed by doctors, but as our data suggest, patients may experience some side effects from one brand compared to another, and this is where our approach can be helpful. We use patients' reviews on particular drugs to suggest to other customers which medication is better depending on their conditions.

II. BACKGROUND AND RELATED WORK

Literature on drug reviews and pharmacovigilance can be divided into studies on identifying aspects such as drug reviews, side effects, and dealing with overall or aspect-based sentiment analysis [4]. Most approaches regarding the analysis of drug reviews are lexical-based, based on mapping modifications and phrases associated with user data to specific words from different individuals or combinations of terminology [4], [5]. However, these approaches are accompanied by spelling or typographical errors, and it needs help to overcome these limitations; machine learning has done its best to overcome these limitations. Most available research uses natural language processing techniques to predict drugs' names [4]. Other researchers use patient reviews and patients' conditions to predict the best suitable therapy available for patients [5].

On the other hand, other studies focus on how different machine learning techniques can be used to discover new drugs [7]. At the same time, they focus on some limitations that may involve using machine learning for predicting features. Other research focuses on how machine learning became a popular tool in the healthcare industry for predicting different aspects of diseases or drugs, using patients' feedback or any other related data regarding patients [8]. Recently, most researchers have been focusing on using machine learning based approaches for categorizing drugs based on their chemical components or discovering new drugs based on combining two existing drugs [9]. Recent research is focused on two different aspects when it comes to using machine learning to predict an event related to drugs. Two categories are the prominent hot spots for most machine learning scientists focusing on using various techniques on pharmaceutical base products. Either they intend to rely heavily on patients' reviews for predicting popular drugs [4], [5], or they are using chemical components of different drugs to predict new drugs and then use the result for clinical trials to check the side effects of newly developed drugs [9].

This paper uses a combination of reviews to predict drug names. This includes patients' opinions about the particular drug and other factors such as patient condition and usefulness of comments. This approach has not been examined before for drug name prediction. As a result, we are motivated to test it out and see how practical this approach will be.

III. DATASET AND DATA PREPROCESSSING

We use the dataset that is available online and is open-source from the University of California Irvine machine learning archive website [4]. It contains 215,063 data points related to participants' reviews of a particular drug they use based on the diseases they have. According to the collected data, we try to categorize them for the specific condition and, with the help of various machine learning techniques, predict the popular drug names out of all the brands we have in our data. Our data consists of both numerical and categorical values. To use the University of California Irvine data, we change the target value that they have and focus on predicting the drug name based on the following features [4]:

- **Condition:** It is categorical data of an individual's conditions (illness).
- **Review:** This section includes customers' reviews re- garding specific drugs.
- **Date:** It contains the day, month, and year of when the review was written
- **Rating:** It consists of a numerical number on a scale of 0-10 where each individual rated the specific drug.
- **Useful Count:** It relates to the number of times other customers find the reviews helpful.

Our data is extensive and contains 115 different conditions. After loading the data and taking a closer look at it, we realize that we are dealing with more than ten classes of drugs, our target label. Our dataset has 8,245 data points that are related to pain medication, while we have only one medication for Wilson's disease. This shows we are dealing with an imbalanced dataset, which can result in poor training for any machine learning techniques we plan to use. If we want to keep our data as it is and train them based on what we have, the training process is unfair since it gets trained better for pain medications than Wilson's disease. Therefore, to avoid such an issue, we separate our data, train two distinct models, and then compare our results.

We know that machine learning techniques perform best when the data is clean and focused [10]. Hence, we need to narrow down the problem we want to solve to get more accuracy and better represent the real-world problem. Since the condition is the most related feature to our target value (drug name), we agree to separate two models based on the condition column. By taking a closer look at our data, we notice that the highest number of data points are related to birth control medications. In total, we have 38,436 data points, which are high enough to be trained and tested separately. Indeed, our first model only focuses on a birth control condition, and all the drugs are related to this condition. Removing birth control from our original data leaves us with

176,627 data points for more than 100 conditions.

Although it is a great idea to use extensive data for training the model using machine learning techniques [11], because the variation of our model is still high, we decide to reduce our model to conditions that are repeated more than 4,000 times in our dataset, which reduces our data to 60,287 data points. This decision has reduced the amount of our original data significantly; however, we prefer to work with less data but be able to represent reality better. In addition, separating data points based on conditions has two benefits for us. First, we deal with two models focusing on the same target value, but one has more specific data than the other. Second, we perform two preprocessing and machine learning technique selections. Given the differences between the two models, these reasons give us a better understanding of how our prediction performs. Below, we explain the preprocessing that we perform for each model separately.

A. Top Nine Most Frequent Conditions Model

By reducing the number of conditions, we have nine different conditions in our data, so there is no need to add or remove anything from the condition column. As a result, we start our preprocessing by looking at the number of unique values in the drug name column. We notice 666 different drug names in our data, some of which only happen to repeat very few times or even appear in our data once. Therefore, we remove drugs repeated less than 400 times to avoid data imbalance and related issues. After all of the reductions, we are left with 31,018 data points. Next, we split our data into a training and testing set. Since we have a large amount of data, we dedicate 20% of our data to the testing set and the rest to the training set. Because conditions play a vital role in the model, we use the Stratified Shuffle Split to distribute different conditions evenly through the training and testing sets.

We visualize our data before preprocessing to understand better how our data looks so that we can apply various preprocessing techniques accordingly. Since the top nine most frequent conditions contain categorical values such as conditions, only numerical values such as ”Useful Count,” ”Rating,” and ”Column 1”, which contains participants’ ID number, is visible. By taking a closer look at the figure we obtained, it is evident that the top nine most frequent conditions model contains many outliers and has imbalance issues, particularly for rating and useful count. Data seems mostly left-skewed for rating and right-skewed for the useful count. We must deal with these features before training the model in our preprocessing. Nevertheless, before dealing with these two features, we need to handle categorical data to visualize our model better and deal with ratings and useful counts that already contain numerical values. Because our data contains many categorical variables, it is difficult for machine learning algorithms to predict the correct output. As a result, the first thing that we need to do is handle participants’ reviews.

