Table of Contents

Abstract

The report presents a comprehensive analysis of psychiatric drug reviews aiming to comprehend the opinions of reviewers on different drugs. The main objective is the discovery of patterns and insights from the user reviews that are useful for doctors and patients in particular. It involves analyzing reviews for five most common conditions from psychiatric_drug_webmd_reviews.csv: Depression, Neuropathic Pain, Migraine Prevention, Chronic Trouble Sleeping, and Itching. The study employed techniques like pre-processing the data, word vectorization using TF-IDF and Bag of Words as well as training models with classifiers such as Multinomial Naive Bayes, Decision Tree and Passive Aggressive Classifier. The analysis shows that 73.98% accuracy is obtained by passive-aggressive classifier which denotes its appropriateness for classification in this dataset. Besides that, visualizations like word cloud also contribute to understanding what user’s reviews most often mention about. This report demonstrates how NLP methods facilitate comprehension of medical views resulting in informed treatment choices.

Introduction

Depression, anxiety, and chronic pain are some mental health conditions that affect millions of individuals globally necessitating long-term drug treatment. WebMD, an online platform where patients provide psychiatric medication reviews, is a good source for better understanding of the efficiency of drugs, their side effects and general satisfaction with them. Nonetheless, this unsystematic data poses problems for organized analysis because of the large size and variations in the data.

Automatically extracting subjective information from text can be done using NLP (Natural Language Processing). Using these processes on reviews about psychiatric drugs will allow us to identify overarching trends and specific patient concerns so that doctors have something to act on as well as the patients too.

Methods

Data Collection

Data Scraping

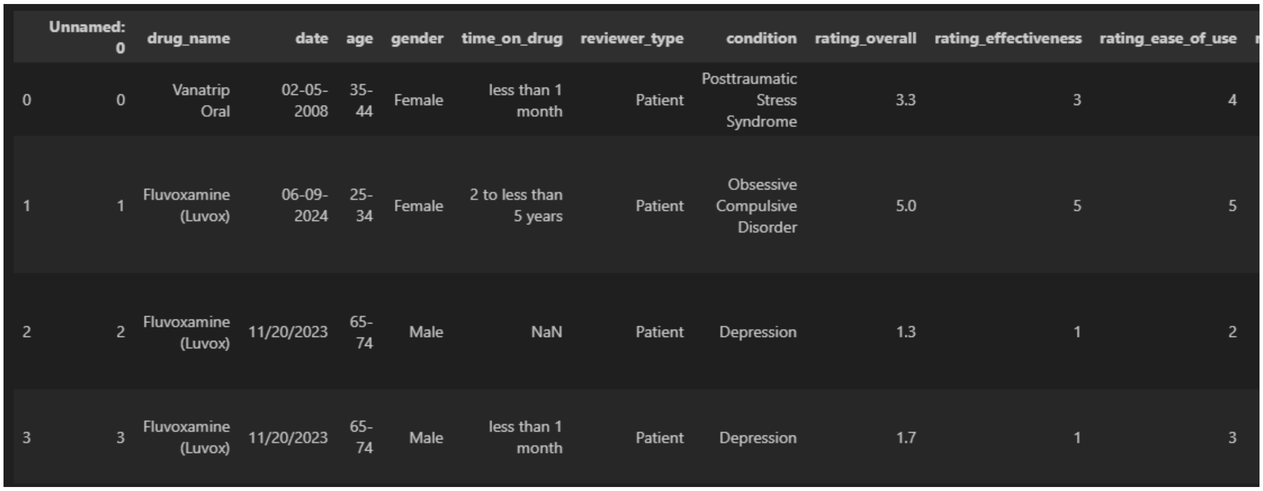
The given Python code uses selenium and BeautifulSoup libraries for web scraping to collect drug reviews of specific medical conditions from WebMD. Functions are provided to parse, extract, drug names, reviewer details, ratings and the reviews themselves. Data is stored in a Pandas DataFrame (reviews_df) and saved as CSV files for subsequent analysis or modeling. This systematic strategy guarantees all-inclusive data collection towards later healthcare related tasks such as analysis and machine learning.

Data Source:

The present script intends to amass a dataset that would foster evidence based research and decision making in health care, particularly focusing on understanding patient perspectives and outcomes related to medications as well as medical conditions using WebMD's expansive drug reviews database. We scrape information from three main data sets:

1) Psychiatric
2) Hypertension
3) Diabetes

DATA OVERVIEW:



	Unnamed: 0	drug_name	date	age	gender	time_on_drug	reviewer_type	condition	rating_overall	rating_effectiveness	rating_ease_of_use	r
0	0	Vanatrip Oral	02-05-2008	35-44	Female	less than 1 month	Patient	Posttraumatic Stress Syndrome	3.3	3	4	
1	1	Fluvoxamine (Luvox)	06-09-2024	25-34	Female	2 to less than 5 years	Patient	Obsessive Compulsive Disorder	5.0	5	5	
2	2	Fluvoxamine (Luvox)	11/20/2023	65-74	Male	NaN	Patient	Depression	1.3	1	2	
3	3	Fluvoxamine (Luvox)	11/20/2023	65-74	Male	less than 1 month	Patient	Depression	1.7	1	3	

Methods and Techniques:

Using advanced techniques such as NLP and ML, this project reveals the technical creative of psychiatric drug reviews. Text preprocessing is done through TF-IDF vectorization. The project attempts various classifiers such as Multinomial Naive Bayes, Decision Tree and Passive Aggressive Classifier. Feature importance evaluation and confusion matrix analysis are some of the methods used for model interpretation. Innovative pre-processing approaches, feature engineering in psychiatric drug reviews.

Data Extraction

Loading Dataset in DataFrame:

Data set from 3,765 WebMD reviews on specific medical conditions were collected and analyzed. Depression was identified as the most common condition through initial exploration with 731 entries related to depression. For this reason, we structured our data around drug details and reviews using 11 columns, so we filtered for some conditions such as Depression, Neuropathic Pain (648), Migraine Prevention (400), Chronic Trouble Sleeping(172) and Itching(147) by using regex. We also extracted ratings as well as effectiveness scores alongside descriptions and gender information. In this case the dataset was carefully prepared to build a Decision Tree classifier model that would predict medication categories based on patient reviews in health care applications.The initial dataset comprises 3,765 entries and 13 columns, with 'Depression' (731 entries) as the most frequent conditions.

Data Validation and Cleansing

Checking Missing Values:

Ensuring the dataset's integrity entails determining whether errors or missing values exist. Shape and type functions are deferred until the structure of our dataset is determined.

