uOttawa

Sentiment Analysis on Drug Reviews

Graduate Project (CSI 6900)

Supervised By: **Verena Kantere**

Submitted By:

Reshma Sri Challa (300071545)

rchal050@uottawa.ca

Abstract: Online review websites contain abundance of information based on user choices and experiences over multiple products. Using data mining techniques like sentiment analysis on this information, we can obtain beneficial vision. In this project we analyze online user reviews within the pharmaceutical domain. Online user reviews in this field contain data relevant to different aspects of drugs like side effects and effectiveness, which build automatic analysis. However, analyzing sentiments on various drug reviews can obtain awareness, help take decisions and promote supervise public health on experience. In this work we perform multiple tasks over drug reviews with data acquired by web scraping online pharmaceutical review websites. We will implement sentiment analysis to predict the sentiments regarding side effects, overall satisfaction and effectiveness of user reviews on distinct drugs. We further examine the transferability of trained classification models among data sources.

Keywords: Web scraping, Sentiment Analysis.

1 Introduction

Unstructured data like images, text, videos comprise a fortune of information. But, due to the complication in processing and analyzing this data, people frequently hold back from spending additional time and effort in taking a chance out from structured datasets to examine these unstructured resources of data, which can be a hidden gold mine. Natural Language Processing (NLP) is all about leveraging tools, practices and procedures to process and understand natural language-based data, which is normally unstructured. Sentiment analysis is one of the extremely widespread applications of NLP, focus on analyzing sentiments of various datasets varying from corporate surveys to movie reviews. The key characteristic of sentiment analysis is to analyze a body of text for understanding the opinion conveyed by it. Normally, we measure the sentiment with polarity which is of positive or negative significance.

Sentiment Analysis is a very challenging problem, as user produced content is portrayed in different complicated ways using natural language. In sentimental analysis, the majority of the researchers worked on general domains such as restaurants, movies, social media and product reviews but not significantly on medical domains. Pharmaceutical product safety presently relies on clinical trials and certain test procedures. Such type of studies is normally done under identical conditions in a specific number of test subjects within a limited time period. Therefore, the conflicts in patient selection and treatment conditions can have substantial impact on the effectiveness and possible risks of adverse drug reactions. Therefore, post-advertising drug surveillance, plays a key role regarding drug safety once a drug has been released. And, patients using drugs are frequently looking for experiences from patients like them on the internet which they cannot at all times find amongst their family and friends. Few studies exploring the influence of social media on patients have shown

that for some health problems, online community support results in a positive effect.

In this project we study the possibility to apply sentiment analysis, techniques and workflows which can be leveraged to extract useful insights from drug reviews, and exploiting its reviews, we identify the polarity of effectiveness of a drug and its side effects. Additionally, to tackle the challenges associated to the limited data availability, we examine the transferability of the trained models across data sources.

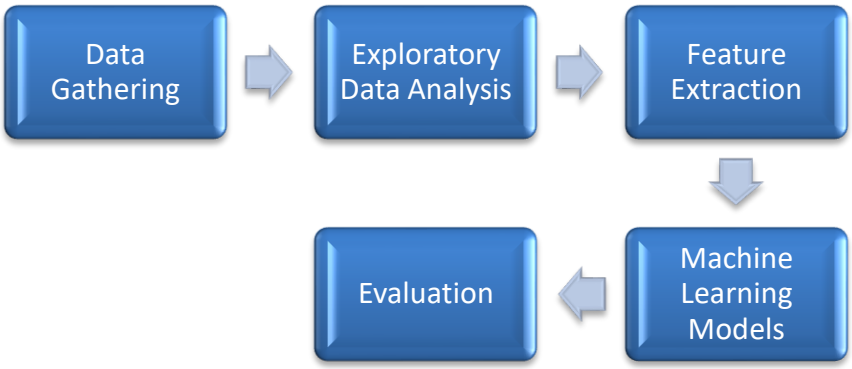

Figure 1: Process Framework

2 Dataset

2.1 Web Scrapping

The data is collected from two separate webpages for recovery of user reviews and ratings on drug experience. Drugs.com presents user reviews on particular drugs along with associated condition and a 10-star user rating indicating overall user satisfaction. Druglib.com contains significantly fewer reviews but reviews and ratings are presented in a more organized way. Reviews are grouped into reports on the three aspects which are benefits, side effects and overall comment. Moreover, ratings are accessible regarding overall satisfaction comparatively to Drugs.com as well as a 5-step side effect rating, varying from no side effects to extremely severe side effects and a 5-step effectiveness rating varying from ineffective to very effective. We collected user comments and ratings from both pages by Web scraping the data. The data was scraped from raw HTML using the Beautiful Soup library in Python. Analyzing the links, we infer that the URL for each drug review is common for all drugs except that the last part is the drugs name. We have first extracted the list of drugs, and then generating the list of URL for each drug and used requests (python module) to fetch this URL. Inspected the HTML code for each attribute to

scrape from the page and saved it into a csv for each domain. From Drugs.com we gathered the reviews and ratings, from Druglib.com we gathered reviews, ratings, effectiveness reviews, side-effects reviews, side-effects from the URL. Crawling these domains resulted in two data sets containing around 250000 reviews from Drugs.com and around 4000 reviews from Druglib.com. Additionally, we developed three level polarity labels for overall patient satisfaction and three level effectiveness and side effect grades using limits as stated in the table 1. This data was be further split into 75% for training and 25% for testing.