This work includes features such as ratings, useful counts, and reviews. One of the most challenging features to process is reviews.

This feature contains written sentences where participants express their thoughts regarding each drug that they use. In this section, patients use different words to express their opinions and share their experiences with any side effects when using a particular medication. Hence, this section includes both positive and negative words or phrases. To use these reviews, we need to apply the sentiment analysis technique to preprocess reviews and detect the polarity of each review. Sentiment Analysis is considered part of Natural Language Processing. This technique is used to detect sentence tone and convert long sentences of reviews into positive, negative, or neutral tones [12]. Few popular libraries are available to perform such a task, and for our first choice, we use the NLTK library to extract the tone of each review. Using this library, we analyze the sentiment of drug reviews based on the overall expression and then classify the sentiments. Finally, we categorize reviews into three categories: positive, negative, or neutral. In addition, we want to check for any relationship between ratings and reviews by looking at the relationship between compound score (the score we get from each review after using the NLTK library for detecting the tone of each review) and rating.

As Fig. 1 shows, a positive relationship exists between ratings and reviews, which means that the compound score increases as the number of ratings increases. This library is handy for our purpose; however, when this library is used, most of our reviews are considered neutral, which in our case is not going to help the model for making the prediction; therefore, we decide to check another popular library to see if we can get a more accurate result. For our second try, we use the Transformers library. Using this library, we obtain more precise results than the NLTK library.

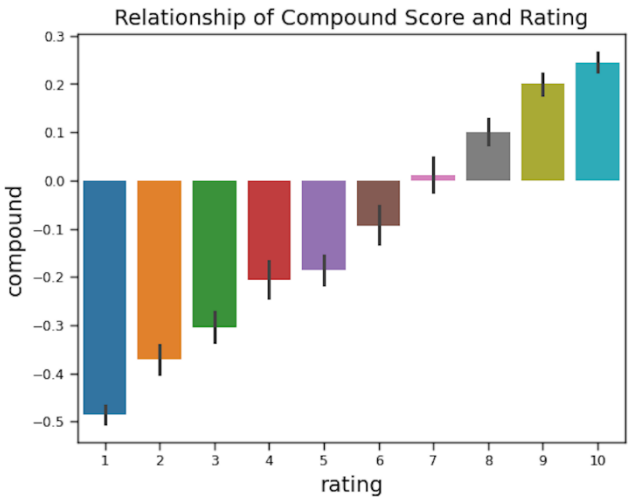

Fig. 1. The relationship between compound scores obtained from NLTK library and the rating.

Using the NLTK library, we obtain the following result: 24,555 neutrals, 159 positives, and 100 negatives reviews out of 24,814 available reviews in the training set. Meanwhile, when we use the Transformer library, 15,466 data points are considered positive, and 9,348 reviews as negative. The analysis of reviews is more evenly distributed between

positive and negative when the Transformer library is used compared to the NLTK library. Therefore, we decide to keep the result that we achieve from the Transformer library instead of the NLTK. Finally, we convert positive and negative reviews into binary values of one and zero to make it easier for the machine learning algorithm. As a result, we have 15,466 ones (refer to positive review) and 9,348 zeros (refer to negative review).

Another feature that we need to take care of is the date. Initially, this column contains the month, day, and year of the time that reviews are collected from participants. We create two different features from the date column. We perform this action by distinguishing between year and month. Doing that helps us increase the number of related features we have for our target value. In addition, it helps us better understand whether or not the date can be valuable for our prediction. In other words, we want to check and make sure that date is a relevant feature and can be helpful for our prediction or not. Creating separate columns for year and month helps us count the number of entities for each month and year; both have a good amount of data, so we keep both features. On the other hand, the day can not be helpful for our model, so we keep month and year as two separate features and remove the date feature.

So far, for our preprocessing, we have finished preprocessing for reviews and dates; as a result, that can easily be used for different machine learning techniques we want to use. The rating is yet another feature that needs to be handled. Even though this feature already contains numerical values, it contains numbers between 0 to 10, a wide range that can affect training. After evaluating the number of unique values in the rating column and calculating the mean and standard deviation, we normalize this column by converting the values into either zero or one. After we check the data distribution, the average rating score is 7.5; indeed, any rate equal to or less than eight is considered ”negative,” and any rate greater than eight is considered ”positive.” Finally, we transfer ”positive” to one, which has 13,560 data points, and ”negative” to zero, which has 11,254 values. We perform the same procedure for the useful count and convert it to either zero or one. To prevent data imbalance issues, we sort the useful count from high to low and consider the first 23 most frequent values as one and the rest as zero.

Finally, we have two columns, condition and drug name. Both columns contain categorical data, the name of the disease, and the drug. We convert these two values into numerical data before training the model using machine learning algorithms. There are a couple of libraries that can do this. Therefore, we use OneHotEncoder to transfer categorical data to numerical values. After preprocessing all data and performing feature encoding, we recheck the data distribution to ensure that the data is evenly distributed among the whole data. Indeed, we can obtain more evenly distributed data. In addition, we convert all the categorical data to numerical values and lower the useful count and rating scale to only one and zero. By comparing the data before and after preprocessing, we see that the preprocessing of the top nine most frequent conditions model is finished, and the data is ready to be trained by different machine learning algorithms. It is important to note that in the end, once we are satisfied with the preprocessing result, the same procedures are performed on our testing set.

