DRUG RECOMMENDATION SYSTEM BASED ON SENTIMENT ANALYSIS OF DRUG REVIEWS USING MACHINE LEARNING

ABSTRACT

Since coronavirus has shown up, inaccessibility of legitimate clinical resources is at its peak, like the shortage of specialists and healthcare workers, lack of proper equipment and medicines etc. The entire medical fraternity is in distress, which results in numerous individual's demise. Due to unavailability, individuals started taking medication independently without appropriate consultation, making the health condition worse than usual. As of late, machine learning has been valuable in numerous applications, and there is an increase in innovative work for automation. This paper intends to present a drug recommender system that can drastically reduce specialist's heap. In this research, we build a medicine recommendation system that uses patient reviews to predict the sentiment using various vectorization processes like Bow, TF-IDF, Word2Vec, and Manual Feature Analysis, which can help recommend the top drug for a given disease by different classification algorithms. The predicted sentiments were evaluated by precision, recall, f1score, accuracy, and AUC score. The results show that classifier Linear SVC using TF-IDF vectorization outperforms all other models with 93% accuracy.

Keywords— Drug, Recommender System, Machine Learning, NLP, Smote, Bow, TF-IDF, Word2Vec, Sentiment analysis.

CHAPTER1: INTRODUCTION

While the number of coronavirus cases growing exponentially, the nations are facing a shortage of doctors, particularly in rural areas where the quantity of specialists is less compared to urban areas. A doctor takes roughly 6 to 12 years to procure the necessary qualifications. Thus, the number of doctors can't be expanded quickly in a short time frame. A Telemedicine framework ought to be energized as far as possible in this difficult time. Clinical blunders are very regular nowadays. Over 200 thousand individuals in China and 100 thousand in the USA are affected every year because of prescription mistakes. Over 40% medicine, specialists make mistakes while prescribing since specialists compose the solution as referenced by their knowledge, which is very restricted. Choosing the top level medication is significant for patients who need specialists that know wide-based information about microscopic organisms, antibacterial medications, and patients.

Every day a new study comes up with accompanying more drugs, tests, accessible for clinical staff every day. Accordingly, it turns out to be progressively challenging for doctors to choose which treatment or medications to give to a patient based on indications, past clinical history. With the exponential development of the web and the web-based business industry, item reviews have become an imperative and integral factor for acquiring items worldwide. Individuals worldwide become adjusted to analyze reviews and websites first before settling on a choice to buy a thing.

While most of past exploration zeroed in on rating expectation and proposals on the E-Commerce field, the territory of medical care or clinical therapies has been infrequently taken care of. There has been an expansion in the number of individuals worried about their well-being and finding a diagnosis online.

As demonstrated in a Pew American Research center survey directed in 2013 , roughly 60% of grown-ups searched online for health-related subjects, and around 35% of users looked for diagnosing health conditions on the web. A medication recommender framework is truly vital with the goal that it can assist specialists and help patients to build their knowledge of drugs on specific health conditions. A recommender framework is a customary system that proposes an item to the user, dependent on their advantage and necessity. These frameworks employ the customers' surveys to break down their sentiment and suggest a recommendation for their exact need.

In the drug recommender system, medicine is offered on a specific condition dependent on patient reviews using sentiment analysis and feature engineering. Sentiment analysis is a progression of strategies, methods, and tools for distinguishing and extracting emotional data, such as opinion and attitudes, from language [7]. On the other hand, Featuring engineering is the process of making more features from the existing ones; it improves the performance of models.

This examination work separated into five segments: Introduction area which provides a short insight concerning the need of this research, Related works segment gives a concise insight regarding the previous examinations on this area of study, Methodology part includes the methods adopted in this research, The Result segment evaluates applied model results using various metrics, the Discussion section contains limitations of the framework, and lastly, the conclusion section.

1.1 RECOMMENDATION SYSTEM

A recommendation system is a subclass of Information filtering Systems that seeks to predict the rating or the preference a user might give to an item. In simple words, it is an algorithm that suggests relevant items to users. Eg: In the case of Netflix which movie to watch, In the

case of e-commerce which product to buy, or In the case of kindle which book to read, etc.

1.1.1 Uses of Recommendation System

There are many use-cases of it. Some are

A. Personalized Content: Helps to Improve the on-site experience by creating dynamic recommendations for different kinds of audiences like Netflix does.

B. Better Product search experience: Helps to categories the product based on their features. Eg: Material, Season, etc.

1.1.2 Types of Recommendation System

A. Content-Based Filtering :

Content-based filtering uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback. To demonstrate content-based filtering, let's hand-engineer some features for the Google Play store. The following figure shows a feature matrix where each row represents an app and each column represents a feature. Features could include categories (such as Education, Casual, and Health), the publisher of the app, and many others.

To simplify, assume this feature matrix is binary: a non-zero value means the app has that feature. You also represent the user in the same feature space. Some of the user-related features could be explicitly provided by the user. For example, a user selects "Entertainment apps" in their profile. Other features can be implicit, based on the apps they have previously installed. For example, the user installed another app published by Science.

The model should recommend items relevant to this user. To do so, you must first pick a similarity metric (for example, dot product). Then, you must set up the system to score each candidate item according to this similarity metric. Note that the recommendations are specific to this user, as the model did not use any information about other users.

B. Collaborative Based Filtering:

To address some of the limitations of content-based filtering, collaborative filtering uses similarities between users and items simultaneously to provide recommendations. This allows for serendipitous recommendations; that is, collaborative filtering models can recommend an item to user A based on the interests of a similar user B. Furthermore, the embedding's can be learned automatically, without relying on hand-engineering of features.

Collaborative Filtering is a Machine Learning technique used to identify relationships between pieces of data. This technique is frequently used in recommender systems to identify similarities between user data and items. This means that if Users A and B both like Product A, and User B also likes Product B, then Product B could be recommended to User A by the system.

The model keeps track of what products users like and their characteristics to see what users, who like products with similar characteristics, enjoyed. The model then makes its recommendations accordingly. Product features should be given numerical values whenever possible as it makes decisions by the model more accurate.

Once features are identified and assigned values, data collection needs to begin. There are two ways the model can identify whether or not a user enjoyed a product. The user can be asked to give a numerical

rating or the system can assume that the user likes whatever product they use. Once user interests have been established, recommendations can be made.

1.2 Machine learning

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. Recommendation engines are a common use case for machine learning. Other popular uses include fraud detection, spam filtering, malware threat detection, business process automation (BPA) and Predictive maintenance.

Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm data scientists choose to use depends on what type of data they want to predict.

1. Supervised learning:

In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the

algorithm to assess for correlations. Both the input and the output of the algorithm are specified.

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix.

Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs.[36] An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task. Types of supervised-learning algorithms include active learning, classification and regression.

Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. As an example, for a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email.

Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation

systems, visual identity tracking, face verification, and speaker verification.

2. Unsupervised learning:

This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data.

A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. Though unsupervised learning encompasses other domains involving summarizing and explaining data features. Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more pre designated criteria, while observations drawn from different clusters are dissimilar.

Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

3. Semi-supervised learning:

This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

4. Reinforcement learning:

Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

Supervised machine learning requires the data scientist to train the algorithm with both labeled inputs and desired outputs. Supervised learning algorithms are good for the following tasks:

a) **Binary classification:** Dividing data into two categories. Multi-class classification: Choosing between more than two types of answers. Regression modeling: Predicting continuous values.
b) **Ensembling:** Combining the predictions of multiple machine learning models to produce an accurate prediction.Unsupervised machine learning algorithms do not require data to be labeled. They sift through unlabeled data to look for patterns that can be used to group data points into subsets. Most types of deep learning, including neural networks, are unsupervised algorithms.
c) **Clustering:** Splitting the dataset into groups based on similarity. Anomaly detection: Identifying unusual data points in a data set. Association mining: Identifying sets of items in a data set that frequently occur together.

d) **Dimensionality reduction:** Reducing the number of variables in a data set. Semi-supervised learning works by data scientists feeding a small amount of labeled training data to an algorithm. From this, the algorithm learns the dimensions of the data set, which it can then apply to new, unlabeled data. The performance of algorithms typically improves when they train on labeled data sets. But labeling data can be time consuming and expensive. Semi-supervised learning strikes a middle ground between the performance of supervised learning and the efficiency of unsupervised learning. Some areas where semi-supervised learning is used include: Machine translation: Teaching algorithms to translate language based on less than a full dictionary of words.

5. Fraud detection:

Identifying cases of fraud when you only have a few positive examples. Labelling data: Algorithms trained on small data sets can learn to apply data labels to larger sets automatically. How does reinforcement learning work? Reinforcement learning works by programming an algorithm with a distinct goal and a prescribed set of rules for accomplishing that goal.

Data scientists also program the algorithm to seek positive rewards - which it receives when it performs an action that is beneficial toward the ultimate goal -- and avoid punishments -- which it receives when it performs an action that gets it farther away from its ultimate goal. Reinforcement learning is often used in areas such as: Robotics: Robots can learn to perform tasks the physical world using this technique.

6. Data mining:

Machine learning and data mining often employ the same methods and overlap significantly, but while machine learning focuses on prediction, based on known properties learned from the training data, data mining focuses on the discovery of (previously) unknown properties in the data (this is the analysis step of knowledge discovery in databases).

Data mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as "unsupervised learning" or as a preprocessing step to improve learner accuracy.

Much of the confusion between these two research communities (which do often have separate conferences and separate journals, ECML PKDD being a major exception) comes from the basic assumptions they work with: in machine learning, performance is usually evaluated with respect to the ability to reproduce known knowledge, while in knowledge discovery and data mining (KDD) the key task is the discovery of previously unknown knowledge.

7. Optimization:

Machine learning also has intimate ties to optimization: many learning problems are formulated as minimization of some loss function on a training set of examples. Loss functions express the discrepancy between the predictions of the model being trained and the actual problem instances (for example, in classification, one wants to assign a label to instances, and models are trained to correctly predict the pre-assigned labels of a set of examples).

8. Generalization:

The difference between optimization and machine learning arises from the goal of generalization: while optimization algorithms can

minimize the loss on a training set, machine learning is concerned with minimizing the loss on unseen samples. Characterizing the generalization of various learning algorithms is an active topic of current research, especially for deep learning algorithms.

9. Video gameplay:

Reinforcement learning has been used to teach bots to play a number of video games. Resource management: Given finite resources and a defined goal, reinforcement learning can help enterprises plan out how to allocate resources.

Today, machine learning is used in a wide range of applications. Perhaps one of the most well-known examples of machine learning in action is the recommendation engine that powers Facebook's news feed. Facebook uses machine learning to personalize how each member's feed is delivered. If a member frequently stops to read a particular group's posts, the recommendation engine will start to show more of that group's activity earlier in the feed.