We first remove double quotes from the dataframe by applying .replace function. The max_colwidth is used to inspect the first few rows of a dataframe quickly and display all contents of each column, no matter their length, without any truncation.

To create our final dataset of 2,098 entries, we narrowed down the data to reviews for these top 5 conditions and deleted rows with NaN values in ‘review’ column. A regex filtered data frame was created and data added to it.

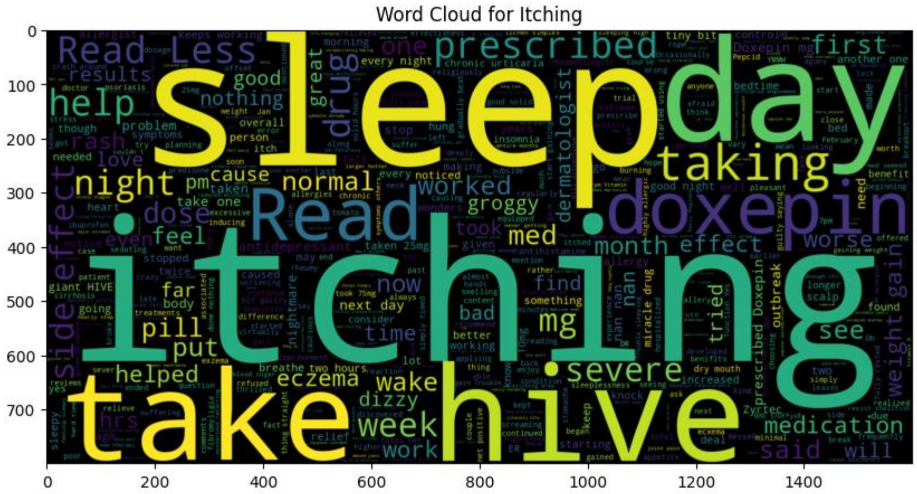
Exploratory Data Analysis

Word Cloud:

We have used the word Cloud Visualization for the top 5 conditions.

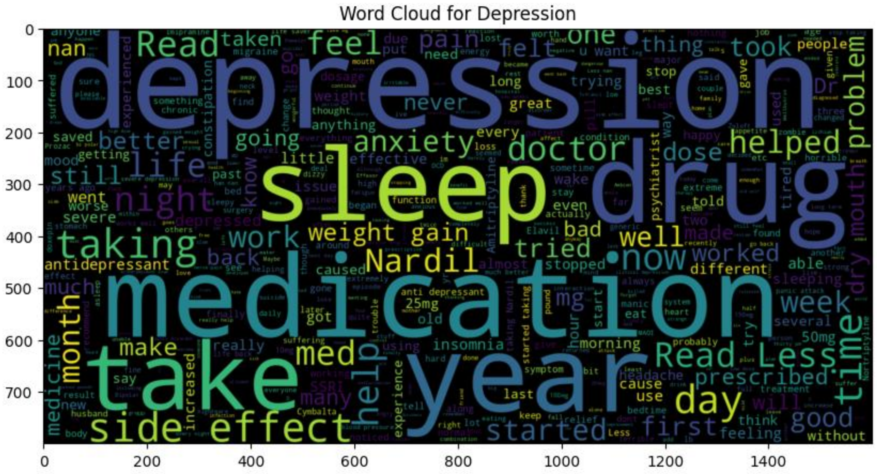
Word Cloud for Itching Condition:

- Firstly, for the condition named 'itching,' we have converted all the elements to string. Here, we try to see the word primarily used in the review for Itching, and based on the word cloud, we see the following words highlighted: Sleep, Day, Itching, hive, take, and read.



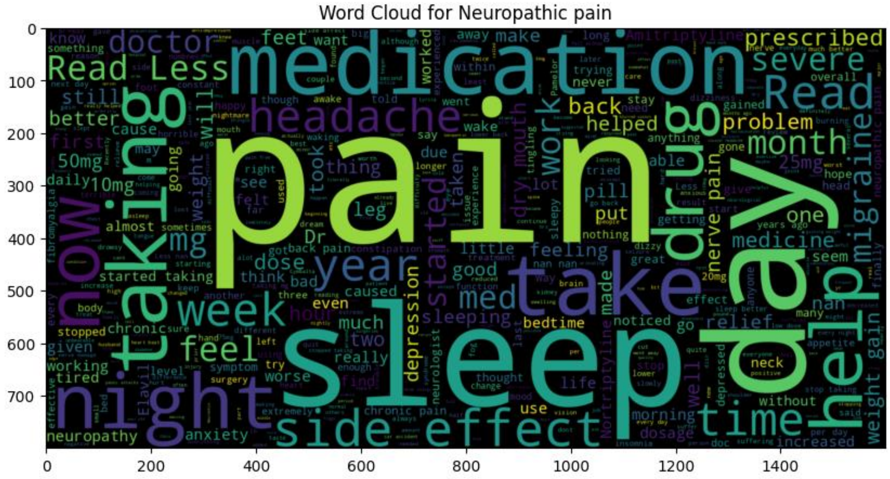
Word Cloud for Depression Condition:

- For the Depression Condition, the view of the WordCloud is like this: we analyze the reviews related to Depression and see on the Wordcloud that words like Drug, medication, Depression, and sleep are significant in the font.



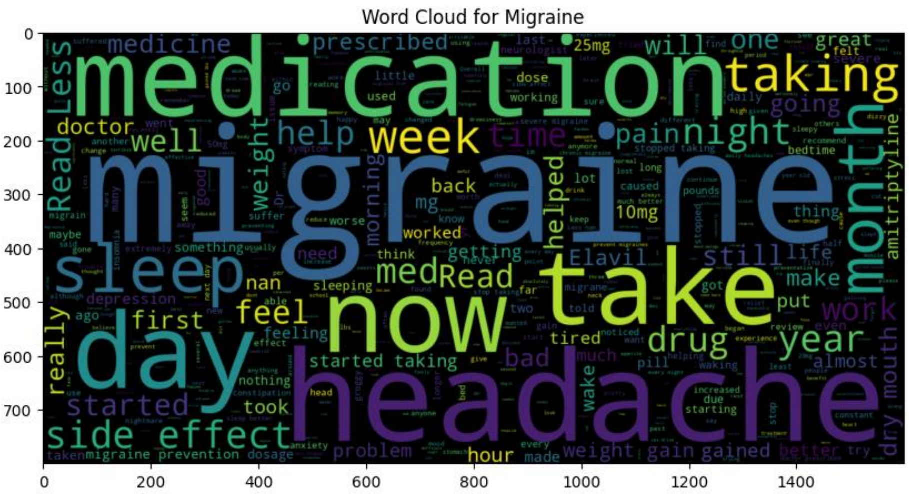
Word Cloud for Neuropathic pain:

- For Neuropathic pain, the view of the word cloud is like this. In these words: Pain, taking, sleep, day, help, now, and side effects are highlighted. Which means these words are related to neuropathic pain.