B. Birth Control Model

The number of features is almost similar to the other model because the birth control and the top nine most frequent conditions models are from the same dataset, except for the condition column. We start our preprocessing by looking at the number of unique values in the drug name column and notice 181 different drug names in our data, some of which only happen to repeat very few times or even once. Hence, we remove drugs that have a frequency of fewer than 400 times to avoid data imbalance and related issues. This leaves us with 29,793 data points. Before moving further with our preprocessing, we split our data into a training and testing set. Since we have a large amount of data, we dedicate 20% of our data to the testing set and the rest to the training set. The birth control model contains many outliers and has imbalance issues, particularly for useful counts. For instance, data seems mostly left-skewed for useful counts. We must deal with these two features before training the model in our preprocessing. Nevertheless, before dealing with them, we need to handle categorical data to visualize our model better and deal with ratings and useful counts that already contain numerical values. Our data contains many categorical data, which makes it difficult for machine learning algorithms to predict the correct output. As a result, the first thing that we do is handle participants’ reviews.

Like the top nine most frequent conditions model, we use two libraries (the NLTK and the Transformers library) to perform sentiment analysis on each review we have for the birth control medications. The reason for using two libraries is similar to the other model. We obtain more accurate results when using the Transformer library than the NLTK. Like the other model, most reviews are considered neutral when we use the NLTK library. Next, we use the Transformer library and obtain more accurate results; therefore, the analysis of reviews is more evenly distributed between positive and negative reviews when the Transformer library is used compare to the NLTK library. As a result, we decide to use the Transformer library instead of the NLTK. Finally, to make it easier for the machine learning algorithm, we convert “positive” reviews to one and “negative” reviews to zero.

Ultimately, we convert reviews to 17,136 (refer to positive reviews) and 6,698 zeros (refer to negative reviews). Like the top nine most frequent conditions model, we create two different features from the date. We do that by distinguishing between year and month. Creating separate columns for year and month helps us count the entities for each month and year. Both have a reasonable amount of data, so we keep both features.

On the other hand, the day is not helpful for our model, so we keep month and year as two separate features and remove the date feature. In contrast, after evaluating the

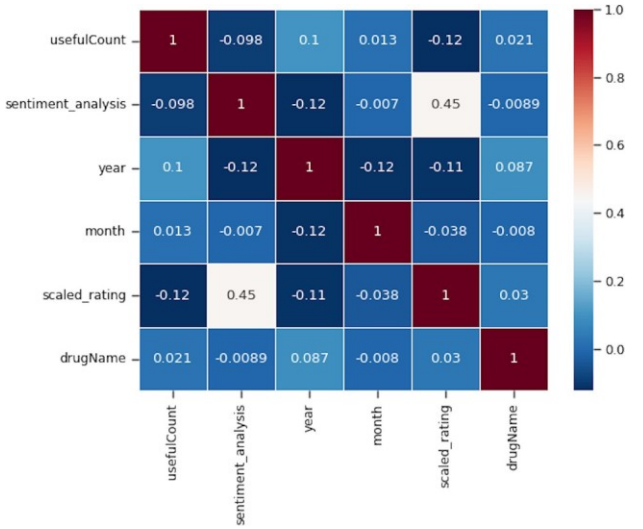

Fig. 2. Pearson correlation for birth control model

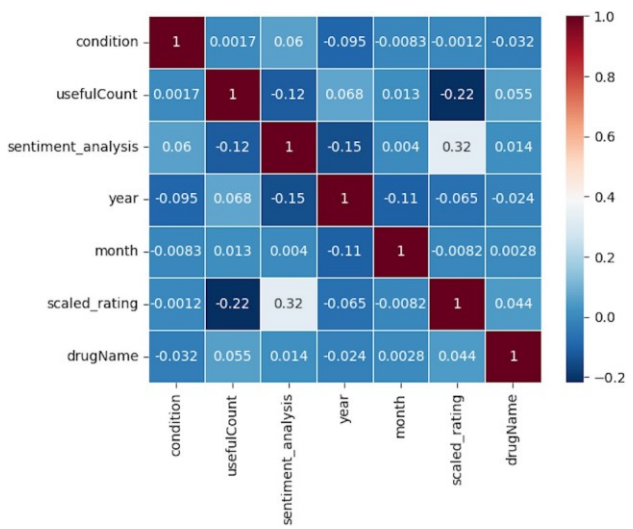

Fig. 3. Pearson correlation for the top nine most frequent conditions model

number of unique values and calculating the mean and standard deviation for the rating column, we normalize this column by converting them into either zero or one. After we check the data distribution, we notice the average rating score was 6.1. Any rate equal to or less than six is considered ”negative,” and any rate greater than six is considered positive. Finally, we transfer ”positive” to one, which has 12,694 data points, and ”negative” to zero, which has 11,140. We perform the same procedure for the useful count and convert it to zero or one. To prevent data imbalance issues, we sort the useful count from high to low and consider the first five most frequent values as one and the rest as zero. Finally, the drug name column contains categorical data, which is the name of the drugs. We convert these values into numerical data before training the model using machine learning algorithms. Since the drug name is our target value, we use LabelEncoder to convert this column to numerical values. Indeed, we obtain data that is more evenly distributed. In addition, we convert all the categorical data to numerical values and lower the useful count and rating scale to one and zero. We achieve balanced data by comparing the data before and after the birth control model preprocessing. As a result, the preprocessing of this model is finished. Once we complete the preprocessing of the training set, we perform the same procedures on the testing data as well.

IV. METHODOLOGY

In this study, we try different algorithms and compare them to see which gives us a better result. This section explains the features we select based on correlations and machine learning techniques we use for our dataset.