2.2 Exploratory Data Analysis

Figures 2-5 depict the percentage distribution of each type of Effectiveness, side effects, ratings respectively mentioned by the user for both websites. Figures 6-13 depict the class distribution and percentage for each type for both drugs.com and druglib.com.

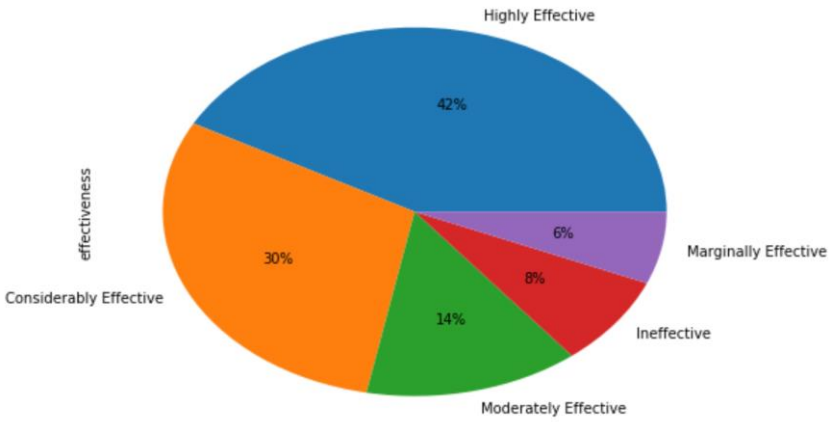

Figure 2: Drglib.com Effectivenes

Table 1: Data Description.

Data	Polarity	Label
Overall Rating	rating <= 4	0
	4 < rating < 7	1
	Rating >=7	2
Effectiveness	Ineffective	0

	Marginally/Moderately Effective	1
	Considerably/Highly Effective	2
Side Effects	No Side Effects	2
	Mild/Moderate Side Effects	1
	Severe/Extremely Severe Side Effects	0