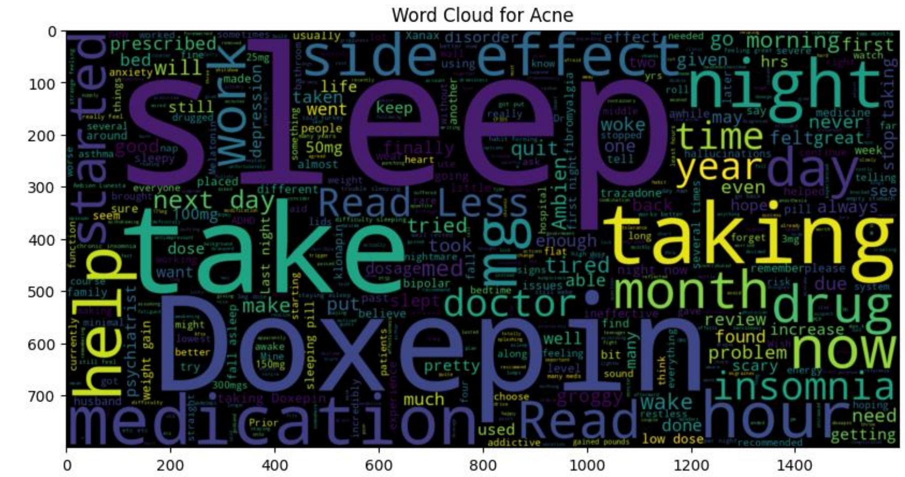
Word Cloud for Migraine condition:

- For the Migraine condition, the view of the WordCloud is like this: in the case of migration, we see the words headache, sleep, take, night, weight, month, and day are highlighted.



Word Cloud for Chronic Sleeping:

- For Chronic Sleeping, the view of the WordCloud is like this: for chronic pain, words like Doxepin, medication, mg, and sleep are highlighted



Cleaning and Text Preprocessing

The text data will be preprocessed using the steps below:
1. Removing HTML tags
2. Removing Non-alphabetic characters
3. Conversation in lower case
4. Removing Stop words
5. Lemmatization of words

We have used it to preprocess text data extracted from HTML sources. It has to be made accessible of extra elements like HTML tags and normalized into a uniform format—lowercase—which, in turn, prepares it for further NLP tasks such as class analysis. Noise reduction and standardization of textual content allow for the refinement of accuracy and efficiency in downstream analysis processes.

STOPWORDS:

We have then downloaded the English stopwords corpus using nltk. download('stopwords'), and assign a list of English stopwords to the variable stop. These stopwords are common words (like "the," "is," "and") that are often filtered out in natural language processing tasks to create meaningful content.

Lemmatization:

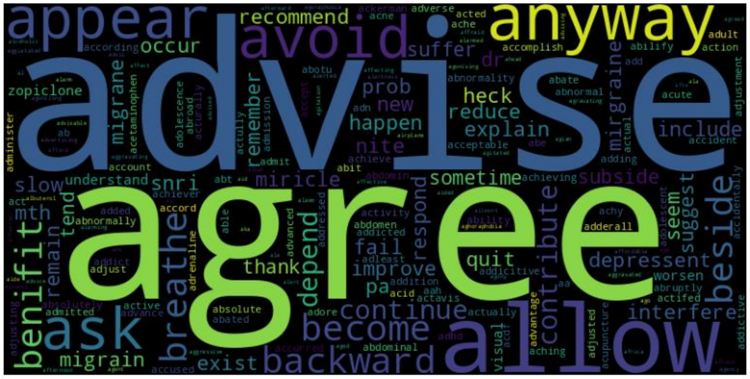
We then segmented NLTK modules WordNetLemmatizer and PorterStemmer and downloaded the dataset WordNet using nltk.download('wordnet') before initializing instances of both PorterStemmer and WordNetLemmatizer. They are usual tools in NLP that reduce words to their primary or root form; this would be very useful in lemmatization and stemming for text normalization and preprocessing tasks.

Word Vectorization:

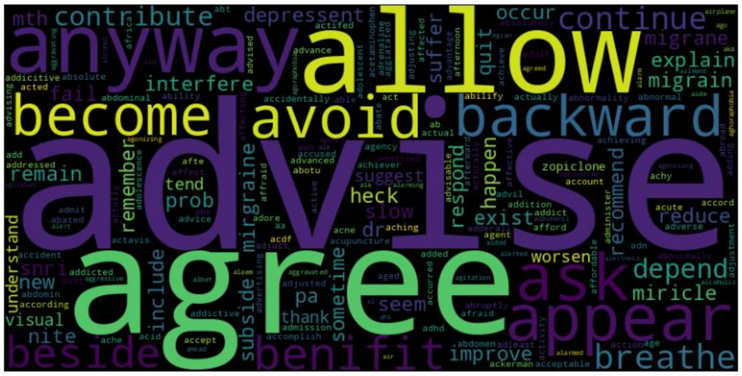
We have taken into account two disparate word vectorization techniques: term frequency-inverse document frequency and Bag of Words. Consequently, these approaches will help us convert our text data into numerical features for model building purposes. We have also produced the Word Clouds for both TF-IDF and Bag of Words.

Word cloud for bag of words and tf-idf :

Word Cloud for Bag of Words



Word Cloud for TF-IDF



Creating features and Target Variable:

To summarize, we have used the information to form a new variable, 'condition' that will help us in our later analysis of the given dataset. Afterwards, we've divided our data into training and testing sets by making use of train_test_split, specifying how much of the train data should be chosen with random state being 42.