A. Feature Selection

Given the type of features and the amount of preprocessing we need to perform on all of them, we choose a backward selection for selecting our features. We start with the dataset’s complete feature set and study each to see if we can remove them. Also, we have to make new features based on existing ones. For example, by studying the date, we split this column into year and month and remove the existing date feature. We calculate the Pearson correlation to better decide the relationship between our features for both models. After checking the correlation between all features, we do not see a feature with very high or very low correlations in both models. Given that we originally started with not too many features, we decide to keep all of them and use them for making predictions. Fig. 2 and Fig. 3 show the correlation between features for birth control and the top nine most frequent conditions models. As both figures show, there are no significant strong or weak relationships between features.

B. Classification

For our two models, we pick almost the same techniques. We choose machine learning techniques that best suit our models:

- **K-Nearest Neighbor (KNN):** Given that we try to solve a classification problem and deal with more than one classifier, KNN is an excellent option since it can calculate the nearest neighbor around the target value. KNN is considered instance base learning, meaning it does not require learning until it reaches the prediction time.
- **Decision Tree:** It is a simple algorithm. It always chooses the best features; therefore, it is an excellent option for us to use and later compare with other algorithms.
- **Support Vector Machine (SVM):** SVM is not the best option for non-binary classification problems; however, since it is an idol option for data with many attributes, we use this model to see if it gives us acceptable accuracy. To be able to use this technique, we adjust the default parameter to one vs. one (OVO) since we have more than two classifiers.
- **Voting Classifiers:** We try hard and soft voting to compare results and see which performs best. We choose Random Forest since it allows replacement. Then, we fit the model using hard and soft voting for this technique. This technique benefits both models since our data contains more than nine classes.

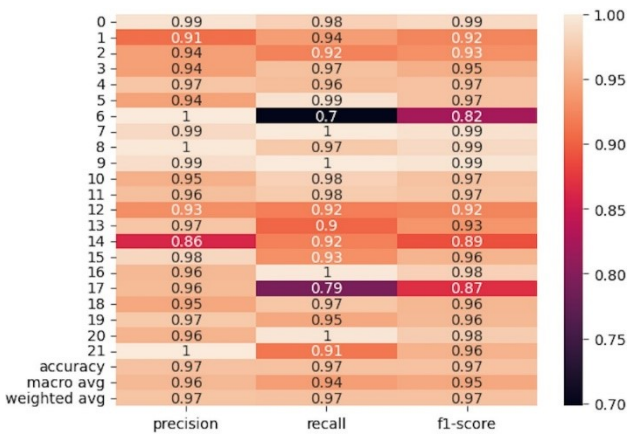

Fig. 4. Classification Report for the KNN - birth control model

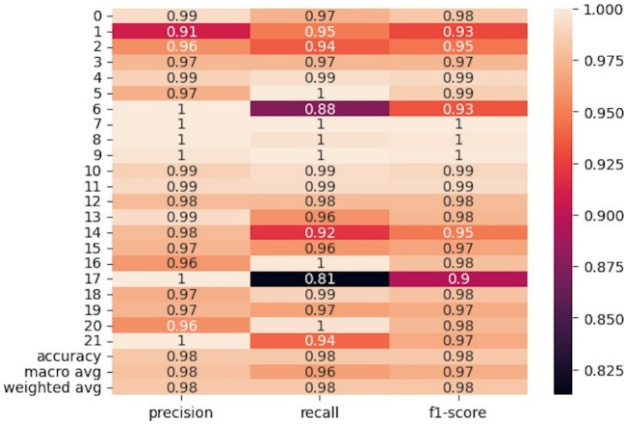

Fig. 5. Classification Report for weighted KNN - birth control model

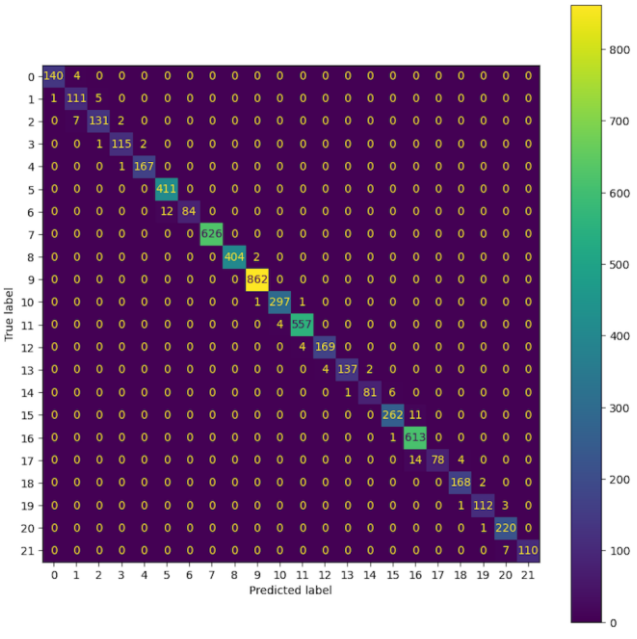

Fig. 6. Confusion Matrix for weighted KNN - birth control model

- **Logistic Regression:** The problem we are trying to solve in both models is not considered a regression problem; however, similar to SVM, this technique does offer an option of all vs. all, which trains the model based on the most critical classifier. Due to this option, we decide to try this technique on our models.
- **Boosting Techniques:** We mainly use Gradient Boost and Adaboost techniques. These techniques are considered part of the prediction group, known as ensemble learning. Since we have more than nine classifiers, the boosting technique benefits our model. The boosting technique acts sequentially, meaning it uses weight to handle miss prediction values which are very handful for both of our models.
- **Weighted KNN:** This technique is only used for the birth control model, which contains fewer classifiers. We try weighted KNN to compare the result with KNN.