Behind the scenes, the engine is attempting to reinforce known patterns in the member's online behavior. Should the member change patterns and fail to read posts from that group in the coming weeks, the news feed will adjust accordingly. In addition to recommendation engines, other uses for machine learning include the following: Customer relationship management. CRM software can use machine learning models to analyze email and prompt sales team members to respond to the most important messages first.

More advanced systems can even recommend potentially effective responses. Business intelligence. BI and analytics vendors use machine learning in their software to identify potentially important data points,

patterns of data points and anomalies. Human resource information systems.

a) Advantages and disadvantages of machine learning:

Machine learning has seen use cases ranging from predicting customer behavior to forming the operating system for self-driving cars. When it comes to advantages, machine learning can help enterprises understand their customers at a deeper level. By collecting customer data and correlating it with behaviors over time, machine learning algorithms can learn associations and help teams tailor product development and marketing initiatives to customer demand.

Some companies use machine learning as a primary driver in their business models. Uber, for example, uses algorithms to match drivers with riders. Google uses machine learning to surface the ride advertisements in searches. But machine learning comes with disadvantages. First and foremost, it can be expensive. Machine learning projects are typically driven by data scientists, who command high salaries. These projects also require software infrastructure that can be expensive.

There is also the problem of machine learning bias. Algorithms trained on data sets that exclude certain populations or contain errors can lead to inaccurate models of the world that, at best, fail and, at worst, are discriminatory. When an enterprise bases core business processes on biased models it can run into regulatory and reputational harm.

b) Right machine learning model

The process of choosing the right machine learning model to solve a problem can be time consuming if not approached strategically.

Step 1: Align the problem with potential data inputs that should be considered for the solution. This step requires help from data scientists and experts who have a deep understanding of the problem.

Step 2: Collect data, format it and label the data if necessary. This step is typically led by data scientists, with help from data wranglers.

Step 3: Chose which algorithm(s) to use and test to see how well they perform. This step is usually carried out by data scientists.

Step 4: Continue to fine tune outputs until they reach an acceptable level of accuracy. This step is usually carried out by data scientists with feedback from experts who have a deep understanding of the problem.

c) Future of machine learning

While machine learning algorithms have been around for decades, they've attained new popularity as artificial intelligence has grown in prominence. Deep learning models, in particular, power today's most advanced AI applications. Machine learning platforms are among enterprise technology's most competitive realms, with most major vendors, including Amazon, Google, Microsoft, IBM and others, racing to sign customers up for platform services that cover the spectrum of machine learning activities, including data collection, data preparation, data classification, model building, training and application deployment.

As machine learning continues to increase in importance to business operations and AI becomes more practical in enterprise settings, the machine learning platform wars will only intensify. Continued research into deep learning and AI is increasingly focused on developing more general applications.

Today's AI models require extensive training in order to produce an algorithm that is highly optimized to perform one task. But some

researchers are exploring ways to make models more flexible and are seeking techniques that allow a machine to apply context learned from one task to future, different tasks.

1.3 Natural Language Processing (NLP)

Natural language processing strives to build machines that understand and respond to text or voice data—and respond with text or speech of their own—in much the same way humans do.

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can. NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models.

Together, these technologies enable computers to process human language in the form of text or voice data and to 'understand' its full meaning, complete with the speaker or writer's intent and sentiment. NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real time.

There's a good chance you've interacted with NLP in the form of voice-operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes.

Human language is filled with ambiguities that make it incredibly difficult to write software that accurately determines the intended

meaning of text or voice data. Homonyms, homophones, sarcasm, idioms, metaphors, grammar and usage exceptions, variations in sentence structure—these just a few of the irregularities of human language that take humans years to learn, but that programmers must teach natural language-driven applications to recognize and understand accurately from the start, if those applications are going to be useful. Several NLP tasks break down human text and voice data in ways that help the computer make sense of what it's ingesting. Some of these tasks include the following:

1. **Speech recognition:** also called speech-to-text, is the task of reliably converting voice data into text data. Speech recognition is required for any application that follows voice commands or answers spoken questions. What makes speech recognition especially challenging is the way people talk—quickly, slurring words together, with varying emphasis and intonation, in different accents, and often using incorrect grammar.
2. **Part of speech tagging:** also called grammatical tagging is the process of determining the part of speech of a particular word or piece of text based on its use and context. Part of speech identifies 'make' as a verb in 'I can make a paper plane,' and as a noun in 'what make of car do you own?'
3. **Word sense disambiguation**: The selection of the meaning of a word with multiple meanings through a process of semantic analysis that determine the word that makes the most sense in the given context. For example, word sense disambiguation helps distinguish the meaning of the verb 'make' in 'make the grade' (achieve) vs. 'make a bet' (place).
4. **Named entity recognition, or NEM:** Identifies words or phrases as useful entities. NEM identifies 'Kentucky' as a location or 'Fred' as a man's name.

5. **Co-reference resolution:** The task of identifying if and when two words refer to the same entity. The most common example is determining the person or object to which a certain pronoun refers (e.g., 'she' = 'Mary'), but it can also involve identifying a metaphor or an idiom in the text (e.g., an instance in which 'bear' isn't an animal but a large hairy person).
6. **Sentiment analysis:** attempts to extract subjective qualities—attitudes, emotions, sarcasm, confusion, suspicion—from text. Natural language generation is sometimes described as the opposite of speech recognition or speech-to-text; it's the task of putting structured information into human language.

a) Python and the Natural Language Toolkit (NLTK)

The Python programing language provides a wide range of tools and libraries for attacking specific NLP tasks. Many of these are found in the Natural Language Toolkit, or NLTK, an open source collection of libraries, programs, and education resources for building NLP programs.

The NLTK includes libraries for many of the NLP tasks listed above, plus libraries for subtasks, such as sentence parsing, word segmentation, stemming and lemmatization (methods of trimming words down to their roots), and tokenization (for breaking phrases, sentences, paragraphs and passages into tokens that help the computer better understand the text).

It also includes libraries for implementing capabilities such as semantic reasoning, the ability to reach logical conclusions based on facts extracted from text. Statistical NLP, machine learning, and deep learning the earliest NLP applications were hand-coded, rules-based systems that could perform certain NLP tasks, but couldn't easily scale

to accommodate a seemingly endless stream of exceptions or the increasing volumes of text and voice data.

Enter statistical NLP, which combines computer algorithms with machine learning and deep learning models to automatically extract, classify, and label elements of text and voice data and then assign a statistical likelihood to each possible meaning of those elements. Today, deep learning models and learning techniques based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) enable NLP systems that 'learn' as they work and extract ever more accurate meaning from huge volumes of raw, unstructured, and unlabeled text and voice data sets.

NLP use cases Natural language processing is the driving force behind machine intelligence in many modern real-world applications. Here are a few examples: Spam detection: You may not think of spam detection as an NLP solution, but the best spam detection technologies use NLP's text classification capabilities to scan emails for language that often indicates spam or phishing. These indicators can include overuse of financial terms, characteristic bad grammar, threatening language, inappropriate urgency, misspelled company names, and more.

Spam detection is one of a handful of NLP problems that experts consider 'mostly solved' (although you may argue that this doesn’t match your email experience). Machine translation: Google Translate is an example of widely available NLP technology at work. Truly useful machine translation involves more than replacing words in one language with words of another. Effective translation has to capture accurately the meaning and tone of the input language and translate it to text with the same meaning and desired impact in the output language.

Machine translation tools are making good progress in terms of accuracy. A great way to test any machine translation tool is to translate text to one language and then back to the original. An oft-cited classic example: Not long ago, translating “The spirit is willing but the flesh is weak” from English to Russian and back yielded “The vodka is good but the meat is rotten.” Today, the result is “The spirit desires, but the flesh is weak,” which isn’t perfect, but inspires much more confidence in the English-to-Russian translation.

Virtual agents and chatbots: Virtual agents such as Apple's Siri and Amazon's Alexa use speech recognition to recognize patterns in voice commands and natural language generation to respond with appropriate action or helpful comments. Chatbots perform the same magic in response to typed text entries. The best of these also learn to recognize contextual clues about human requests and use them to provide even better responses or options over time.

The next enhancement for these applications is question answering, the ability to respond to our questions—anticipated or not—with relevant and helpful answers in their own words. Social media sentiment analysis: NLP has become an essential business tool for uncovering hidden data insights from social media channels.

Sentiment analysis can analyze language used in social media posts, responses, reviews, and more to extract attitudes and emotions in response to products, promotions, and events–information companies can use in product designs, advertising campaigns, and more.

- **Text summarization:** Text summarization uses NLP techniques to digest huge volumes of digital text and create summaries and synopses for indexes, research databases, or busy readers who don't have time to read full text. The best text summarization

applications use semantic reasoning and natural language generation (NLG) to add useful context and conclusions to summaries.

1.4 Sentiment analysis

Sentiment analysis is an automated process capable of understanding the feelings or opinions that underlie a text. It is one of the most interesting subfields of NLP, a branch of Artificial Intelligence (AI) that focuses on how machines process human language.

a) Types:

A basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral.

Advanced, "beyond polarity" sentiment classification looks, for instance, at emotional states such as enjoyment, anger, disgust, sadness, fear, and surprise. Precursors to sentimental analysis include the General Inquirer, which provided hints toward quantifying patterns in text and, separately, psychological research that examined a person's psychological state based on analysis of their verbal behavior. Subsequently, the method described in a patent by Volcani and Fogel, looked specifically at sentiment and identified individual words and phrases in text with respect to different emotional scales.

A current system based on their work, called Effect Check, presents synonyms that can be used to increase or decrease the level of evoked emotion in each scale. Many other subsequent efforts were less sophisticated, using a mere polar view of sentiment, from positive to negative applied different methods for detecting the polarity of product reviews and movie reviews respectively. This work is at the document

level. One can also classify a document's polarity on a multi-way scale, expanded the basic task of classifying a movie review as either positive or negative to predict star ratings on either a 3- or a 4-star scale, while performed an in-depth analysis of restaurant reviews, predicting ratings for various aspects of the given restaurant, such as the food and atmosphere (on a five-star scale).

First steps to bringing together various approaches—learning, lexical, knowledge-based, etc.—were taken in the 2004 AAAI Spring Symposium where linguists, computer scientists, and other interested researchers first aligned interests and proposed shared tasks and benchmark data sets for the systematic computational research on affect, appeal, subjectivity, and sentiment in text. Even though in most statistical classification methods, the neutral class is ignored under the assumption that neutral texts lie near the boundary of the binary classifier, several researchers suggest that, as in every polarity problem, three categories must be identified. Moreover, it can be proven that specific classifiers such as the Max Entropy and SVMs can benefit from the introduction of a neutral class and improve the overall accuracy of the classification.