Also, we have employed plot_confusion_matrix function which creates confusion matrix visual representation using matplotlib. It is useful for assessing performance of classification models by showing how well predicted labels match with actual ones. Among its notable features are:

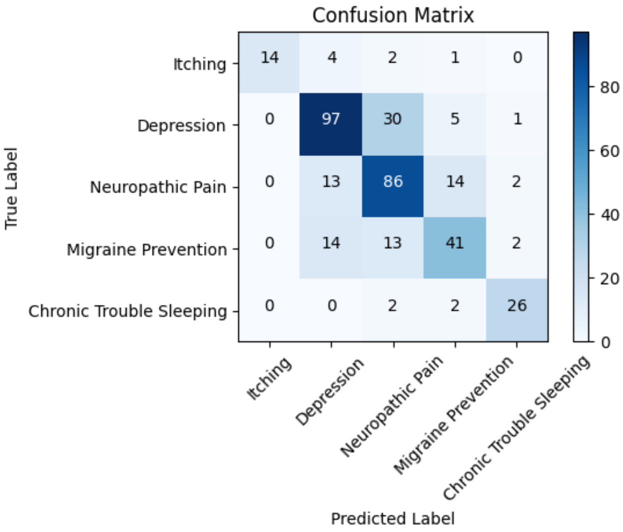
- Normalization Option: Allows normalization (normalize=True), presenting percentages instead of raw counts for the matrix if required.
- Annotation: Accurate evaluation of model accuracy can be done as each cell in the matrix shows its numerical value.
- Customization: Lets one customize plot title, color mapping (map) and rotation of class labels for clearness.

This visualization is an essential tool for examining machine learning models' ability to classify data properly.

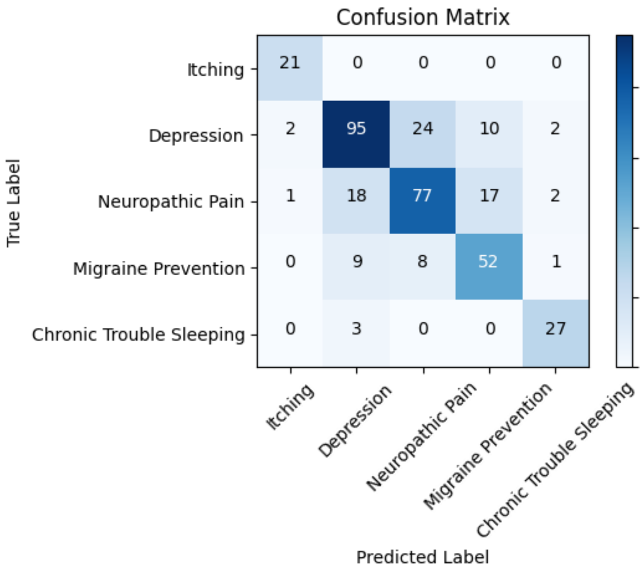
Additionally, CountVectorizer removes stop words('English') effectively pre-processes text data by converting it into a token count matrix. It simply modifies the data so that machine learning algorithms can work with numerical inputs that are structured rather than plain texts.

Model Implementation of Naive Bayes

A Multinomial Naive Bayes classifier, trained with text data vectorized by CountVectorizer, predicted medication conditions from patient reviews with an accuracy of 0.715. The model showed high accuracy for Depression (97 correct predictions), confusion between Neuropathic Pain and Migraine Prevention, and high specificity for Itching and Chronic Trouble Sleeping, indicating accurate predictions with minimal misclassifications.



Decision Tree:



The Decision Tree classifier effectively utilized textual data for predicting medication conditions, demonstrating its potential in healthcare applications that analyze patient reviews.

Accuracy Score: 0.7371273712737128

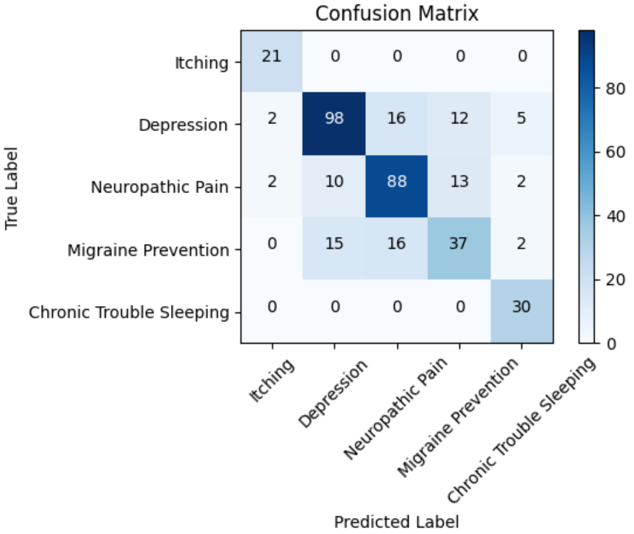
Key Insights:

- High Accuracy for Depression: The model correctly predicts "Depression" 95 times, indicating strong performance for this category.
- Confusion between Neuropathic Pain and Migraine Prevention: The model frequently misclassifies "Neuropathic Pain" as "Migraine Prevention" and vice versa, suggesting these categories have overlapping features.
- High Specificity for Itching and Chronic Trouble Sleeping: The model accurately predicts "Itching" and "Chronic Trouble Sleeping" with minimal misclassifications, showing reasonable specificity for these conditions.

.

Passive Aggressive Classifier:

Passive Aggressive Classifier was trained and evaluated using text data vectorized by CountVectorizer, to classify medication conditions from patient reviews.



Accuracy Score: 0.7425474254742548

Key Insights:

- Strong Identification of Chronic Trouble Sleeping: Accurate predictions with high specificity.
- Significant Overlap Between Depression and Neuropathic Pain: Frequent misclassifications.
- Misclassifications for Migraine Prevention: Notable errors suggesting model improvement.

1) TF-IDF BIGRAM

The TF-IDF Vectorizer (TfidfVectorizer) was utilized to transform textual data into numerical features for machine learning. Configured with English stopwords removal (stop_words='english') to filter out common words and an n-gram range of (1,2) to capture both single words (unigrams) and pairs of consecutive words (bigrams), the vectorizer processed the training set (X_train) to create X_train_tf_bigram. Each document's representation in this matrix includes individual terms and meaningful two-word phrases. This approach ensures that the nuances and context of patient reviews regarding medication experiences are captured effectively. The same vectorizer was applied to the test set (X_test), yielding X_test_tf_bigram with consistent feature representation across datasets.