V. EXPERIMENTS

In this section, we describe our results and analyze the achieved result. Condition is a very important attribute for predicting drug names, so we want to ensure that the training and testing set is representative of the various conditions; hence, we use StratifiedShuffleSplit to distribute data evenly among the testing and training sets. We randomly split the 31,018 data points for the top nine most frequent conditions model into training sets containing 24,814 data points, and testing sets with 6,204 data points. For the birth control model, we split the total of 29,793 data points randomly into a training set that contains 23,834 data points and a testing set that contains 5,959 data points. We try to use the same algorithm and method for both data sets since we want to compare machine learning approaches on two data sets and see which of these models gives us a better result. We implement the proposed approach in Python programming language. We use the Pandas library and other popular libraries to achieve our goal. We are using eight different techniques for our top nine most frequent conditions model, and for the birth control model, we are using the same techniques plus weighted KNN. We use a confusion matrix for each technique and check the accuracy, f1-marco, f1-micro, ROC curve, and loss function. For calculating the loss function, cross-entropy is used to examine the loss function value for all of our techniques. In addition, we use cross-validation to see how our models behave when data is shuffled and the model deals with new data sets.

A. Analysis of Birth Control Model

For this model, we gain different results for each technique. KNN and weighted KNN algorithms work very well for the birth control model since we obtain a high accuracy rate, which means it fits our model well. As precision or recall decreases, the f1 score decreases too; therefore, having higher precision and recall scores is better. As Fig. 4 and Fig. 5 show, the f1-micro, f1-macro, and accuracy are over 90% for KNN and weighted KNN. The accuracy of weighted KNN is slightly higher than KNN. Also, the loss function for both models is low (KNN=16% and weighted KNN=6%).

The lower value for this particular loss function means a lower level of uncertainty, which is what we are looking for. Even though both KNN and weighted KNN have low loss function values, the weighted KNN with cross-entropy of 6% is a better result than KNN. The accuracy of weighted KNN is more than KNN, and the loss function of it is less than KNN. So weighted KNN works better for the birth control model. In addition, Fig. 6 represents the confusion matrix of weighted KNN. As the figure suggests, each class has a high number of true positive values. Meaning that this technique provides reasonable training for the birth control model. In the table, for some of the classes, such as 5 and 16, we have few missed predictions; however, the overall performance of this technique is acceptable for the birth control model.

For the decision tree technique, we start with the shallower tree (max-depth = 3) because, as we know, shallower trees are more efficient, and the accuracy we achieve is low. Even for some individual classes, it is zero. To check if we can represent our model by the decision tree algorithm, we change the depth to 7, and we get an accuracy of 95%, and the loss function is around 7%. So the decision tree works well.

Similarly, gradient boosting and AdaBoost classifiers do not fit well. We obtain 100% accuracy by using the gradient boosting classifier, which is unacceptable. The loss function is deficient and almost near zero, which is unacceptable. This result can be because of the over-fitting that happens when this technique is used for the model. In the AdaBoost classifier, the most accuracy we get is around 60%, and the training and test results do not match. So these two ensemble algorithms are unsuitable for the birth control model.

For both hard and soft voting, we can achieve reasonable results. For some of the classifiers, the accuracy is 100%, while for others, the accuracy is 78%. The results we obtain from both of these models are very similar; however, compared to KNN and weighted KNN, it takes longer to run these two algorithms. This means the runtime for both techniques is higher than KNN and weighted KNN. On the other hand, the SVM technique we used does not give us as high accuracy as expected. Initially, we know that SVM is not an excellent option for the birth control model, but we hope to increase the accuracy by adding the ”ovo” parameter and using this model. After all, as Fig. 7 shows, we obtain an accuracy of 14%, and the recall and f1-score for most of the classifiers are zero; therefore, this machine learning technique cannot be a reasonable option for training our data.

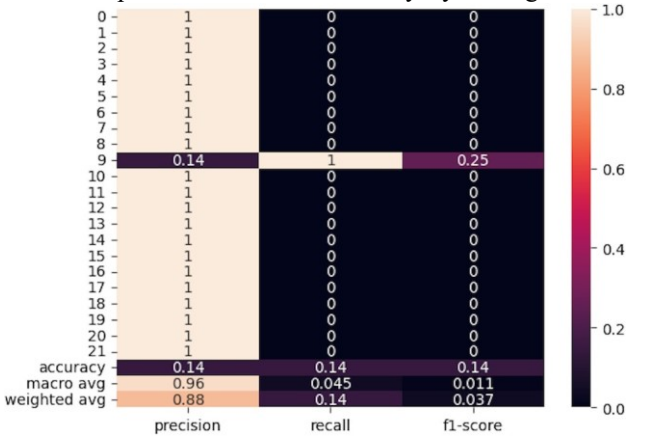

Fig. 7. Classification Report for SVM - birth control model

Like the gradient boosting technique, the logistic regression method is unsuitable since it gives us 100% accuracy, precision, and recall for all classifiers; therefore, logistic regression is not a good representation for training our data. This result is not a surprise for us because, from the beginning, we know that the problem that we are trying to solve is not a regression problem; however, similar to SVM, we hope to get a more reasonable result from this technique since we add ”ovo” as a parameter to the function before fitting the data.

Since the problem we are trying to solve is considered a classifier problem, for seven of the techniques we use for our model, we plot the ROC (receiver opening characteristic) curve and calculate the AUC (area under the ROC curve) score. Using the ROC curve, we understand each machine learning algorithm’s number of ”true positive” and ”true negative” values. This curve is beneficial for our analysis since it can give us a better understanding of how close model predictions are to the actual value. We plot the ROC curve for all our techniques except for logistic regression and SVM. Fig. 8 is an example of the ROC curve obtained from the decision tree. As this figure suggests, the average score under the curve is 97%. For most of the algorithms we use, the AUC score is above 95%, which is acceptable since our prediction’s quality is high. We cannot calculate the AUC score for both logistic regression and SVM. It is important to note that if we add a parameter ”Probability=True” to our SVM model before fitting the training set, we can calculate the AUC for SVM; however, it is very time-consuming to perform this action because it takes more than two hours for our model to get trained. As a result, we decide to ignore the AUC score for the Support Vector Machine technique.