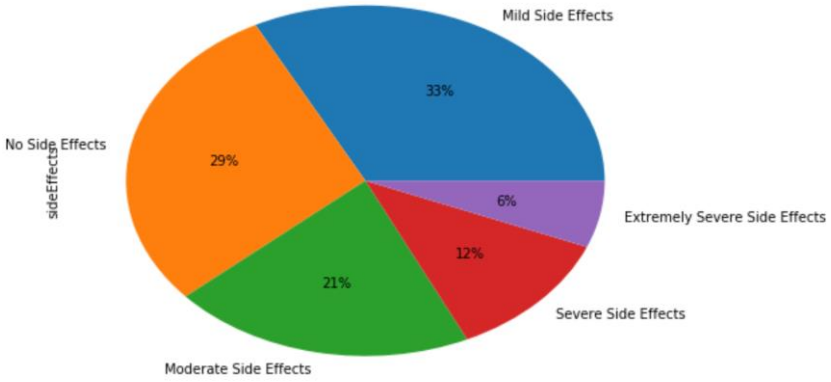

Figure 3: Drglib.com Side Effects

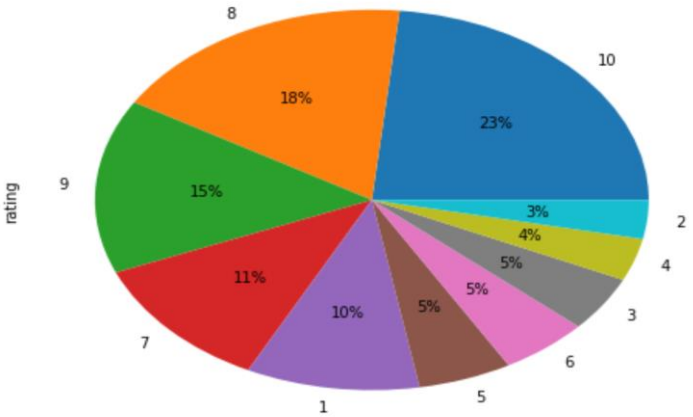

Figure 4: Drglib.com Rating

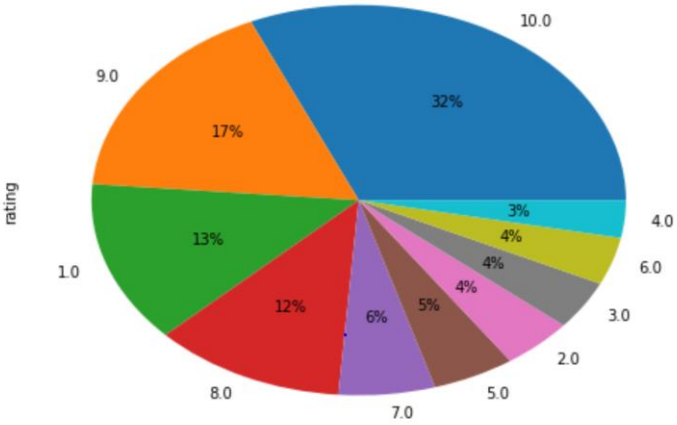

Figure 5: Drugs.com Rating

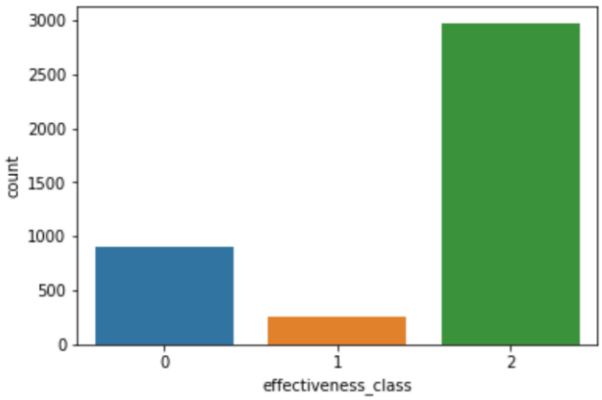

Figure 6: Drglib.com Effectivness class label distribution

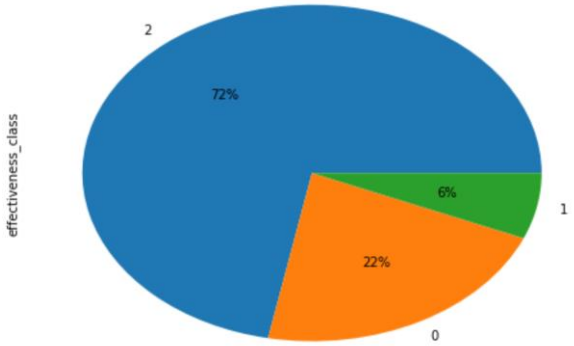

Figure 7: Drglib.com Effectivness class label percentage

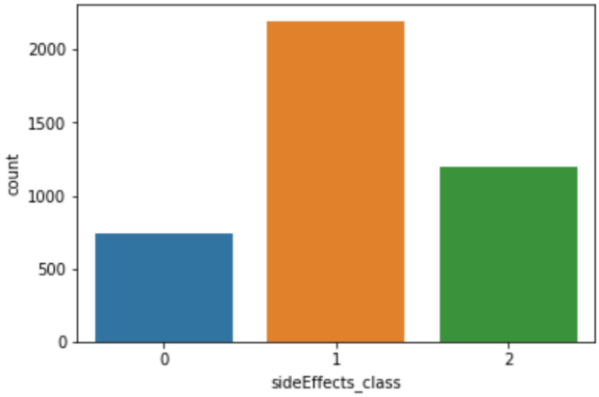

Figure 8: Drglib.com Sideeffects class label distribution

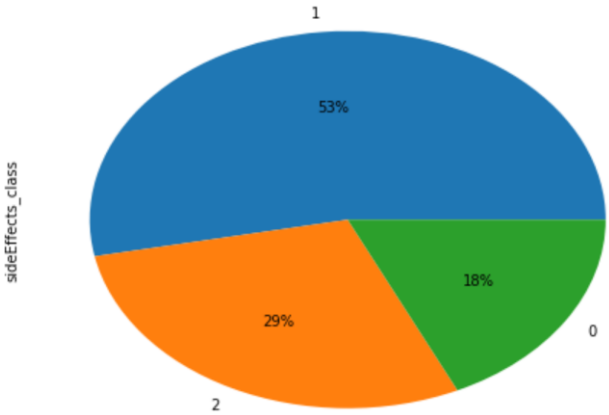

Figure 9: Drglib.com Sideeffects class label percentage

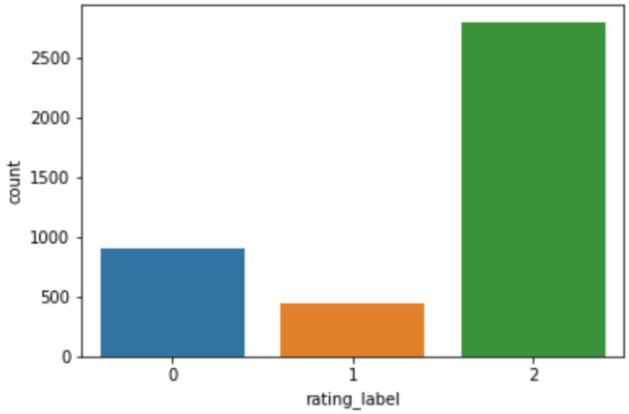

Figure 10: Drglib.com Rating class label distribution

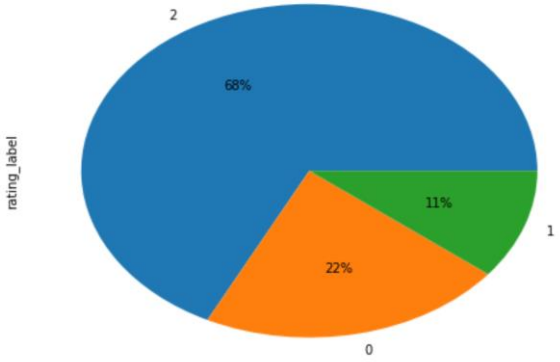

Figure 11: Druglib.com Rating class label percentage

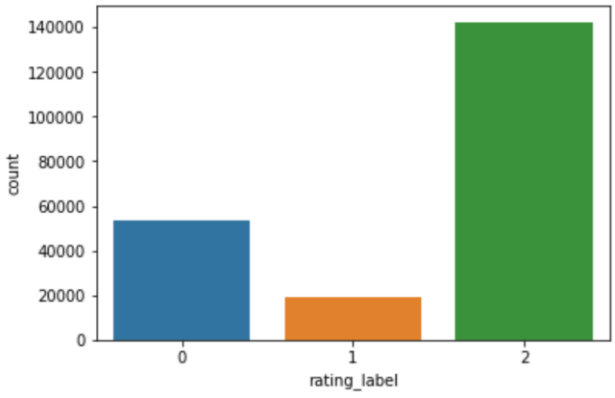

Figure 12: Drugs.com Rating class label distribution

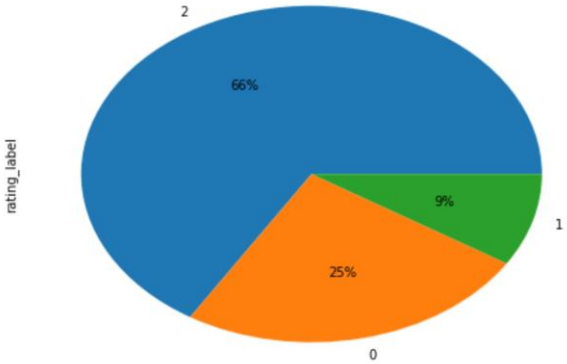

Figure 13: Drugs.com Rating class label percentage

Word Cloud is a data visualization technique utilized for depicting text data, where the size of each word signifies its frequency. Critical textual data points can be emphasized using a word cloud. Word clouds are widely used for analyzing data from social networking websites, here we use it on drug data. The disadvantage of word clouds is they are not suitable for all situations. We represent the highly positive reviews and highly negative reviews in separate word cloud for effectiveness, side effects, ratings in drugslib.com and rating for drugs.com data. Refer to figures 14-25 for this information.

Frequency Distribution reveals the frequency of each vocabulary item in the text. It is called "distribution" as it shows how the total number of word tokens in the text are distributed across the vocabulary items. We automatically identify the words of a text that are most informative about the topic and genre of the text. Here we use frequency distribution to find the top most 50 words in our drug reviews for the highly positive reviews and highly negative reviews in separate frequency distribution for effectiveness, side effects, rating in drugslib.com and rating for drugs.com data. Refer to figures 26-29 for this information.

Throughout the project the drugslib.com reviews used for effectives class are effectives reviews, for side effects class are side effect reviews and for ratings are all reviews combined that is effectiveness reviews plus side effects reviews plus reviews and the drugs.com reviews are used for rating. The most frequently used words are taking, effects, medication, first, would, could, severe, experienced, months, started. We can also observe from the word clouds that these have been common for all reviews in all classes of both datasets.