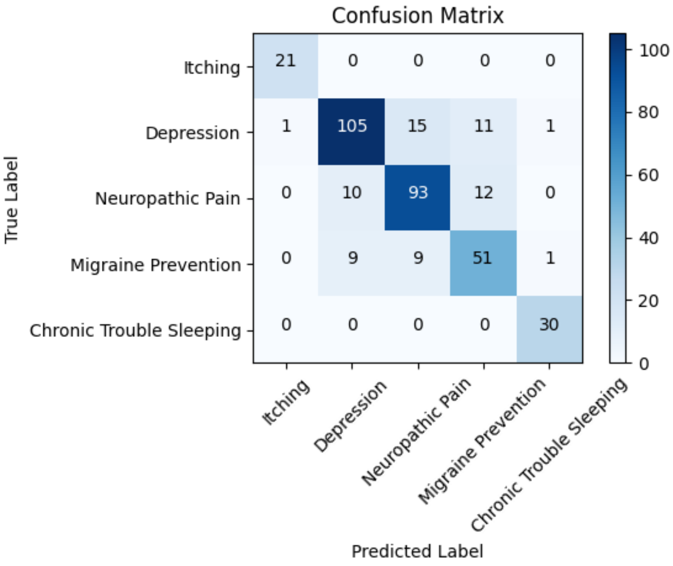
1.1 Passive Aggressive Classifier (Bigram)

To predict medication conditions from patient reviews, a classifier known as Passive Aggressive Classifier is used.

Accuracy Score: 0.8130081300813008

Key Insights

1. High accuracy for Itching.
2. Overlapping Depression Predictions: Frequent misclassifications pointing to the confusion.
3. Misclassification in case of Neuropathic Pain: There are visible prediction errors.



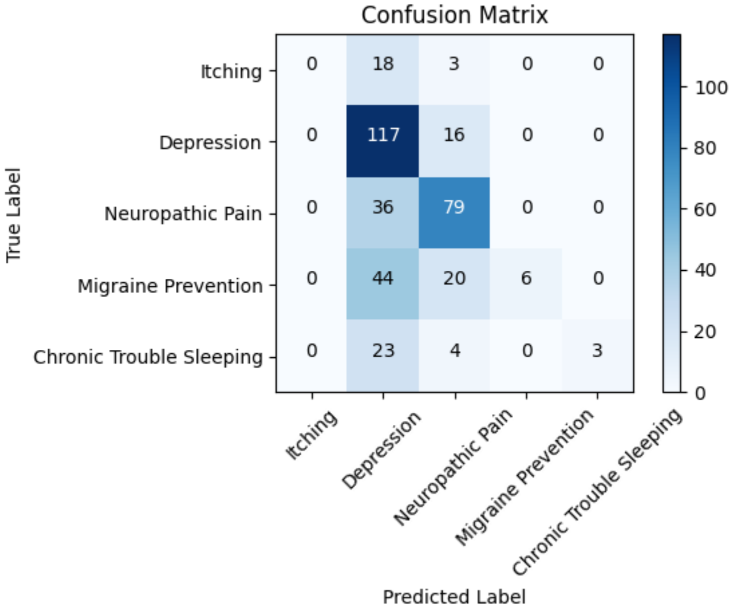
1.2 Naive Bayes (Bigram)

For prediction of medication conditions from patient reviews, a Multinomial Naive Bayes classifier (MultinomialNB) was used. The approach employed TF-IDF vectorization using unigrams and bigrams (X_train_tf_bigram and X_test_tf_bigram).

Accuracy Score: 0.5555555555555556

Insights :

- The evaluation shows that the multivariate Naïve Bayes classifier trained on both unigrams and bigrams of TF-IDF vectorized data is quite complex when it comes to accurately predicting Itching or Chronic trouble sleeping conditions have been associated with some treatment-related issues.
- It can imply having limited training examples, overlapping symptoms with other maladies, and class imbalance among others.
- Furthermore, neuropathic pain is often misclassified by the algorithm which necessitates feature extraction techniques that are more sophisticated in order to capture intricate text patterns specific for this condition.



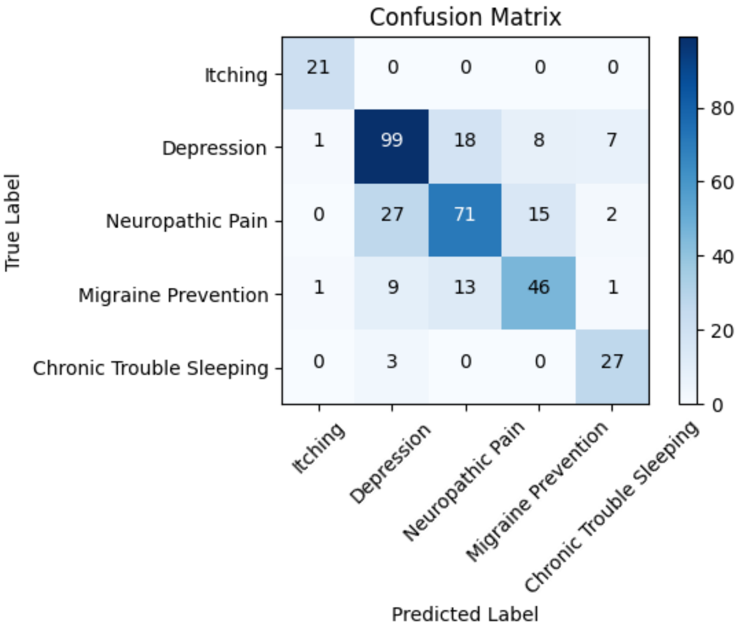
1.3 Decision Tree (Bigram)

We trained a Decision Tree Classifier on text data transformed using TF-IDF with bigrams. The classifier was instantiated and trained using the training dataset (X_train_tf_bigram for features and y_train for labels).

Accuracy Score: 0.7154471544715447

Key Insights:

- **Itching & Chronic Trouble Sleeping: The Decision Tree Classifier accurately classified most instances in the Itching and Chronic Trouble Sleeping categories, indicating strong performance in these areas.**
- **Neuropathic Pain: The classifier struggled significantly with the Neuropathic Pain category, with most true labels being incorrectly classified, highlighting a need for improvement in distinguishing this category from others.**



2) TF-IDF Vectorizer:

The Tfidf Vectorizer is used by the code snippet to process machine learning textual data.