There are in principle two ways for operating with a neutral class. Either, the algorithm proceeds by first identifying the neutral language, filtering it out and then assessing the rest in terms of positive and negative sentiments, or it builds a three-way classification in one step. This second approach often involves estimating a probability distribution over all categories (e.g. naive Bayes classifiers as implemented by the NLTK). Whether and how to use a neutral class depends on the nature of the data: if the data is clearly clustered into neutral, negative and positive language, it makes sense to filter the neutral language out and focus on the polarity between positive and

negative sentiments. If, in contrast, the data are mostly neutral with small deviations towards positive and negative affect, this strategy would make it harder to clearly distinguish between the two poles. A different method for determining sentiment is the use of a scaling system whereby words commonly associated with having a negative, neutral, or positive sentiment with them are given an associated number on a −10 to +10 scale (most negative up to most positive) or simply from 0 to a positive upper limit such as +4.

This makes it possible to adjust the sentiment of a given term relative to its environment (usually on the level of the sentence). When a piece of unstructured text is analyzed using natural language processing, each concept in the specified environment is given a score based on the way sentiment words relate to the concept and its associated score.

b) Subjectivity/objectivity identification:

This task is commonly defined as classifying a given text (usually a sentence) into one of two classes: objective or subjective. This problem can sometimes be more difficult than polarity classification. The subjectivity of words and phrases may depend on their context and an objective document may contain subjective sentences (e.g., a news article quoting people's opinions). Moreover, results are largely dependent on the definition of subjectivity used when annotating texts. However, showed that removing objective sentences from a document before classifying its polarity helped improve performance.

1. Metaphorical expressions:

The text contains metaphoric expression may impact on the performance on the extraction. Besides, metaphors take in different forms, which may have been contributed to the increase in detection.

2. Discrepancies in writings:

For the text obtained from the Internet, the discrepancies in the writing style of targeted text data involve distinct writing genres and styles Context-sensitive. Classification may vary based on the subjectiveness or objectiveness of previous and following sentences. Time-sensitive attribute. The task is challenged by some textual data's time-sensitive attribute. If a group of researchers wants to confirm a piece of fact in the news, they need a longer time for cross-validation, than the news becomes outdated. Cue words with fewer usages.

3. Ever-growing volume:

The task is also challenged by the sheer volume of textual data. The textual data's ever-growing nature makes the task overwhelmingly difficult for the researchers to complete the task on time.

4. Feature/aspect-based:

It refers to determining the opinions or sentiments expressed on different features or aspects of entities, e.g., of a cell phone, a digital camera, or a bank. A feature or aspect is an attribute or component of an entity, e.g., the screen of a cell phone, the service for a restaurant, or the picture quality of a camera. The advantage of feature-based sentiment analysis is the possibility to capture nuances about objects of interest. Different features can generate different sentiment responses, for example a hotel can have a convenient location, but mediocre food. This problem involves several sub-problems, e.g., identifying relevant entities, extracting their features/aspects, and determining whether an opinion expressed on each feature/aspect is positive, negative or neutral. The automatic identification of features can be performed with syntactic methods, with topic modeling, or with deep learning.

5. Intensity Ranking:

Emotions and sentiments are subjective in nature. The degree of emotions/sentiments expressed in a given text at the document, sentence, or feature/aspect level—to what degree of intensity is expressed in the opinion of a document, a sentence or an entity differs on a case-to-case basis. However, predicting only the emotion and sentiment does not always convey complete information. The degree or level of emotions and sentiments often plays a crucial role in understanding the exact feeling within a single class (e.g., 'good' versus 'awesome'). Some methods leverage a stacked ensemble method for predicting intensity for emotion and sentiment by combining the outputs obtained and using deep learning models based on convolutional neural networks, long short-term memory networks and gated recurrent units.

CHAPTER 2: LITERATURE SURVEY

M.Emmanuel [8] ”Probabilistic aspect mining approach for interpretation and evaluation of drug reviews,” 2016 International Conference on Signal Processing, Communication, Power and Embedded System, Paralakhemundi, 2016. Text summarization uses NLP techniques to digest huge volumes of digital text and create summaries and synopses for indexes, research databases, or busy readers who don't have time to read full text. The best text summarization applications use semantic reasoning and natural language generation (NLG) to add useful context and conclusions to summaries.

Leilei Sun [10] “Collaborative Filtering is a Machine Learning technique used to identify relationships between pieces of data.” This technique is frequently used in recommender systems to identify similarities between

user data and items. This means that if Users A and B both like Product A, and User B also likes Product B, then Product B could be recommended to User A by the system. . Data-driven and Recommendation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16).

V. Goel [11] ”Sentiment Analysis of Multilingual Twitter Data using Natural Language Processing,” 2018 8th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2018, pp. 208-212, doi: 10.1109/CSNT.2018.8820254. Determining the opinions or sentiments expressed on different features or aspects of entities A feature or aspect is an attribute or component of an entity, The advantage of feature-based sentiment analysis is the possibility to capture nuances about objects of interest. metaphoric expression may impact on the performance on the extraction. Besides, metaphors take in different forms, which may have been contributed to the increase in detection.

Y. Bao [13] ”An intelligent medicine recommender system framework,” 2016 IEEE 11th Conference on Industrial Electronics and Applications Classification may vary based on the subjectiveness or objectiveness of previous and following sentences. Time-sensitive attribute. The task is challenged by some textual data's time-sensitive attribute. If a group of researchers wants to confirm a piece of fact in the news, they need a longer time for cross-validation, than the news becomes outdated. Cue words with fewer usages.

J. Deng [15] ”Tweet modeling with LSTM recurrent neural networks for hashtag recommendation,” 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 1570-1577, doi: 10.1109/IJCNN.2016.7727385. When a piece of unstructured text is

analyzed using natural language processing, each concept in the specified environment is given a score based on the way sentiment words relate to the concept and its associated score. This allows movement to a more sophisticated understanding of sentiment, because it is now possible to adjust the sentiment value of a concept relative to modifications that may surround it. Words, for example, that intensify, relax or negate the sentiment expressed by the concept can affect its score.

J. Ramos [17] "Using tf-idf to determine word relevance in document queries," in Proceedings of the first instructional conference on machine learning, vol. 242, pp. 133–142, Piscataway, NJ, 2003. Text summarization uses NLP techniques to digest huge volumes of digital text and create summaries and synopses for indexes, research databases, or busy readers who don't have time to read full text. The best text summarization applications use semantic reasoning and natural language generation (NLG) to add useful context and conclusions to summaries.

Haibo He [24] , "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," 2008 IEEE International Joint Conference on Neural Networks. The NLTK includes libraries for many of the NLP tasks listed above, plus libraries for subtasks, such as sentence parsing, word segmentation, stemming and lemmatization methods of trimming words down to their roots, and tokenization for breaking phrases, sentences, paragraphs and passages into tokens that help the computer better understand the text.

Z. Wang [25] "SMOTE Tomek Based Resampling for Personality Recognition," in IEEE Access, vol. 7, pp. 129678-129689, 2019, doi: 10.1109/ACCESS.2019.2940061. Semi-supervised learning works by data scientists feeding a small amount of labeled training data to an

algorithm. From this, the algorithm learns the dimensions of the data set, which it can then apply to new, unlabeled data. The performance of algorithms typically improves when they train on labeled data sets. But labeling data can be time consuming and expensive. Semi-supervised learning strikes a middle ground between the performance of supervised learning and the efficiency of unsupervised learning.

Danushka Bollegala [19] "Content-based filtering uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback". To demonstrate content-based filtering, let's hand-engineer some features for the Google Play store. The following figure shows a feature matrix where each row represents an app and each column represents a feature. Features could include categories such as Education, Casual, Health, the publisher of the app, and many others. The model should recommend items relevant to this user. To do so, you must first pick a similarity metric. Then, you must set up the system to score each candidate item according to this similarity metric. Note that the recommendations are specific to this user, as the model did not use any information about other users.

Doulaverakis [9] Health Online 2013. Pew Research Internet Project Report. "Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs". The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix.

CHAPTER 3 :EXISTING SYSTEM

A medication recommender framework is truly vital with the goal that it can assist specialists and help patients to build their knowledge of drugs on specific health conditions. A recommender framework is a customary system that proposes an item to the user, dependent on their advantage and necessity.

A) The Cold-Start Problem: Collaborative filtering systems are based on the action of available data from similar users. If you are building a brand new recommendation system, you would have no user data to start with. You can use content-based filtering first and then move on to the collaborative filtering approach.

B) Scalability: As the number of users grow, the algorithms suffer scalability issues. If you have 10 million customers and 100,000 movies, you would have to create a sparse matrix with one trillion elements. The lack of right data: Input data may not always.

Recommender frameworks point to supply clients with personalized stock and repair to alter the expanding online information over-burden drawback. Various recommender frame work methods are anticipated since the mid1990s, and numerous shapes of recommender framework code were created as of late for a spread of applications.

The health-related substance shared through on-line feedbacks or surveys contains covered up assumption designs that emerges through totally distinctive sources from medical world which offer benefits to the pharmaceutical industry. Amid this, the on-line component is fantastically standard of late for online looking, diverse stock through distinctive websites like on-line buying of drugs at entryway step. Numerous websites and blogs offers clients to rate their stock with their fulfillment and quality of stock, logistics, administrations and criticism etc., which the clients examines for a particular medicine or on quality of administration. In the existing work, the system did not implement an exact sentiment analysis for large data sets.

First, we defined search terms based on population, intervention, outcome of relevance and experimental design. However, we concluded that for our approach the population contains all healthcare facilities. Since this population is so comprehensive and non-specific, we excluded keywords about the population.

This resulted in the following major keywords:

1. Intervention: medication recommendation system
2. Outcome of relevance: system for medication recommendation
3. Experimental Design: empirical studies, systematic literature reviews, solution descriptionsThe intervention and outcome of relevance category are the same. Therefore, they were only included one time. Once this has been agreed on, the search algorithm was constructed. The logical operators AND as well as

OR were used to combine the search terms defined in the previous step.

The following synonyms were considered:

1. Medication: Medication, drug, Drug
2. Recommendation: Recommendation,Recommender,Recommender
3. System: System, framework, Framework, algorithm, Algorithm, engine, Engine.

This resulted in the following search algorithm:

{{medication OR Medication OR drug OR Drug} AND {recommendation OR Recommendation OR recommender OR Recommender} AND {system OR System OR Engine OR engine OR framework OR Framework OR algorithm OR Algorithm.

Most of the people tend to live a long and healthy life, where they are more conscious about their health. But many studies show that almost many people die due to the medical errors caused in terms of taking wrong medicines and these errors are caused by doctors, who prescribe medicines based on their experiences which are quite limited. As machine learning, deep learning and data mining like technologies that are emerging day by day, these technologies can help us to explore the medical history and can reduce medical errors by being doctor friendly.

CHAPTER 4 :PROPOSED SYSTEM

The dataset used in this research is Drug Review Dataset (Drugs.com) taken from the UCI ML repository . This dataset contains six attributes, name of drug used (text), review (text) of a patient, condition (text) of a patient, useful count (numerical) which suggest the number of individuals who found the review helpful, date (date) of review entry, and a 10-star patient rating (numerical) determining overall patient contentment.