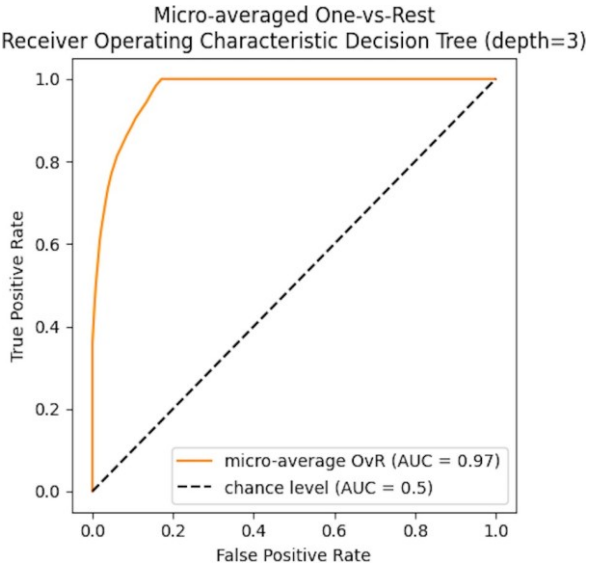

Fig. 8. ROC curve for decision tree - birth control model

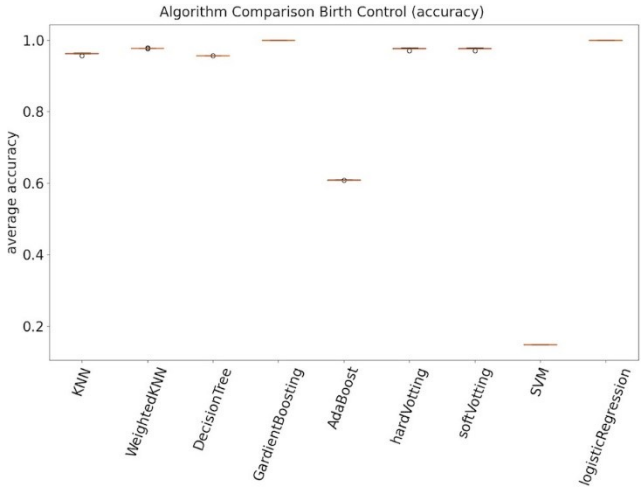

Fig. 9. Comparison of all machine learning techniques using their average accuracy

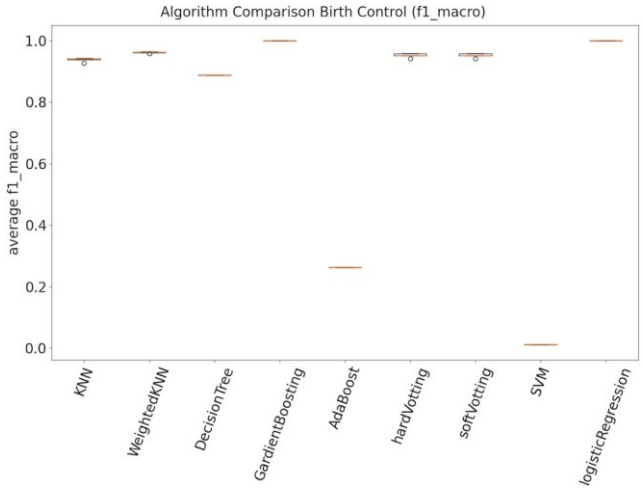

Fig. 10. Comparison of all machine learning techniques using their average f1-macro score

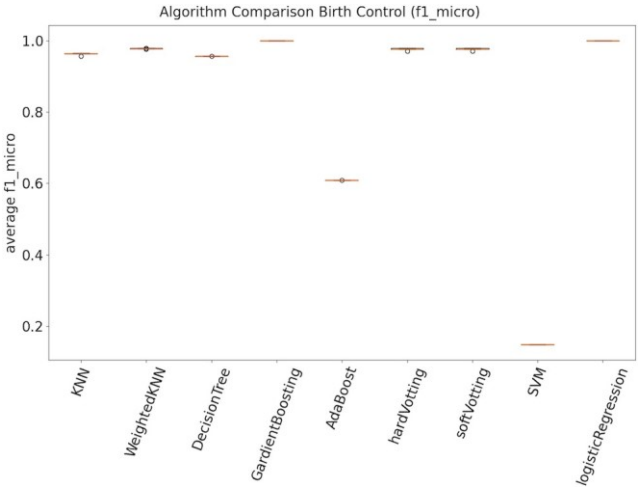

Fig. 11. Comparison of all machine learning techniques using their average f1-micro score

We use cross-validation (kFold = 10) to train and test the birth control model and use the average result we achieve for each fold to compare all the machine learning algorithms by plotting box plots for accuracy, f1-macro, and f1-micro. Fig. 9, Fig. 10, and Fig. 11 show the comparison of all nine different algorithms used for training and testing the model. As all three graphs suggest, there are not a lot of variations for each of these techniques. This means that regardless of how good or bad a technique is performed for the model, the result we obtain from each fold is similar. For Fig. 9, we compare all the average accuracy we achieve for each technique. As the graph suggests, SVM has the lowest accuracy, which is less than 20%, and gradient boosting and logistic regression have the highest accuracy, which is 100%. The results from Fig. 10 and Fig. 11, which show the average f1-macro and f1-micro, also suggest the same result. Given the overall performance of each technique, which combines accuracy, precision, recall, loss function, AUC score, and confusion matrix, weighted KNN is the best technique that fits the birth control model.

A. Analysis of Top Nine Most Frequent Conditions Model

Similar to the birth control model, for this model, we use eight different machine learning techniques to train the top nine most frequent conditions model. The only technique we do not use for this model while we use it for birth control is weighted KNN. We first use the LabelEncoder library to transform drug names (target values) into numerical values.