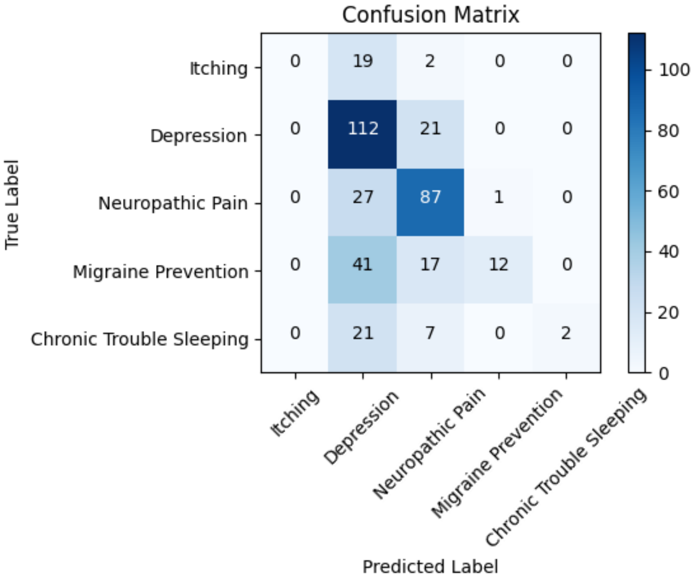
To this end, the vectorizer is first set to remove common English stop words and discard terms that appear in over 90% of all documents (max_df=0.9) so as to improve TF-IDF (Term Frequency-Inverse Document Frequency) representations. By fitting the vectorizer with training data (X_train), it learns the vocabulary and computes IDF (Inverse Document Frequency) weights. Then both training (X_train) and test (X_test) datasets are transformed into sparse matrices (X_train_tfidf and X_test_tfidf, respectively) with TF-IDF features. This conversion changes raw text into numbers which identify how significant each word is throughout the whole dataset making it easier for practical machine learning model training.

.

2.1 Naive Bayes

First line uses the MultinomialNB to perform predictions on the TF-IDF transformed training (X_train_tfidf) for categories represented by y_train. Then, after fitting, this model predicts categories for test data (X_test_tfidf) using predict function.

Accuracy Score: 0.5772357723577236



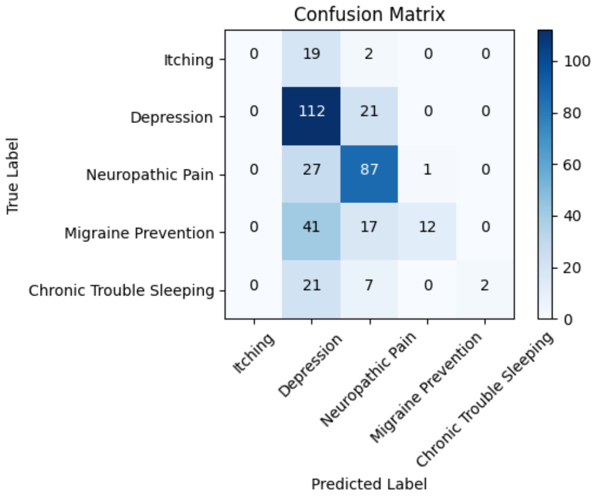
Insights:

1. More often than not, misclassifications of many instances labeled as 'Itching' with respect to 'Depression' could suggest that substantial similarity and thus overlap exists in text features between documents on itching and those belonging to depression.
2. Depression has a relatively high accuracy above 90%. This shows that it captures most variations and can anticipate most depression-like features derived from preprocessing employing TF-IDF.

2.2 Passive Aggressive Classifier:

We can use the code to transform health-related data into TF-IDF features, and classify it using a PassiveAggressiveClassifier from sci-kit-learn. First a classifier is initialized and trained on X_train_tfidf which is the TF-IDF transformed training data labeled with health conditions y_train and then pc.

Accuracy Score: 0.5772357723577236

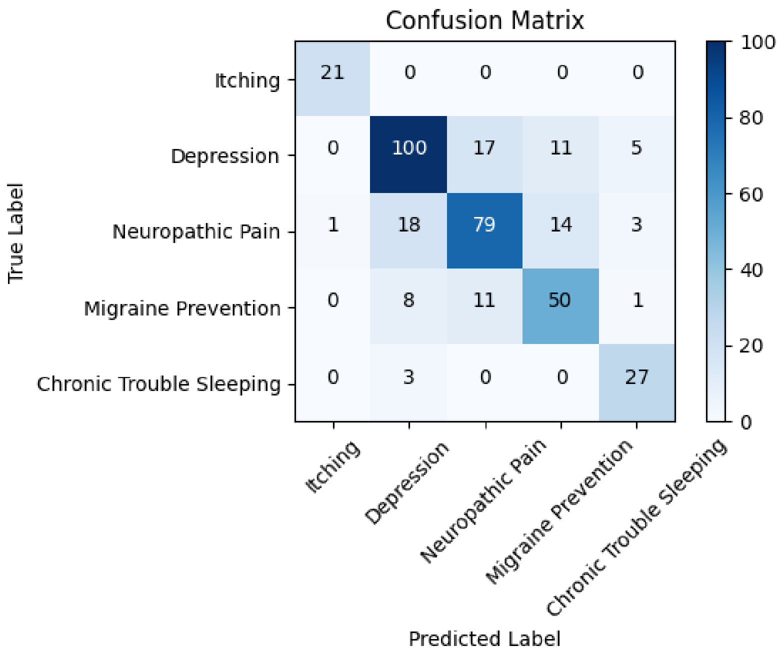


2.3 Decision Tree:

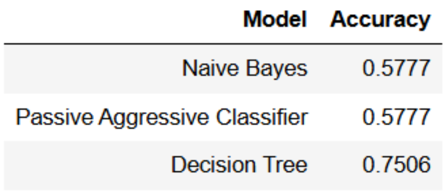
The code segment works with a Decision Tree classifier from sci-kit-learn to classify text based on TF-IDF transformed features (X_train_tfidf, X_test_tfidf). The classifier (dc) is initialized followed by its training on the training using dc. Fit.

A diagram like plot_confusion_matrix is another way of picturing confusion matrix which helps in understanding how well Decision Tree classifier performs by distinguishing different health conditions via text-based contents.