It contains a total of 215063 instances. Fig. 3.1 shows the proposed model used to build a medicine recommender system. It contains four stages, specifically, Data preparation, classification, evaluation, and Recommendation.

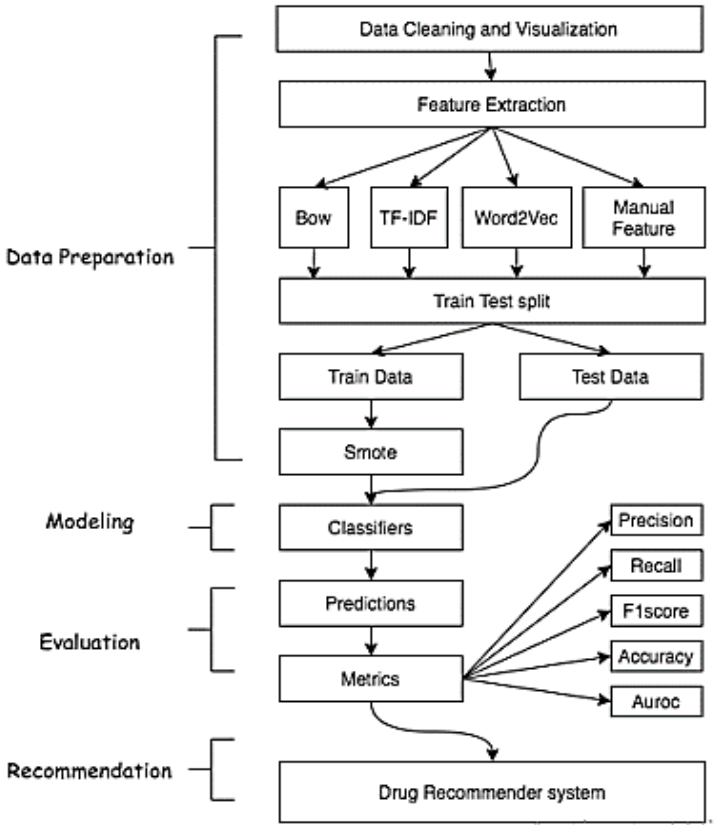

Fig 4.1 Flow Chart of the Proposed Model.

A. Data Cleaning and Visualizations :

Applied standard Data preparation techniques like checking null values, duplicate rows, removing unnecessary values, and text from rows in this research. Subsequently, removed all 1200 null values rows in the conditions column, as shown in Fig. 3.2. We make sure that a unique id should be unique to remove duplicacy.

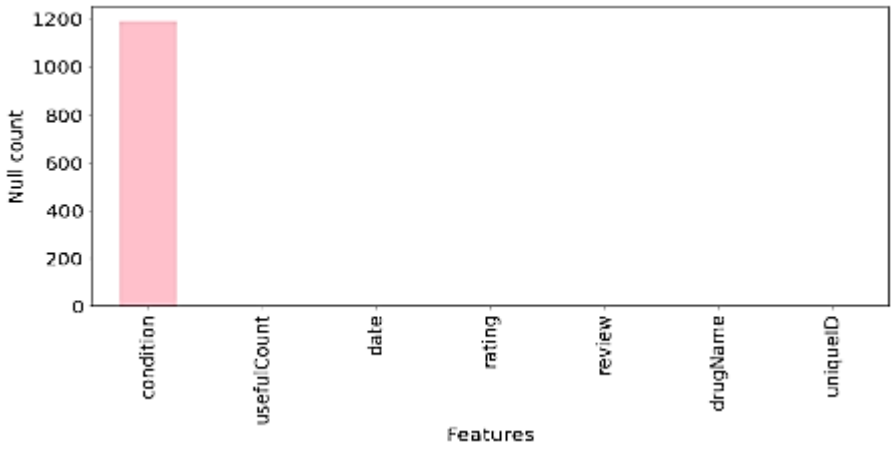

Fig 4.2 Bar Plot of The Number of Null Values Versus Attributes.

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

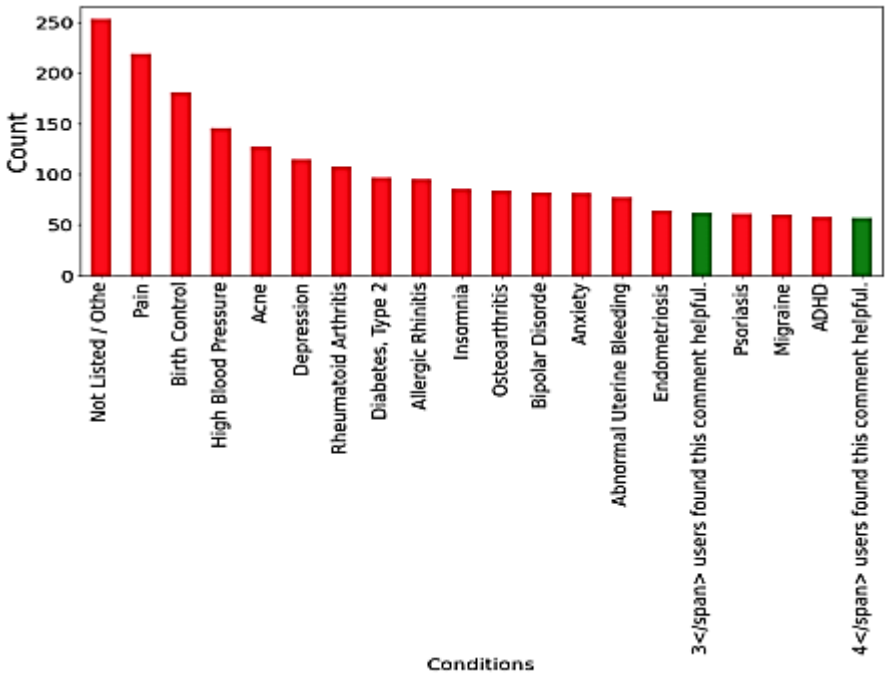

Fig 4.3 Bar plot of Top 20 conditions that has a maximum number of drugs available.

The visualization of value counts of the 10-star rating system. The rating beneath or equivalent to five featured with cyan tone otherwise blue tone. The vast majority pick four qualities; 10, 9, 1, 8, and 10 are more than twice the same number. It shows that the positive level is higher than the negative, and people's responses are polar. The condition and drug column were joined with review text because the condition and medication words also have predictive power.

Before proceeding to the feature extraction part, it is critical to clean up the review text before vectorization. This process is also known as text preprocessing. We first cleaned the reviews after removing HTML tags, punctuations, quotes, URLs, etc. The cleaned reviews were lowercased to avoid duplication, and tokenization was performed for converting the texts into small pieces called tokens. Additionally, stop words,

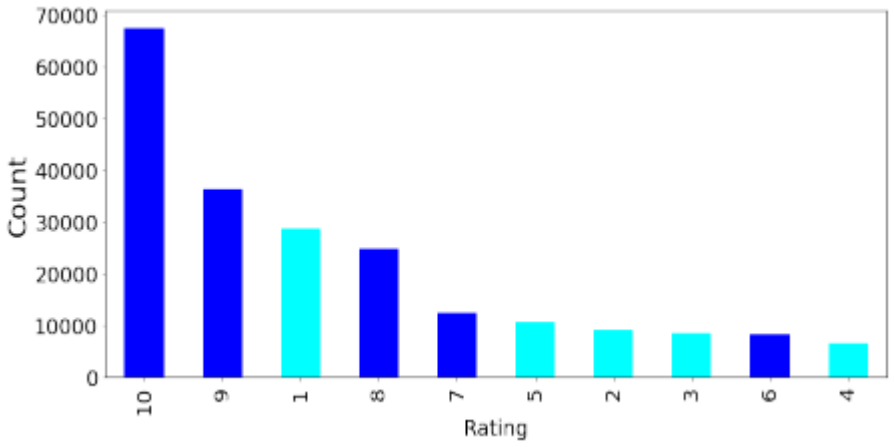

Fig 4.4 Bar plot of count of rating values versus 10 rating number.

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

B. Feature Extraction :

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

1) Bow: Bag of words is an algorithm used in natural language processing responsible for counting the number of times of all the tokens in review or document.

A term or token can be called one word (unigram), or any subjective number of words, n-grams. In this study, (1,2) n-gram range is chosen. Fig.3.5 outlines how unigrams, diagrams, and trigrams framed from a sentence. The Bow model experience a significant drawback, as it considers all the terms without contemplating how a few terms are exceptionally successive in the corpus, which in turn build a large matrix that is computationally expensive to train.

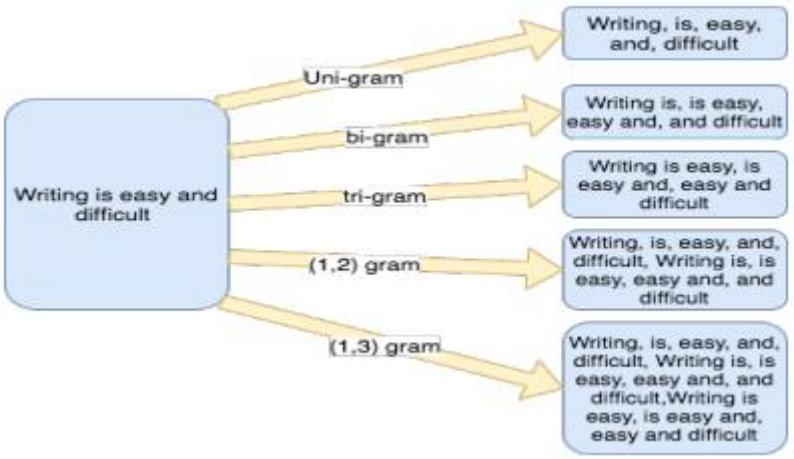

Fig 4.5 Comparison of various types of grams framed from a sentence.

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

$$tf(t,d) = log(1 + freq(t,d)) \qquad (1)$$

Inverse document frequency (IDF) is the opposite of the number of times a specific term showed up in the whole corpus. It catches how a specific term is document specific.

$$idf(t,d) = log(\frac{N}{count(d \epsilon D : t \epsilon d)}) \qquad (2)$$

TF-IDF is the multiplication of TF with IDF, suggesting how vital and relevant a word is in the document.

$$tfidf(t,d,D) = tf(t,d).idf(t,D) \qquad (3)$$

Like Bow, the selected n-gram range for TF-IDF in this work is (1,2).

3) Word2Vec: Even though TF and TF-IDF are famous vectorization methods used in different natural language preparing tasks,they disregard the semantic and syntactic likenesses between words. For instance, in both TF and TFIDF extraction methods, the words lovely and delightful are called two unique words in both TF and TF-IDF vectorization techniques although they are almost equivalents.