When we test the accuracy of our model using this library, we notice that the model's accuracy is 7% for KNN. We try other techniques, such as decision tree and gradient boosting, but do not get a better result, and the accuracy increase only by a few percent. The accuracy for the decision tree is 10%, and for gradient boosting is 9%. Since the accuracy obtained is low and unacceptable, we try to change the preprocessing step we perform to transform the drug names. For this reason, we try the OneHotEncoder library, which we initially used for transforming conditions. We specifically choose this library because it converts our target value into zero and one, reducing the number of classifiers we have and changing the problem we want to solve into a binary instead of a multi-classification problem. We reapply the same techniques and get reasonable results when we calculate the accuracy. For instance, the accuracy we achieve from the KNN is 95%, and for the decision tree is 99%. As a result, we move forward with this library; however, once we try to use other evaluation techniques, such as confusion matrix and classification report, we notice that the results are the same regardless of the machine learning technique we use. Similarly, as Fig. 12 suggests, all our techniques have a miss prediction when the target value equals one. This result states that using the OneHotEncoder for converting the problem into zero and one is not a good option for this model since the result does not represent the whole dataset, and the model is not appropriately trained.

Even though we achieve the same result for all machine learning techniques, we use cross-validation (kFold = 10) to train the model and compare the results for all the techniques we use. In Fig. 13, we compare all the techniques concerning their average accuracy. Fig. 14 and Fig. 15 compare all techniques vs. f1-macro and f1-micro. The result that we

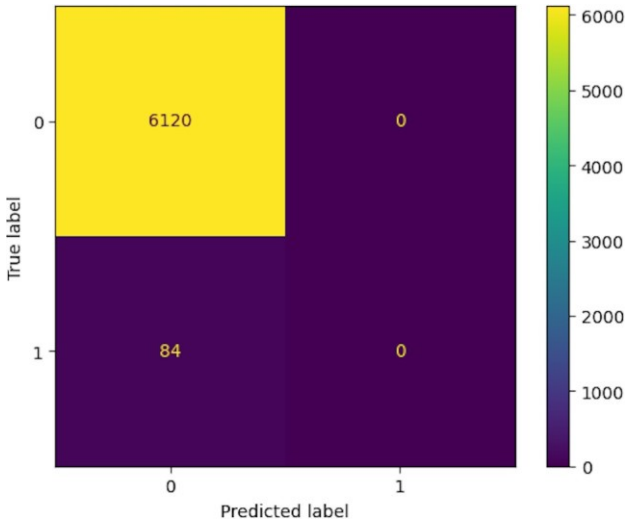

Fig. 12. Confusion Matrix - top nine most frequent conditions model

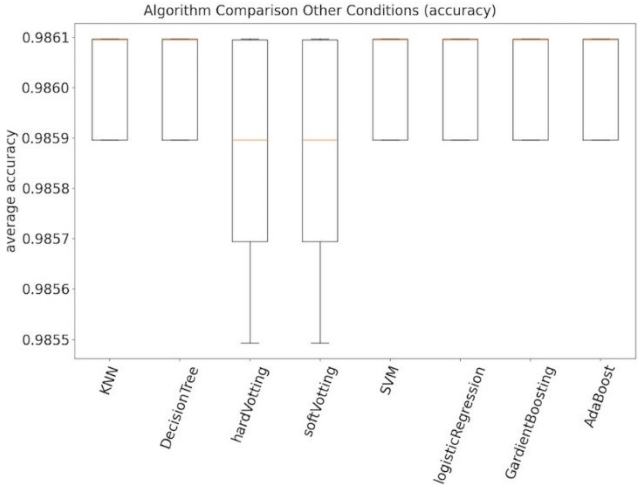

Fig. 13. Comparison of all machine learning techniques using their average accuracy - top nine most frequent conditions model

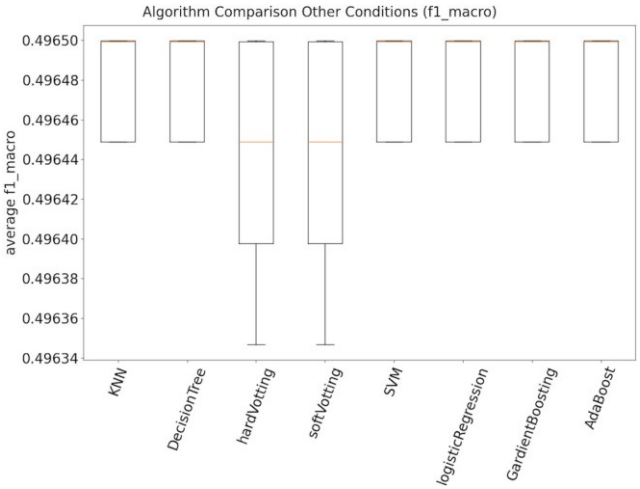

Fig. 14. Comparison of all machine learning techniques using their average f1-macro score - top nine most frequent conditions model

obtain from all three graphs is similar. Most of the techniques are right-skewed, except for our voting classifier technique.