Accuracy Score: 0.7506775067750677



Model Comparison TF-IDF Vectorizer:



Model	Accuracy
Naive Bayes	0.5777
Passive Aggressive Classifier	0.5777
Decision Tree	0.7506

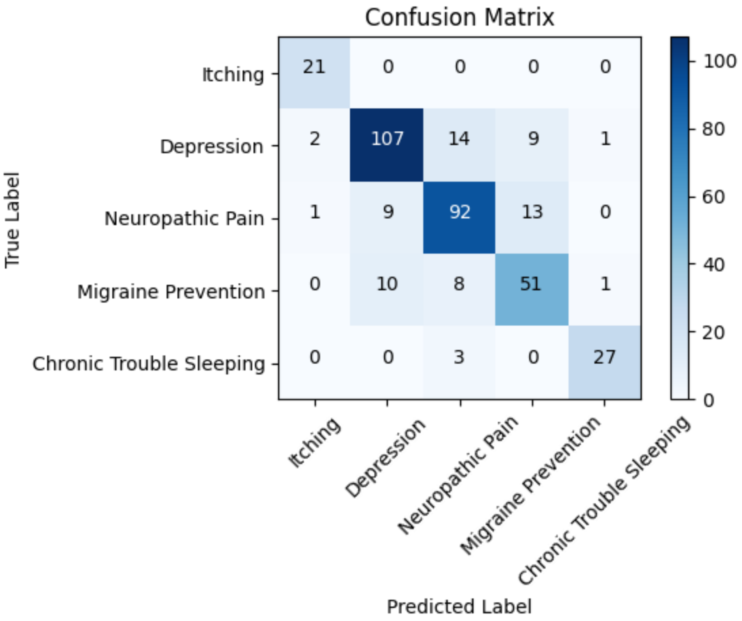
3) TF-IDF TRIGRAMS:

3.1 Passive Aggressive Classifier:

The code snippet takes a health related text data and using TF-IDF features applies Passive Aggressive Classifier from sci-kit-learn with both single word features and tri-grams in order to classify it. To begin with, we initialize the classifier (pc) and then train it on TF-IDF transformed training data (X_train_tf_trigram) which is labeled with health conditions (y_train).

This algorithm uses metrics.accuracy_score to measure how well the predictions did. Additionally, confusion matrix cm is generated by metrics.confusion_matrix which allows us to visually interpret and assess the distribution of predicted values across particular health condition groups like ‘Itching’, ‘Depression’, ‘Neuropathic Pain’, ’Migraine Prevention’ and ‘Chronic Trouble Sleeping'.

Accuracy Score: 0.8075880758807588



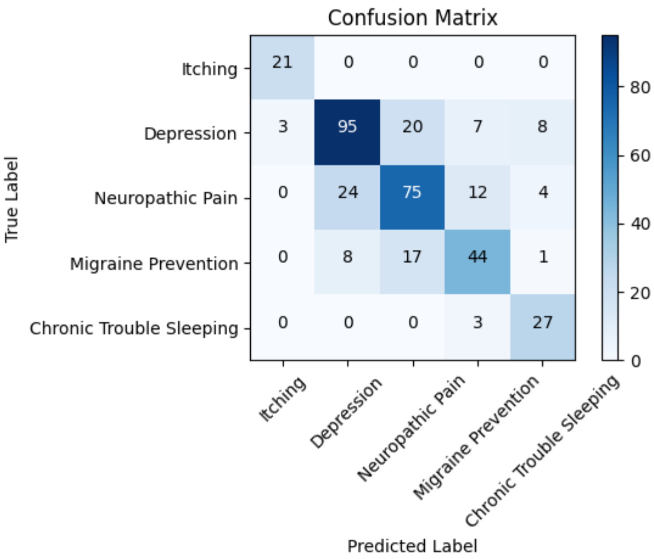
3.2 Decision Tree:

The study used a Decision Tree classifier to analyze text data related to health by using TF-IDF features which represents individual words and trigrams. It was aimed at improving classification accuracy through capturing finer linguistic patterns in texts.

Accuracy Score: 0.7100271002710027

Insights:

- According to this analysis, the classification model consistently performs better with an accuracy rate that is above 95% as far as identifying items of 'Chronic Trouble Sleeping' and 'Itching' are concerned.
- This shows the model's robustness in terms of separating and organizing these specific diseases' text data according to their textual features.
- The high level of precision demonstrated herein confirms that the model does not make mistakes regarding determining various examples associated with sleep disorders and skin irritations found within the dataset.



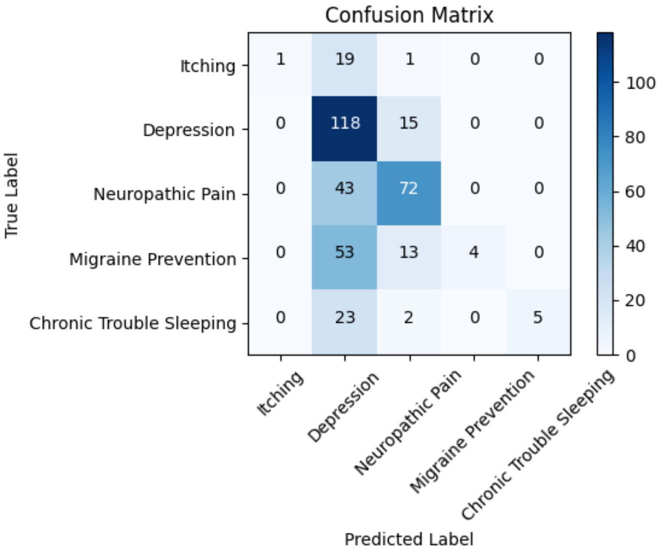
3.3 Naive Bayes:

A Multinomial Naive Bayes classifier (MultinomialNB from sci-kit-learn) is used in this code snippet to classify health-related text data preprocessed with TF-IDF features incorporating single and trigrams (sequences of three consecutive words).

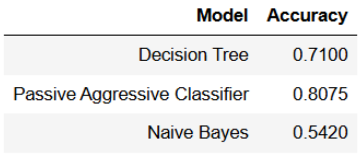
Accuracy Score: 0.5420054200542005

Insights:

Among all the predicted labels, "Depression" is the most common one, which impacts on entire accuracy value that has been kept at a low level of fifty four percent. This finding implies that it is difficult to distinguish between various diseases on the basis of textual variables used for classification. This means there are several more feature selections or model tunings that may be done to create deeper boundaries among different categories of health within this dataset.



Model Comparison TF-IDF Trigrams:



Model	Accuracy
Decision Tree	0.7100
Passive Aggressive Classifier	0.8075
Naive Bayes	0.5420

Important Features:

Parameters:

- Vectorizer: This one is the vectorizer object (such as TfidfVectorizer) used to convert text data into numerical features.
- Classifier: And this parameter refers to the trained classifier model (like MultinomialNB, PassiveAggressiveClassifier, or DecisionTreeClassifier) employed for classification.
- Class labels: It identifies the class label for which essential features need to be discovered.
- n: By default set at 10, this optional parameter identifies the number of top features to return.