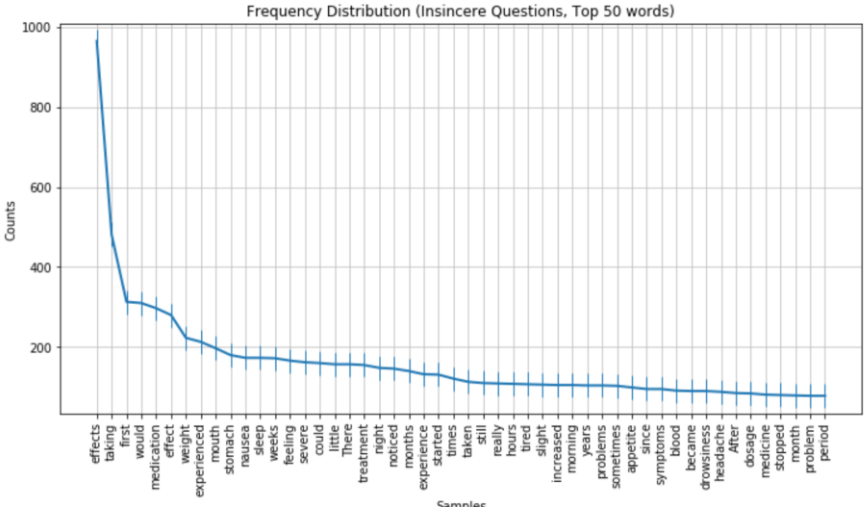

Figure 14: Frequency distribution of Druglib.com Considerably/Highly Effective

Figure 15: Word Cloud of Druglib.com Considerably/Highly Effective

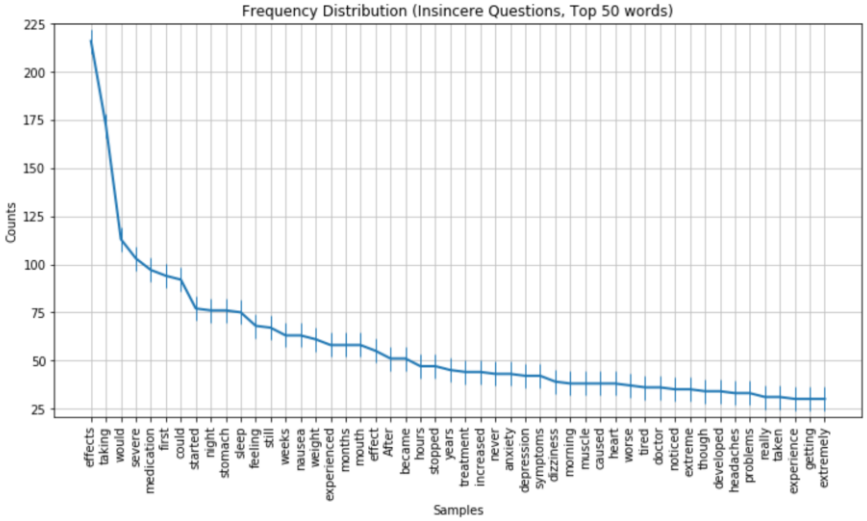

Figure 16: Frequency distribution of Druglib.com Ineffective class

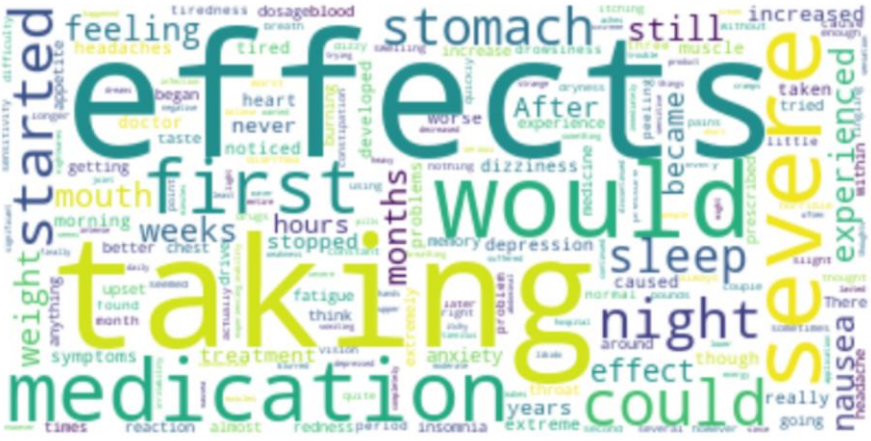

Figure 17: Word Cloud of Druglib.com Ineffective class

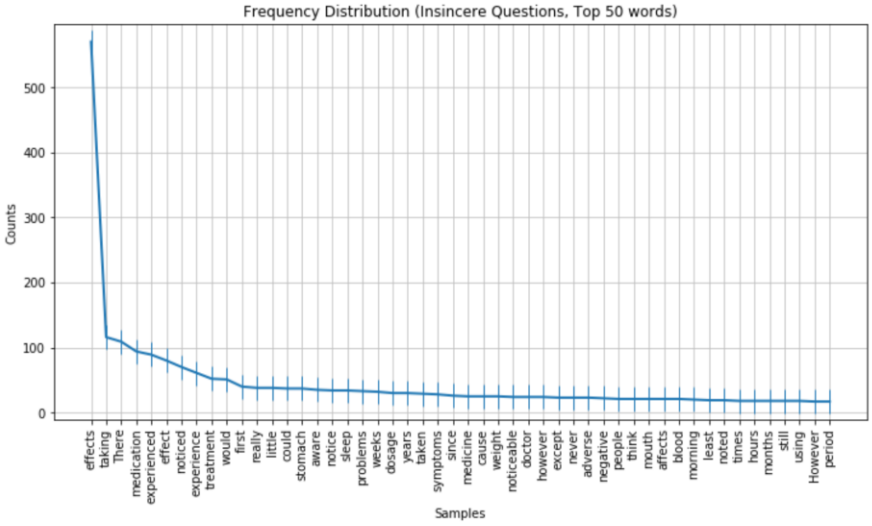

Figure 18: Frequency distribution of Druglib.com No side effects class

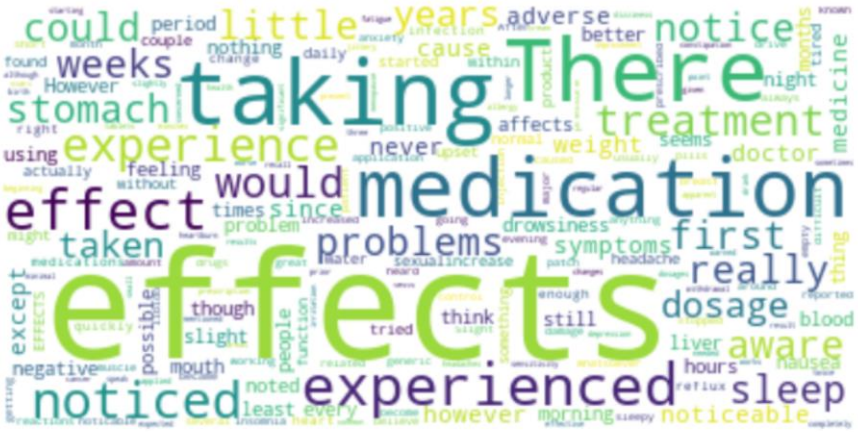

Figure 19: Word Cloud of Druglib.com No side effects class

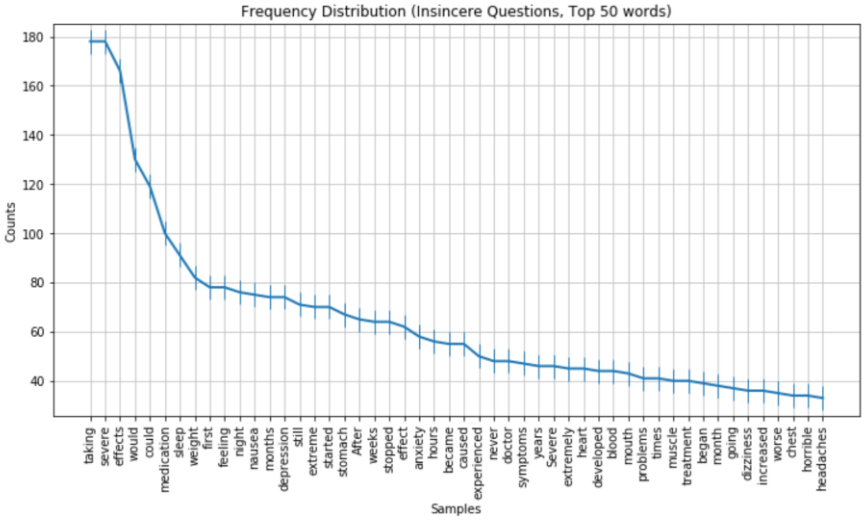

Figure 20: Frequency distribution of Druglib.com Extremely side effects class

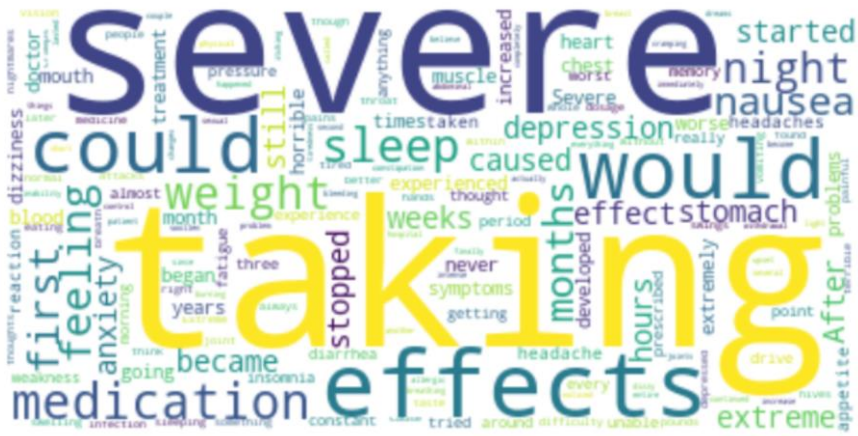

Figure 21: Word Cloud of Druglib.com Extremely side effects class