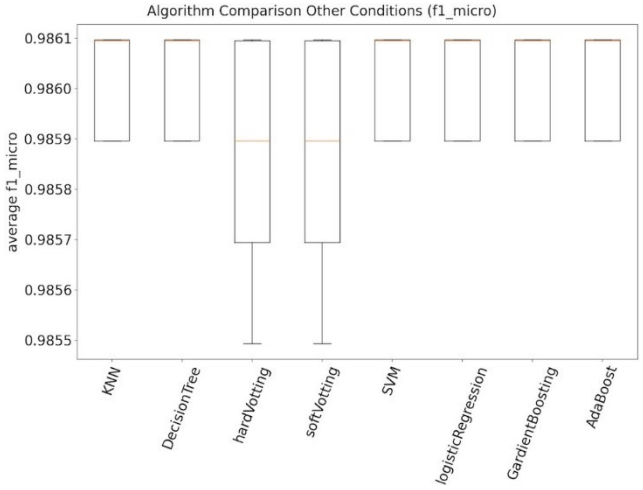

Fig. 15. Comparison of all machine learning techniques using their average f1-micro score - top nine most frequent conditions model

Based on the graph below, hard and soft voting is the best option for our model. We want to gain more realistic results from the top nine most frequent conditions model. Since different techniques are used to train and test this model, we expect to achieve more variation regarding the classification report, similar to what we obtain from the birth control model. Furthermore, the number of true positive predictions visualized using the confusion matrix needs more variation. For example, the decision tree can predict 5,000 true positive values while the KNN generates 4,000 true positive values; however, what we observe in this model, is the same result for all of the techniques we used. In addition, as Fig. 13, Fig. 14, and Fig. 15 suggest, the variation of the result we achieve when cross-validation is used for our data is higher than the birth control model.

VI. CONCLUSION

Machine learning can be beneficial for predicting drugs name based on customer reviews. As discussed in detail in our paper, the birth control model gives us more precision accuracy than the top nine most frequent conditions. A more focused machine learning model can make data preprocessing and model prediction easier. In our case, the birth control model, which focuses only on one medical condition, tends to be preprocessed easier and trained better than the top nine most frequent conditions model, which has more than five different conditions; therefore, having more focused data can generate more precise results. We believe using machine learning to predict drug names based on customer reviews can benefit patients and physicians prescribing the medication. Indeed, this model can be improved if more data is used for training; therefore, collecting more data from patients can play an essential role in obtaining more accurate results. Our proposed model can be used on a bigger scale to help doctors and patients choose the best type of a particular medication. For example, if one type of birth control medication causes more hair loss compared to another, doctors can recommend that to patients. As a result, this model can be used on a large

scale to help patients and physicians.

We train our models using various machine learning techniques to compare the result and get a more precise understanding of how each of the models performs. For instance, given the wide variety of classifiers for our top nine most frequent conditions model, we do not use the proper library (LabelEncoder) to convert the categories to numerical values, which impacts the result we obtain for this model. Similarly, we learn that looking at the accuracy percentage alone will not be enough to judge how reasonably a machine learning technique performs on the model. The accuracy we obtain from each technique for the top nine most frequent conditions model is high and varies between techniques; however, when we evaluate the model using other techniques, we notice an issue in the preprocessing step.

VII. FUTURE WORK

In the future, we would like to obtain more accurate results from the top nine most frequent conditions model. One of the ways that this issue can be fixed is by reducing the variation of the target value. Initially, this model contains 40 unique target values. As a result, the model does not get enough chance to be adequately trained for each target value. We can try to group drugs based on conditions or chemical components to fix this issue. This way, we keep the data and reduce the number of classifiers. Another way can be creating multiple sub-models containing one or two different drugs. To prevent data loss, we can use web scraping to generate more data regarding each drug.

REFERENCES

[1] P. Dhanush and N. Nalini, ”Drug Review System Using machine learn- ing by Comparing Linear Support Vector Machine with Na¨ıve Bayes Classifier to Measure Accuracy,” 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2022, pp. 1-5.

[2] M. D. Hossain, M. S. Azam, M. J. Ali and H. Sabit, ”Drugs Rating Generation and Recommendation from Sentiment Analysis of Drug Reviews using machine learning,” 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), Bangladesh, 2020, pp. 1-6.

[3] D. N. Swathi and K. U., ”Predicting Drug Side-effects from Open Source Health Forums using Supervised Classifier Approach,” 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 796-800.

[4] F. Gra¨ßer, S. Kallumadi, H. Malberg, and S. Zaunseder, ”Aspect-based sentiment analysis of drug reviews applying cross-domain and cross- data learning,” In Proceedings of the 2018 international conference on digital health, 2018, pp. 121-125.

[5] Gra¨ßer, Felix, Stefanie Beckert, Denise Ku¨ster, Susanne Abra-ham, Hagen Malberg, Jochen Schmitt, and Sebastian Zaunseder. ”Neighborhood-based Collaborative Filtering for Therapy Decision Support.” HealthRecSys@ RecSys (2017): 22-26.

[6] F.J. Grajales, S. Sheps, K. Ho, H. Novak-Lauscher, and G. Eysenbach, “Social Media: A Review and Tutorial of Applications in Medicine and Health Care,” Journal of medical Internet research, vol 16, no. 2, pp.e2912, Feb, 2014.

[7] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, and S. Zhao, ”Applications of machine learning in drug discovery and development,” Nature reviews Drug discovery, vol 18, no. 6, pp.463-477, April, 2019.

[8] Sahoo, Abhaya Kumar, Sitikantha Mallik, Chittaranjan Pradhan, Bha- bani Shankar Prasad Mishra, Rabindra Kumar Barik, and Himansu Das. ”Intelligence-based health recommendation system using big data analytics.” In Big data analytics for intelligent healthcare management,pp. 227-246. Academic Press, 2019.

[9] H. Ding, Hao, I. Takigawa, H. Mamitsuka, and S. Zhu, ”Similarity-based machine learning methods for predicting drug–target interactions: a brief review,” Briefings in bioinformatics, vol 15, no. 5, pp.734- 747, September, 2014.

[10] B. Mahesh, ”Machine learning algorithms-a review,” International Journal of Science and Research (IJSR) [Internet], vol 9, no. 1, pp.381- 386, January, 2020.

[11] A. Vabalas, E. Gowen, E. Poliakoff, and A.J. Casson, ”Machine learning algorithm validation with a limited sample size.” PloS one, vol 14(11), pp. e0224365, November, 2019.

[12] Mejova, Yelena. ”Sentiment analysis: An overview.” University of Iowa, Computer Science Department (2009).