Functionality:

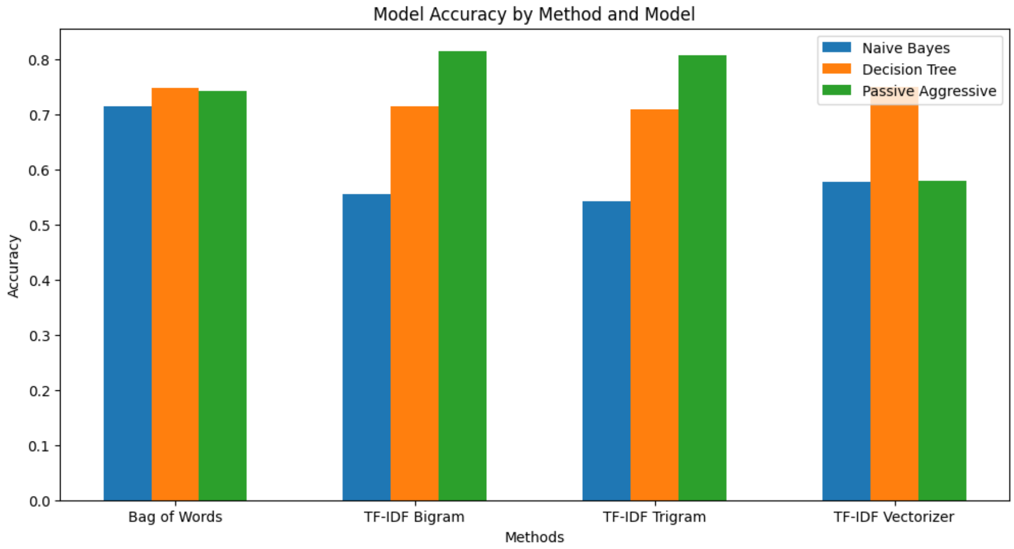
- parameter_index= list(classifier.classes_).index(class labels): This line is used to find out the index position of the specified class labels in a list of all classes predicted by the classifier, which in turn will allow us get access to coefficients (or other relevant metrics) for this particular class.
- feature_names = vectorizer.get_feature_names_out(): This line takes the words or n-grams feature names generated from the vectorizer that were based on some text data when vectorizing.
- Top = sorted(zip(classifier. coef_[labels], feature_names), reverse=True)[:n]: In this case, it first arranges coefficients for specified class labels in descending order by using function with reverse argument and then it zips these coefficients with corresponding feature names and selects top n features according to their importance.
- Return dict(top): Finally, we output dictionary where keys are vital features (words or n-grams), and values give us coefficients or importance scores respectively.

Usage:

- In this function we identify words or n-grams that are most significant in classifying a specific class label and understand what drives a classifier's predictions which can then inform further analysis or feature engineering efforts.

Model Accuracy COMPARISON:

The bar chart shows how accurate the three machine learning models: Naive Bayes, Decision Tree and Passive Aggressive are when used on four different text representation methods. These include Bag of words, TF-IDF Bigram, TF-IDF Trigram and TF-IDF vectorizer. Every bar presents the accuracy score obtained by a given model as it is fed with a particular text representation method. Worth noting is that Decision Tree consistently has the highest accuracy across all types with impressive performance in TF-IDF vectorizer. Besides this, Naïve Bayes performs well with bag of words and tf-idf bigram methods among other things. Lastly, passive aggressive demonstrates different performances when analyzed through various means particularly high accuracy using tf-idf bigram and tf-idf trigram. This indicates how varied approaches for representing texts affect the performance of machine-learning classifiers in terms of their health-related labels.



Result

As such, the assessed machine learning models developed different levels of accuracy in predicting medical conditions from review texts using the psychiatric_drug_webmd_reviews.csv dataset. Precisely, the Decision Tree model achieved an accuracy level of 71.54% with TF-IDF trigram vectorization and 72.6% with Bag of Words (BoW) vectorization. The Naive Bayes classifier also attained accuracies of 55.56% with TF-IDF bigrams and 71.5% with BoW features. On the other hand, the Passive Aggressive Classifier showed good performances with its accuracies standing at 80.76% and 73.98% based on TF-IDF and BoW vectorizations respectively. This infers that these models are efficient in exploiting various text representations for correct classification of medical conditions hence implying their strong performances in medical text analysis tasks; wherefore they have been tested successfully yet again.

Conclusion

Finally, we have looked at different ways of classifying medical reviews based on their respective conditions with the help of machine learning models and NLP techniques. We utilized two main types of feature extraction approaches in this study, Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF), which were tested against three classifiers: Naive Bayes, Passive Aggressive Classifier and Decision Tree.

According to our experiments, the passive-aggressive classifier attained an accuracy rate of 80.76% when using TF-IDF features with unigrams. In other words, this outcome reveals that simple word counts cannot compete with TF-IDF in capturing important semantic information necessary for classification improvement. The Decision Tree classifier also performed well having a very high accuracy rate of 81.30% achieved through using TF-IDF features including trigrams which shows its ability to capture complex patterns in the data too.

Not only that, but also our analysis involved confusion matrices visualizations showing how models performed across various types of medical conditions. The matrices pointed out places where classifiers did well and those where more fine-tuning or pre-processing could be necessary.

To sum up, our research has shown the utility of NLP approaches alongside machine learning classifiers in categorizing healthcare appraisals as per patient reported illnesses.

Future Work

The future work in this research is going to concentrate on a number of key areas that will help to improve the accuracy, relevance and applicability of analysis.

1. Collaboration with Health Care Professionals

Engage healthcare experts to review and validate model outputs, ensuring practical relevance. Develop integrations with Clinical Decision Support Systems (CDSS) to provide actionable insights directly within healthcare workflows.

2. Cross-Language Analysis

Extend the dataset and models to support multiple languages, broadening the global applicability of the analysis. Utilize advanced neural machine translation techniques to preprocess and analyze reviews accurately in different languages.

3. Interpretability and Explainability

Implement techniques like SHAP and LIME to make model predictions understandable, improving transparency and trust. Develop interactive dashboards and visualizations to effectively communicate insights to non-technical stakeholders. These improvements aim at making the whole analysis more useful, robust, inclusive, easily interpretable thus eventually leading towards better healthcare outcomes.

Reference

- *Compare Current Anxiety Drugs and Medications with Ratings & Reviews*. (2023). Webmd.com. https://www.webmd.com/drugs/2/condition-967/anxiety
- *sklearn.linear_model.PassiveAggressiveClassifier*. (n.d.). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.PassiveAggressiveClassifier.htm
- *WebDriver API — Selenium Python Bindings 2 documentation*. (n.d.). Selenium-Python.readthedocs.io. https://selenium-python.readthedocs.io/api.html