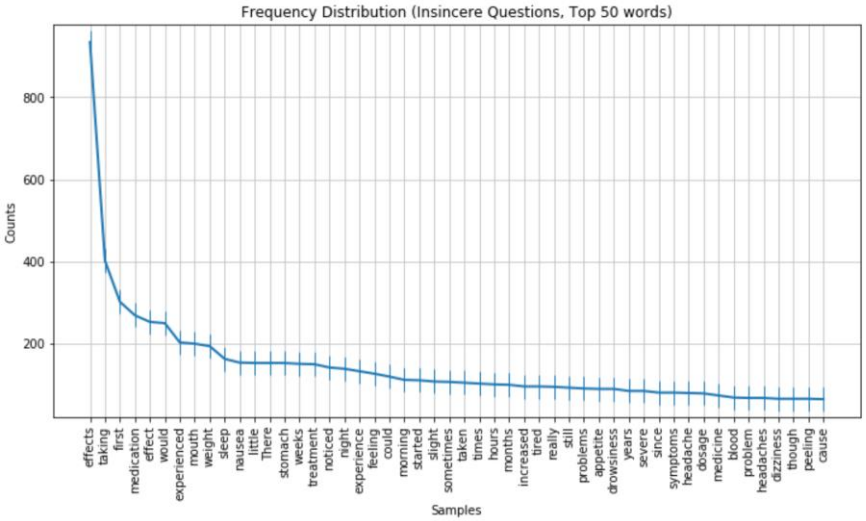

Figure 22: Frequency distribution of Druglib.com rating >=7 class

Figure 23: Word Cloud of Druglib.com rating >=7 class

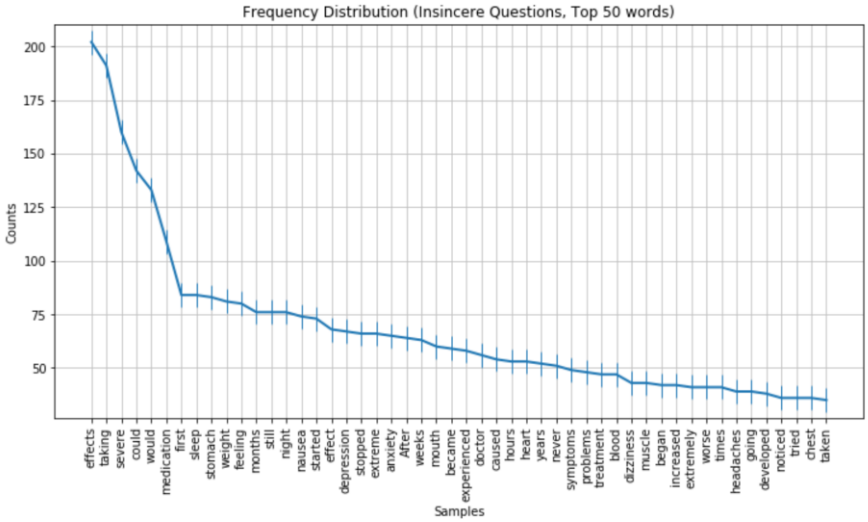

Figure 24 : Frequency distribution of Druglib.com rating <=4 class

Figure 25: Word Cloud of Druglib.com rating <=4 class

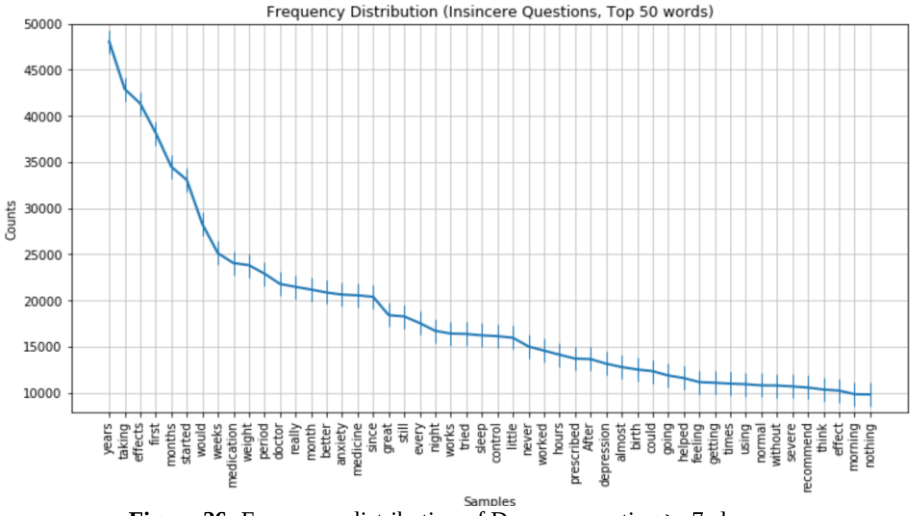

Figure 26: Frequency distribution of Drugs.com rating >=7 class

Figure 27: Word Cloud of Drug Drugs lib.com rating >=7 class

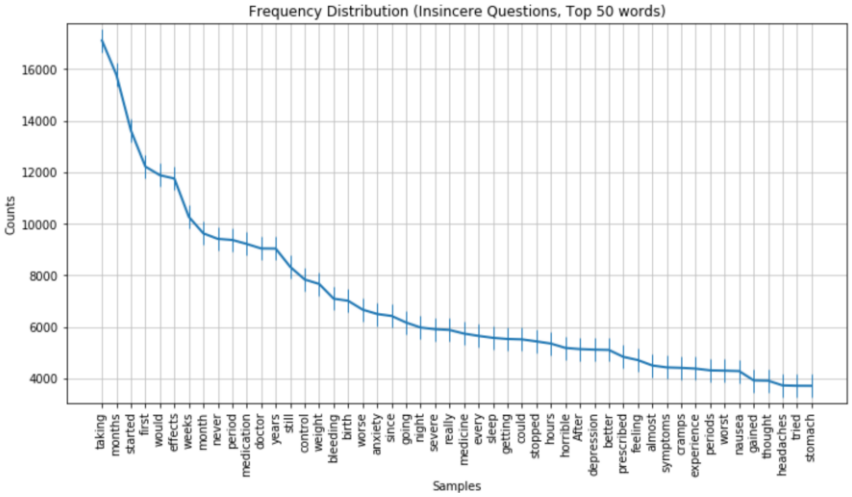

Figure 28: Frequency distribution of Drugs.com rating <=4 class

Figure 29: Word Cloud of Drug Drugs lib.com rating <= 4 class

After the above analysis we settle for the two approches mentioned below.

3 Approaches

In this section a description of the methods used in this work is detailed. The objective of this study was two fold:

3.1 In-domain Sentiment Analysis

Prediction of the overall patients' satisfaction with applied medications and sentiments on side effects and effectiveness by commissioning classification-based sentiment analysis. All the models are implemented for each data set (Drugs.com and Druglib.com), to classify overall patient satisfaction, reviews will be trained and evaluated employing the subsequent training and test data. Furthermore, as in case of the Druglib.com, a combination of all three reports (benefits, side effects and comments) of a patient on a particular drug were concatenated to signify the overall patient satisfaction review. We studied the expression of sentiments on the two aspects side effects and effectiveness within patient generated texts.