Word2Vec is a model used to produce word embedding. Word embeddings reproduced from gigantic corpora utilizing various deep learning models.Word2Vec takes an enormous corpus of text as an input and outputs a vector space, generally composed of hundred

dimensions.The fundamental thought was to take the semantic meaning of words and arrange vectors of words in vector space with the ultimate objective that words that share similar sense in the dataset are found close to one another in vectors space.

4) Manual Features: Feature engineering is a popular concept which helps to increase the accuracy of the model. We used fifteen features, which include use full count, the condition column which is label encoded using label encoder function from Scikit library, day, month, year features were developed from date column using Date Time function using pandas.

Text blob toolkit was used to extract the cleaned and uncleaned reviews polarity and added as features along with a total of 8 features generated from each of the text reviews as shown in Table 3.1.

C. Train Test Split:

We created four datasets using Bow, TF-IDF, Word2Vec, and manual features. These four datasets were split into 75% of training and 25% of testing. While splitting the data, we set an equal random state to ensure the same set of random numbers generated for the train test split of all four generated datasets.

Feature	Description
Punctuation	Counts the number of punctuation
Word	Counts the number of words
Stopwords	Counts the number of stopwords
Letter	Counts the number of letters
Unique	Counts the number of unique words
Average	Counts the mean length of words
Upper	Counts the uppercase words
Title	Counts the words present in title

Table 4.1 List of Features Extracted Manually from User Reviews.

D. Smote:

After the Train Test split, only the training data has undergone a synthetic minority over-sampling technique (Smote) to prevent the class imbalance problem. Smote is an oversampling technique that synthesized new data from existing data. Smote generates the new minority class data by linear interpolation of randomly selected minority instance 'a' in combination with its k nearest neighbor instance 'b' in the feature space. Table II shows the total distribution of data on final dataset i.e. after data cleaning.

It shows the projection of non-smote and smote using t-distributed stochastic neighbor embedding (t-SNE) of 1000 rows on manual features data. It displays that there are more orange points in the non-smote t-SNE projection, which represents the majority class dominance. It also shows that there has been an increment in blue points after using smote that brings out the balance between a majority and minority class that curbs the predominance of the majority class.

Smote	Class	Train (75%)	Test (25%)
No	Negative	47522	15841
	Positive	111583	37195
	Total	159105	53036
Yes	Negative	78108	15841
	Positive	111583	37195
	Total	189691	53036

Table 4.2 Dataset Distribution.

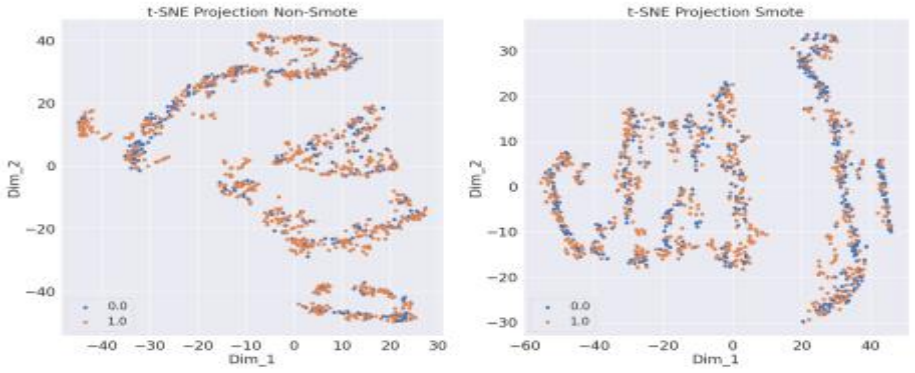

Fig. 4.6 T-SNE subplot before and after smote using 1000 training samples E.Classifiers.

Distinct machine-learning classification algorithms were used to build a classifier to predict the sentiment. Logistic Regression, Multinomial Naive Bayes, Stochastic gradient descent, Linear support vector classifier, Perceptron, and Ridge classifier experimented with the Bow, TF-IDF model since they are very sparse matrix and applying tree-based classifiers would be very time-consuming.

Applied Decision tree, Random Forest, LGBM, and Cat Boost classifier on Word2Vec and manual features model. A significant problem with this dataset is around 210K reviews, which takes substantial computational power. We selected those machine learning classification algorithms only that reduces the training time and give faster predictions.

E. Metrics:

The predicted sentiment were measured using five metrics, namely, precision (Prec), recall (Rec), f1score (F1), accuracy (Acc.) and AUC score [23]. Let the letter be: Tp = True positive or occurrences where model predicted the positive sentiment truly, Tn = True negative or occurrences where model predicted the negative class truly, Fp = False positive or occurrences where model predicted the positive class falsely, Fn = False negative or occurrences where model predicted the negative class falsely, Precision, recall, accuracy, and f1score shown in equations given below,

$$Precision = \frac{T_p}{T_p + F_p} \qquad (4)$$

$$Recall = \frac{T_p}{T_p + F_n} \quad (5)$$

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \quad (6)$$

$$F1score = 2.\frac{Precision.Recall}{Precision + Recall} \quad (7)$$

Area under curve (Auc) score helps distinguish a classifier's capacity to compare classes and utilized as a review of the region operating curve (roc) curve. Roc curve visualizes the relationship between true positive rate (Tpr) and false positive rate (Fpr) across various thresholds.

F. Drug Recommender system:

After assessing the metrics, all four best-predicted results were picked and joined together to produce the combined prediction. The merged results were then multiplied with normalized useful count to generate an overall score of drug of a particular condition. The higher the score, the better is the drug. The motivation behind the standardization of the useful count was looking at the distribution of useful count in one may analyze that the contrast among the least and most extreme is around 1300, considerable.

Moreover, the deviation is enormous, which is 36. The purpose behind is that the more medications individuals search for, the more individuals read the survey regardless of their review is positive or negative, which makes the useful count high. So while building the recommender system, we normalized useful count by conditions.

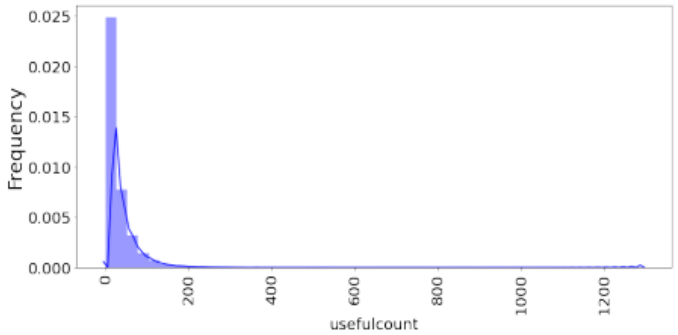

Fig. 4.7. Distribution of Useful Count.

CHAPTER 5 : IMPLEMENTATION

5.1 Explanation of Key Elements:

5.1.1 Python:

Guido Van Rossum designed Python, an interpreted, high-level, general-purpose programming language that was initially released in 1991.

The Python language has the following features:

- Simple to Understand and Apply Python is a simple language to learn and use. It's a high-level programming language that's helpful to developers.
- Expressive Language Python is more expressive than other languages, making it more intelligible and readable.
- Language Interpretation Python is an interpreted language, which means that the interpreter runs the code line by line. This makes debugging simple, making it suitable for novices.
- Language that is cross-platform Python can operate on a variety of platforms, including Windows, Linux, Unix, and Macintosh,

among others. As a result, we may conclude that Python is a portable language.

- Open Source and Free Software Python is a free programming language that can be downloaded from a secure internet address. It's also possible to get the source code. As a result, there is an open supply.
- Object-Oriented Programming Python aids the development of object-oriented languages and concepts such as classes and gadgets.
- Adaptable It means that other languages, such as C/C++, may be used to put the code together, and as a result, it can be utilised in our Python code.

5.2 Library Files

5.2.1 NUMPY:

NumPy is a Python library that includes support for enormous, multi-dimensional arrays and matrices, as well as a large set of high-level mathematical functions that may be applied to those arrays. Numeric, the forerunner to NumPy, was designed by Jim Hugunin with help from a number of other people. Travis Oliphant built NumPy in 2005 by heavily modifying Numeric and combining features from the competitor Numarray. NumPy is a large open-source software project with numerous contributors.

The Python programming language was not initially intended for numerical computing, but it quickly caught the attention of the clinical and engineering communities, prompting the formation of a special interest group known as matrix-sig in 1995 with the goal of defining an array computing bundle. Guido van Rossum, a Python clothier and maintainer, was one of its contributors, adding changes to Python's

grammar (especially the indexing syntax) to make array computation easier.

Jim Fulton completes a matrix package, which is later modified by Jim Hugunin to create Numeric, also known as Numerical Python extensions or NumPy. Hugunin, a PhD student at MIT, joined the Corporation for National Research Initiatives (CNRI) to work on JPython in 1997, leaving the maintainer position to Paul Dubois of Lawrence Livermore National Laboratory (LLNL). David Ascher, Konrad Hinsen, and Travis Oliphant were among the early members.

NumPy is a non-optimizing bytecode interpreter that targets the CPython Python reference implementation. Algorithms created for this version of Python are frequently much slower than their compiled counterparts. NumPy tackles the slowness issue in part by providing multidimensional arrays and capabilities and operators that work appropriately on arrays, which necessitates rewriting some code, generally inner loops, in order to use NumPy. NumPy arrays are used to store and perform information in the Python bindings of the widely used computer vision library OpenCV.

Because images with more than one channel are really stored as 3-dimensional arrays, indexing, slicing, and protecting with separate arrays are all highly eco-friendly ways to access specific pixels in a picture. The NumPy array, which is used as a standard statistical structure in OpenCV for pictures, extracted feature factors, kernel filtering, and other tasks, greatly simplifies the development process and debugging.

5.2.2 PANDAS:

Pandas is an open-source, BSD-certified library for the Python programming language that provides high-overall performance, easy-to-

use statistics systems, and data analysis tools. Pandas is a Python module that provides quick, flexible, and expressive facts structures for working with "relational" or "labelled" data in a clean and understandable manner. Its goal is to become the most important high-level building element for undertaking realistic, real-world international records evaluations in Python.

It also has the larger goal of being the most powerful and versatile open source data analysis/manipulation device available in any language. It is already well on its way to achieving this goal.

Pandas is well-suited to a wide range of statistical applications:

- Tabular statistics containing columns of varying types, such as those seen in a SQL database or an Excel spreadsheet
- Time collecting data that are sorted and unordered (but not necessarily at the same frequency).
- Row and column labels for arbitrary matrix information (homogeneously typed or heterogeneous)
- Any observational/statistical statistics set of any kind.

To be placed into a panda's facts form, the data does not need to be tagged in any way. Pandas' two basic statistics systems, Series (1-dimensional) and DataFrame (2-dimensional), address the vast majority of common use cases in finance, information, social science, and a wide range of engineering disciplines. Pandas is built on top of NumPy and is intended to work in conjunction with a variety of different third-party libraries in scientific computing.

5.2.3 MATPLOT LIB:

Matplotlib is a Python 2D plotting toolkit that generates book-quality figures in a variety of hardcopy codecs and interactive contexts. Matplotlib is a Python library that may be used in scripts, the Python and IPython shells, the Jupyter notebook, web applications servers, and four graphical user interface toolkits. Matplotlib aims to make both smooth and difficult tasks feasible. With just a few lines of code, you can create graphs, histograms, electrical spectra, bar charts, error charts, scatter plots, and more.

See the pattern plots and thumbnail galleries for samples. The pyplot package, when used with IPython, provides a MATLAB-like interface for convenient plotting. Through an object-oriented interface or a set of methods common to MATLAB users, you have complete control over line styles, font houses, axis houses, and so on for the electricity consumer.

5.2.4 SEABORN:

Seaborn is a data visualisation package for Python that is mostly based on matplotlib. It provides a high-level interface for creating visually beautiful and useful statistics graphs. Seaborn is a Python module for creating statistical visuals. It's based on matplotlib, and it's tightly integrated with pandas data systems. The goal of Seaborn is to make visualisation a major aspect of information exploration and understanding. Its dataset-oriented charting features operate on facts frames and arrays holding whole datasets, doing the necessary semantic mapping and statistical aggregation internally to provide relevant charts.

5.2.5 SCIKIT-LEARN:

To put it another way, sci-kit study is a free software system studying library for the Python computer language. It includes support vector machines, random forests, gradient boosting, k-method, and DBSCAN, among other categorization, regression, and clustering algorithms, and is designed to work with the Python numerical and clinical libraries NumPy and SciPy. Scikit-research was created in 2007 as a Google summers of code initiative by David Cornopean. Matthieu Brucher afterwards joined the challenge and began using it in his thesis paintings. INRIA was given consideration in 2010, and the initial public release (v0.1 beta) became released in late January 2010. The project presently has over 30 active participants and has received financial support from INRIA, Google, Tiny Clues, and the Python Software Foundation.

CHAPTER 6 :RESULTS AND DISCUSSIONS

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

Now-a-days new diseases are attacking human world and corona virus is such disease and this diseases require lots of medical systems and medical human experts and due to growing disease medical experts and systems are not sufficient and patients will take medicines on their risk which can cause serious death or serious damage to patient body.

To overcome from above problem author of this paper introducing sentiment and machine learning based drug recommendation system which will accept disease names from patient and then recommend DRUG and simultaneously display SENTIMENT rating based on reviews given by old users based on their experience. If predicted rating is high then patient can trust and took recommended drug.

In propose paper author has used various features extraction algorithms such as TF-IDF (term frequency – inverse document frequency), BAG of WORDS and WORVEC and this extracted features will be applied on various machine learning algorithm such as Logistic Regression, Linear SVC, Ridge classifier, Naïve Bayes, Multilayer Perceptron classifier, SGD classifier and many more. Among all algorithms TF-IDF is giving better performance so we are using TF-IDF features extraction algorithm with above mention algorithm.

To implement this project author has used DRUGREVIEW dataset from UCI machine learning website and below is the dataset screen shots

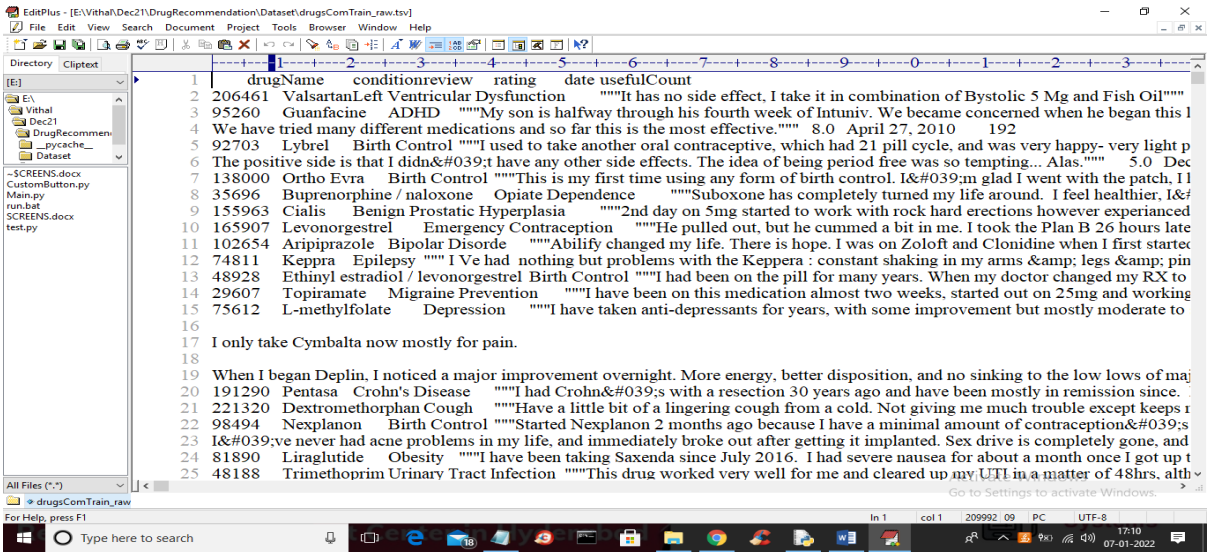

Fig 6.1 Dataset of Drugs, Condition and their Reviews and Ratings.

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

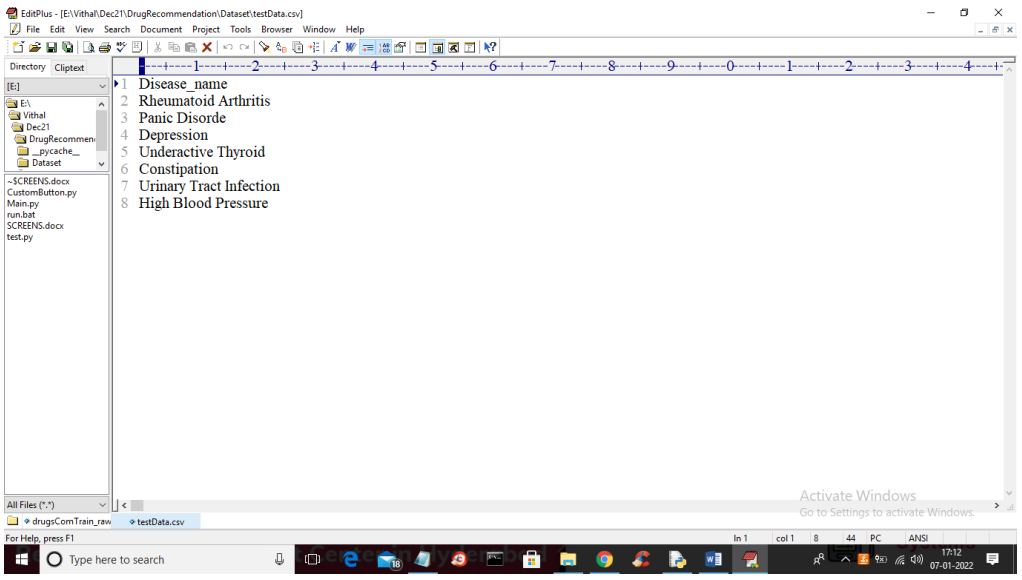

Fig 6.2 Test Data.

In above test data we have only disease name and machine learning will predict ratings and drug names. To implement this project we have designed following modules

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

5. Comparison Graph: using this module we will plot accuracy graph of each algorithm
6. Recommend Drug from Test Data: using this module we will upload disease name test data and then ML will predict drug name and ratings.

SCREENSHOTS :

To run project double click on 'run.bat' file to get below screen

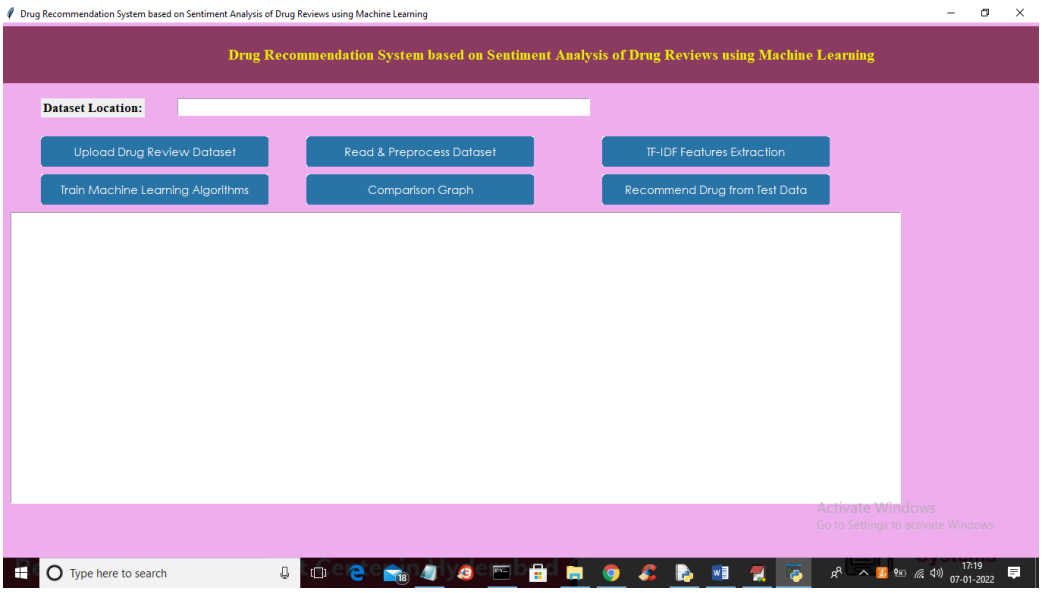

Fig 6.3 Application's GUI.

In above screen click on 'Upload Drug Review Dataset' button to upload dataset to application and to get below screen

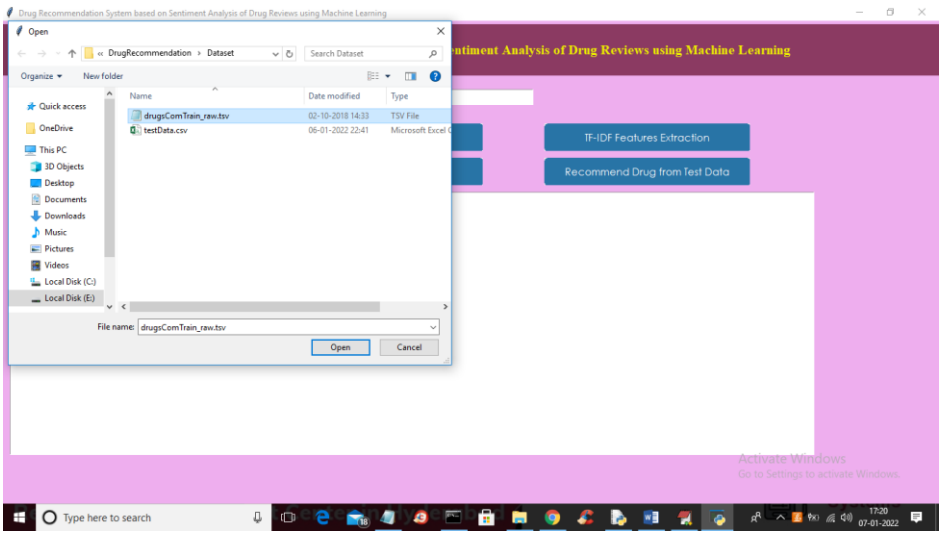

Fig 6.4 Uploading drug Review Dataset.

In above screen selecting and uploading DRUG dataset and then click on 'Open' button to load dataset and to get below screen

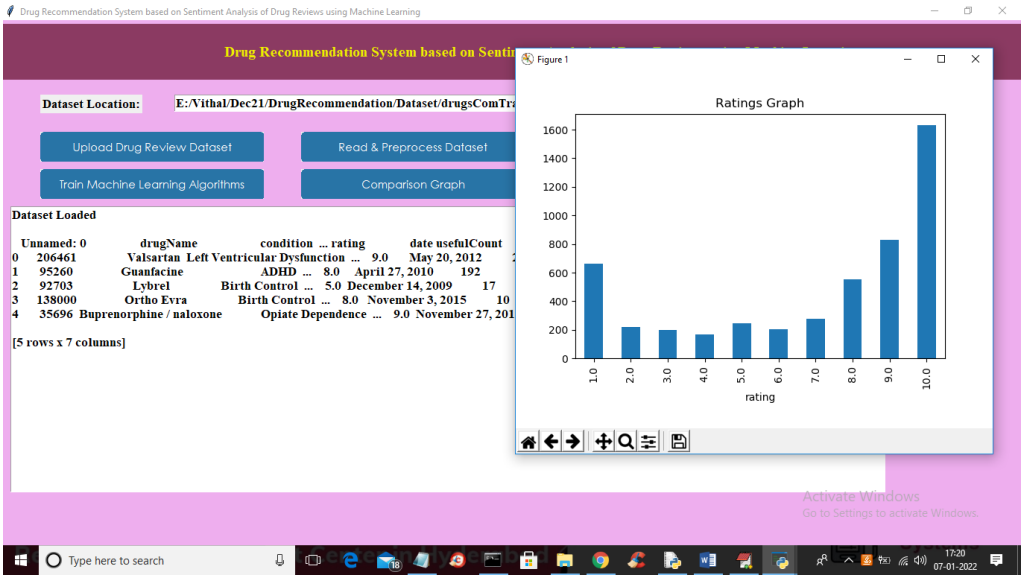

Fig 6.5 Rating Graph of Drugs.

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

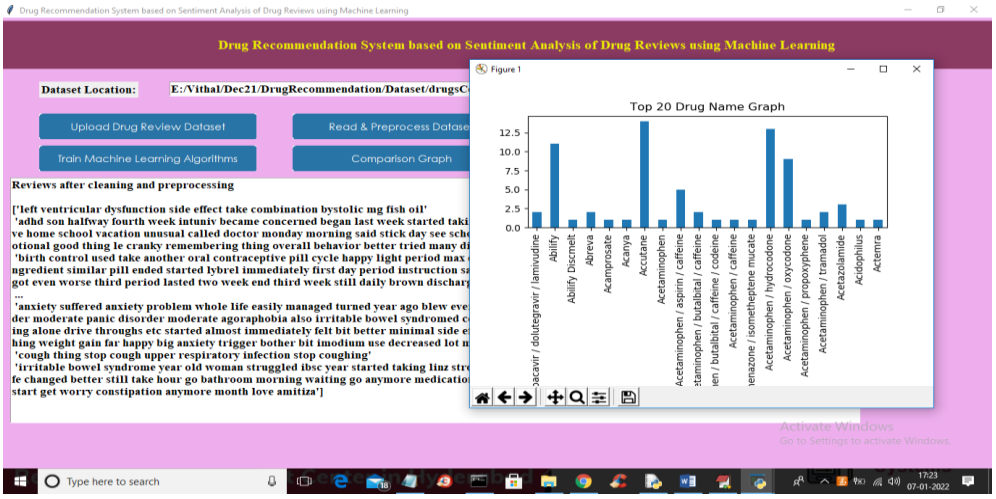

Fig 6.6 Graph Of Top 20 Drugs And Their Count.

In above screen we can see from all reviews stop words and special symbols are removed and in graph I am displaying TOP 20 medicines exist in dataset. In above graph x-axis represents drug name and y-axis represents its count. Now close above graph and then click on 'TF-IDF Features Extraction' button to convert all reviews in to average frequency vector

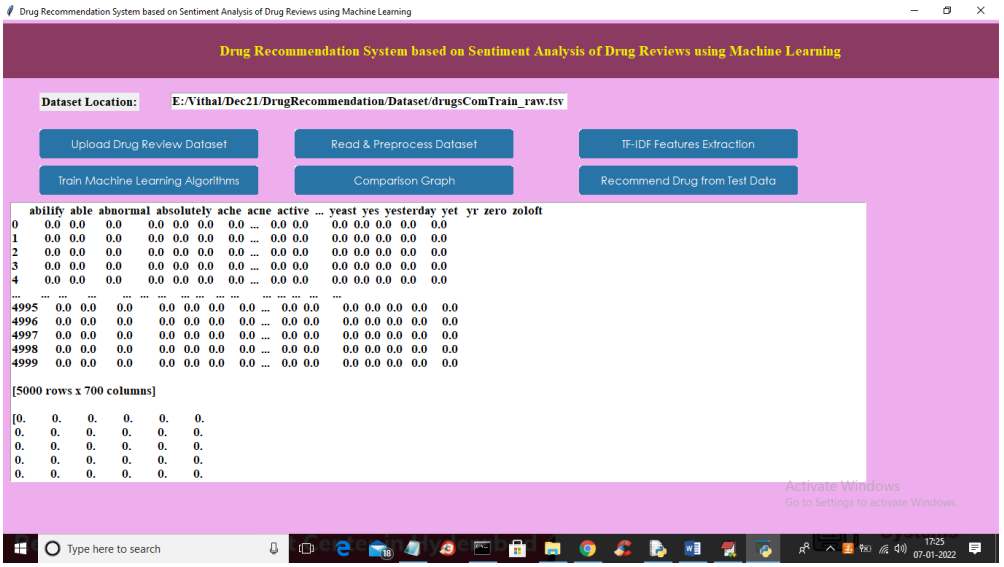

Fig 6.7 Reviews Converted into TF-IDF.

In above graph all reviews converted to TF-IDF vector where first row represents review WORDS and remaining columns will contains that word average frequency and if word not appear in review then 0 will put. Now scroll down above screen to view some non-zero frequency values

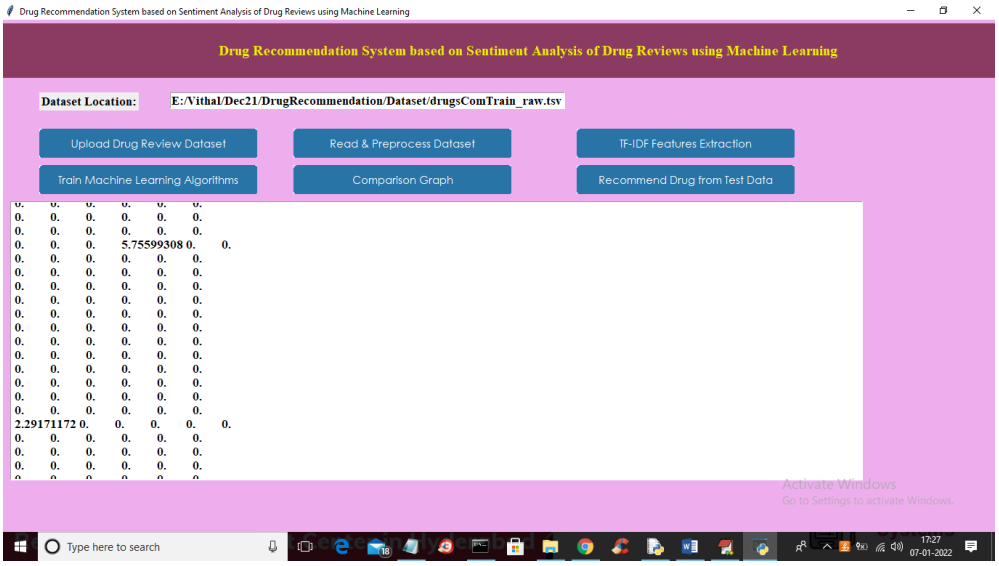

Fig 6.8 Non-Zero Average Frequency Values In TF-IDF.

In above screen you can see some columns contains non-zero average frequency values and now TF-IDF vector is ready and now click on 'Train Machine Learning Algorithm' button to train all algorithm and get below output

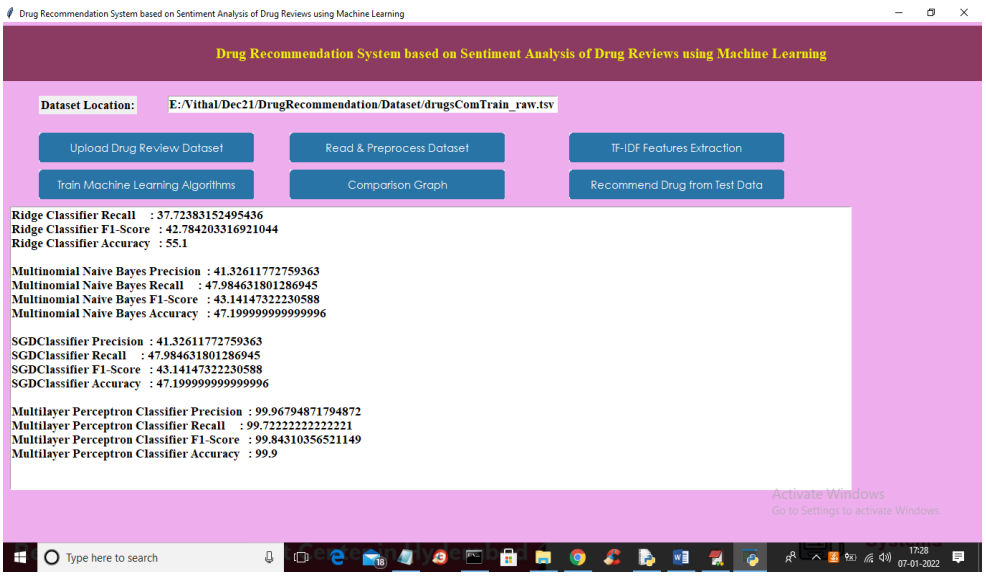

Fig 6.9 Performance of Drugs.

In above screen for each algorithm we calculate accuracy, precision, recall and FSCORE and in all algorithms MLP has got high performance and now click on 'Comparison Graph' button to get below graph

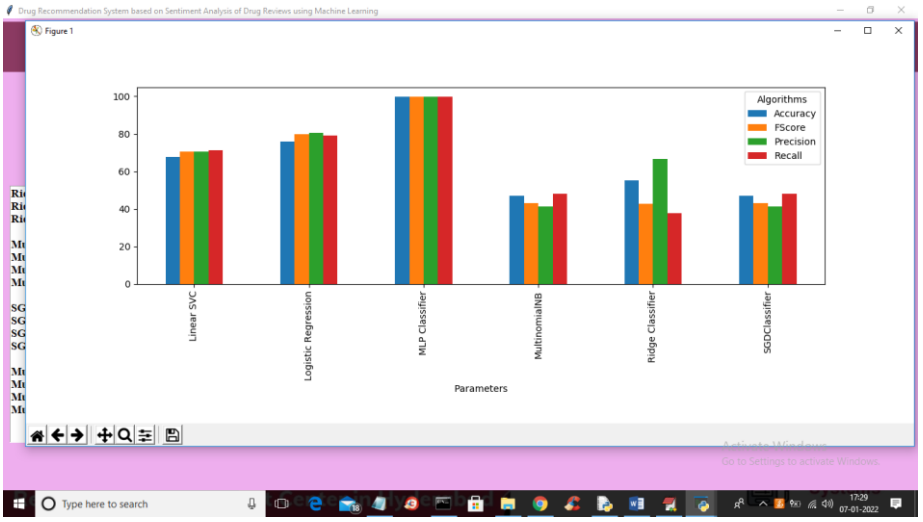

Fig 6.10 Comparison Graph.

In above graph x-axis represents algorithm name and y-axis represents accuracy, precision recall and FSCORE where each different colour bar will represents one metric and in above graph we can see MLP got high performance. Now close above graph and then click on 'Recommend Drug from Test Data' button to upload test data and to get predicted result as drug name and ratings.

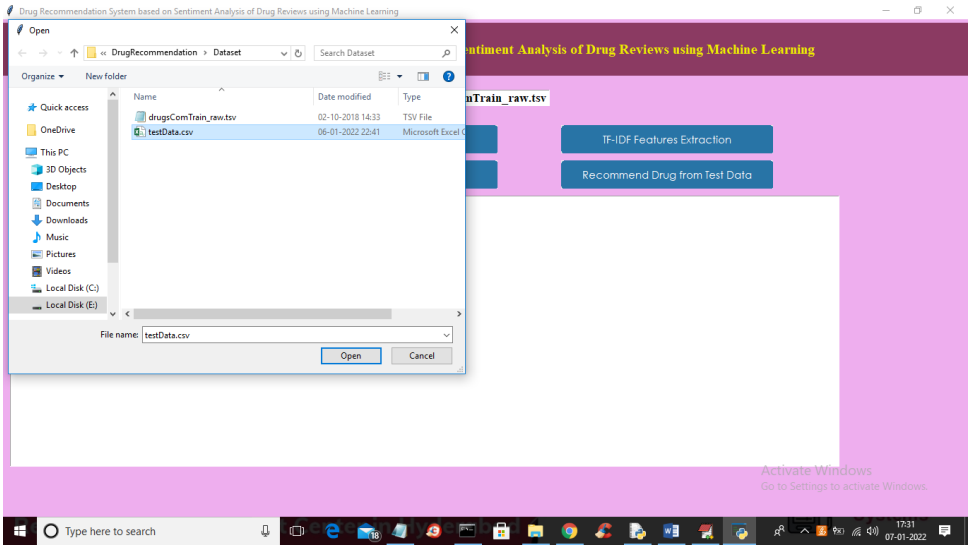

Fig 6.11 Uploading Test Data.

In above screen selecting and uploading 'testData.csv' file and then click on 'Open' button to load test data and get below prediction result

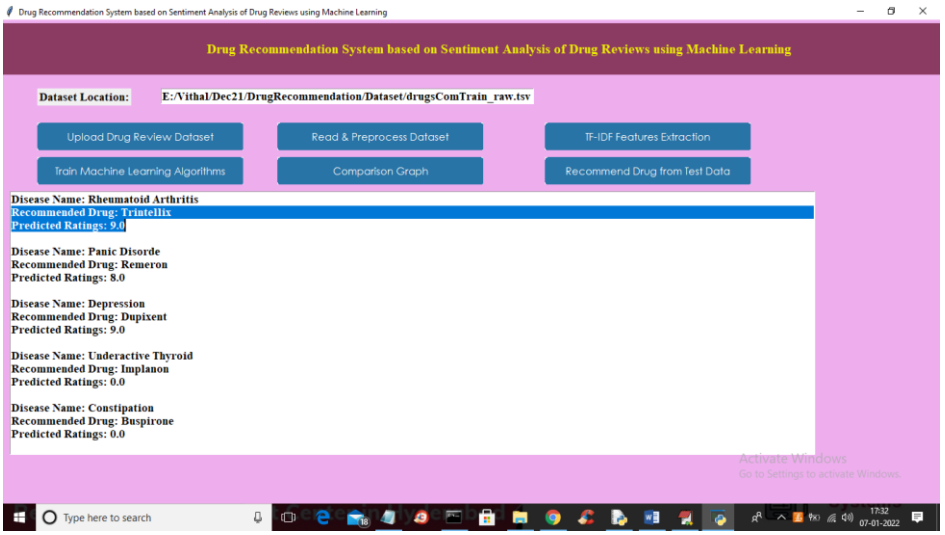

Fig 6.12 Final Output.

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

CHAPTER 7: CONCLUSION AND FUTURE SCOPE

CONCLUSION:

Reviews are becoming an integral part of our daily lives; whether go for shopping, purchase something online or go to some restaurant, we first check the reviews to make the right decisions. Motivated by this, in this research sentiment analysis of drug reviews was studied to build a recommender system using different types of machine learning classifiers, such as Logistic Regression, Perceptron, Multinomial Naive Bayes, Ridge classifier, Stochastic gradient descent, Linear SVC, applied on Bow, TF-IDF, and classifiers such as Decision Tree, Random Forest, Lgbm, and Cat boost were applied on Word2Vec and Manual features method. We evaluated them using five different metrics, precision, recall, f1score, accuracy, and AUC score, which reveal that the Linear SVC on TF-IDF outperforms all other models with 93% accuracy. On the other hand, the Decision tree classifier on Word2Vec showed the worst performance by achieving only 78% accuracy. We added best-predicted emotion values from each method, Perceptron on Bow (91%), Linear SVC on TF-IDF (93%), LGBM on Word2Vec (91%), Random Forest on manual features (88%), and multiply them by the normalized useful Count to get the overall score of the drug by condition to build a recommender system.

FUTURE SCOPE:

Future work involves comparison of different oversampling techniques, using different values of n-grams, and optimization of algorithms to improve the performance of the recommender system.

CHAPTER 8 : BIBILOGRAPHY

1. Wittich CM, Burkle CM, Lanier WL. Medication errors: an overview for clinicians. Mayo Clin Proc. 2014 Aug;89(8):1116-25.
2. CHEN, M. R., & WANG, H. F. (2013). The reason and prevention of hospital medication errors. Practical Journal of Clinical Medicine,
3. Fox, Susannah, and Maeve Duggan. ”Health online 2013. 2013.”
4. Bartlett JG, Dowell SF, Mandell LA, File TM Jr, Musher DM, Fine MJ. Practice guidelines for the management of community-acquired pneumonia in adults. Infectious Diseases Society of America. Clin Infect Dis. 2000 Aug; 31(2):347-82. Doi: 10.1086/313954. Epub 2000 Sep 7. PMID: 10987697; PMCID: PMC7109923.
5. Fox, Susannah & Duggan, Maeve. (2012). Health Online 2013. Pew Research Internet Project Report.
6. 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), Paralakhemundi, 2016, pp. 1471-1476, doi: 10.1109/SCOPES.2016.7955684.
7. Doulaverakis, C., Nikolaidis, G., Kleontas, A. et al. Galen OWL: Ontology-based drug recommendations discovery. J Biomed Semant 3, 14 (2012). doi.org/10.1186/2041-1480-3-14
8. Leilei Sun, Chuanren Liu, Chonghui Guo, Hui Xiong, and Yanming 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,

9. V. Goel, A. K. Gupta and N. Kumar, ”Sentiment Analysis of Multilingual Twitter Data using Natural Language Processing,” 2018 8th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2018, pp. 208-212, doi: 10.1109/CSNT.2018.8820254.
10. Shimada K, Takada H, Mitsuyama S, et al. Drug-recommendation system for patients with infectious diseases. AMIA Annu Symp Proc. 2005; 2005:1112.
11. Y. Bao and X. Jiang, ”An intelligent medicine recommender system framework,” 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), Hefei, 2016, pp. 1383-1388, doi: 10.1109/ICIEA.2016.7603801.
12. Zhang, Yin & Zhang, Dafang & Hassan, Mohammad & Alamri, Atif & Peng, Limei. (2014). CADRE: Cloud-Assisted Drug Recommendation Service for Online Pharmacies. Mobile Networks and Applications. 20. 348-355. 10.1007/s11036-014-0537-4.
13. 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), Vancouver.
14. Zhang, Yin & Jin, Rong & Zhou, Zhi-Hua. (2010). Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics. 1. 43-52. 10.1007/s13042-010-0001- 0.
15. J. Ramos et al., “Using tf-idf to determine word relevance in document queries,” in Proceedings of the first instructional conference on machine learning, vol. 242, pp. 133–142, Piscataway, NJ, 2003.

16. Yoav Goldberg and Omer Levy. Word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method, 2014; arXiv: 1402.3722.
17. Danushka Bollegala, Takanori Maehara and Kenichi Kawarabayashi. Unsupervised Cross-Domain Word Representation Learning, 2015; arXiv: 1505.07184.
18. van der Maaten, Laurens & Hinton, Geoffrey. (2008). Viualizing data using t-SNE. Journal of Machine Learning Research. 9. 2579-2605.
19. N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique, 2011, Journal Of Artificial Intelligence Research, Volume 16, pages 321-357, 2002; arXiv:1106.1813. DOI: 10.1613/jair.953.
20. Powers, David & Ailab,. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. J. Mach. Learn. Technol. 2. 2229-3981. 10.9735/2229-3981
21. Haibo He, Yang Bai, E. A. Garcia and Shutao Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, 2008, pp. 1322-1328, doi: 10.1109/IJCNN.2008.4633969.
22. Z. Wang, C. Wu, K. Zheng, X. Niu and X. Wang, "SMOTE Tomek Based Resampling for Personality Recognition," in IEEE Access