3.2 Cross-data Sentiment Analysis

Evaluating the transferability of models across data sources, that is Drugs.com and Druglib.com, by learning a model on reviews from one data source to classify overall patient satisfaction. We again used two methods, that is model is trained on drugs.com and tested on druglib.com and vice versa to predict the ratings in each dataset. We used the same machine learning models that we used for In-domain sentimental analysis. We tried approaching this problem in a different way than proposed earlier as manual labelling side-effects is not easy to achieve and also the problem of test and train on different domains of pharmaceutical reviews was much required at this point of time. The results were discussed in the results section.

4 Methods

4.1 Text Processing

Before we start with the classification, we need to clean the text in our reviews. Usually, this phase is crucial and typically takes a long time than building Machine Learning models.

Text preprocessing is amongst the highly essential methods in Natural Language Processing (NLP). For example, you would want to remove all punctuation marks from text documents prior to text classification. Likewise, you would want to remove numbers from a text string. Writing manual scripts for such preprocessing methods involves a lot of work and likely to errors. Keeping in mind the significance of these preprocessing methods, the Regular Expressions have been established in different languages in order to make these text preprocessing tasks easier. A Regular Expression is a text string that illustrates a search model which is used to match or replace patterns inside a string.

4.1.1 Data Cleaning

Now Let's see what concerns and what doesn't in Sentiment Analysis. Words are the most crucial part of sentiment analysis. Nevertheless, when we talk about factors like punctuation, you cannot understand the sentiment of the review from its punctuation. So, punctuation is not important in Sentiment Analysis. Furthermore, review elements like URLs, images, usernames, emojis, special characters, etc. does not add to the sentiment of the review. After obtaining the text, text normalization is applied on it using regular expression.

Text normalization includes:

- converting all letters to lower or upper case
- removing numbers
- removing white spaces
- removing stop words
- removing URL
- removing usernames
- removing special characters
- removing repeated characters
- removing all single characters
- substituting multiple spaces with single space
- removing Spaces from Start and End

Special characters include – [! " # $ % & ' ( ) * + , - . / : ; < = > ?]. Stop Words are words which have a bit or no importance, particularly when forming essential features from text. These are mostly words that hold the highest frequency in a corpus. These can be prepositions, articles, conjunctions, etc. The list of stop words is mentioned in appendix section 7.1. Consider the words such as "effect" and "Effect", from our reviews. These words give the same meaning for humans, the only distinction between them for us would be that the initial word is capitalized, for the reason that, it may be the initial word of a sentence or typing error. But for the models, these words will have a different meaning because of their different spelling, so we first convert all the words into lower case. Remove repeated characters is converting "helloooooooo" into "hello".

4.1.2 Tokenization

The function of splitting or tokenizing the provided text into smaller pieces termed as tokens is called tokenization. Numbers, Words, punctuation marks, etc. can be treated as tokens. We use Word Tokenization which is Splitting words in a sentence using word tokenize function built in nltk package in python.

4.1.3 N-grams

NLP technique, N-grams is basically a sequence of N words. For example, “Toy Story” is a 2-gram and “the dark Knight” is a 3-gram. Now let’s see how this is important in our context, if a review had the three word sequence “didn’t like drug” we would only consider these words individually with a unigram and maybe not capture that this is in fact a negative review since the word ‘like’ by itself will be extremely correlated with a positive review. This is the reason we use n-grams in our approaches. We use this with Count Vectorization in scikit learn with a range of one to three words, that is 1-grams to 3-gram. We tried all possibilities and settled for this range as this was more suitable for our data. For example, in our data set some reviews mentioned side effects as one word and some as two words and this is very crucial for our analysis, to take such words into judgement we used range 1 to 3.

4.2 Feature Extraction

4.2.1 TF-IDF

A popular way to signify each document in a corpus is to use the TF-IDF statistic (term frequency-inverse document frequency) for each word, which is a weighting factor that can be used instead of binary or word count representations. The concept behind the TF-IDF method is that the words that appear less in all the documents and more in individual document participate more towards classification.TF-IDF is a combination of two terms.

TF: Term Frequency, calculates how repeatedly a term appears in a document. Because every single document is distinct in length, it is likely that a term would occur much more times in longer documents than shot ones. Therefore, the term frequency is usually divided by the length of the document, which is the total number of terms in the document ,as a method of normalization.

$$TF = \frac{(frequency\ of\ a\ word\ in\ the\ document\ )}{(Total\ words\ in\ the\ document)}$$

IDF: Inverse Document Frequency, calculates how valuable a term is. While calculating TF, all terms are considered uniformly significant. However, it is known that some terms, such as "is", "of", and "that", may occur a lot of times but have slight significance. Therefore, we must weigh down the frequent terms while scale up the infrequent ones, by calculating the following:

$$IDF = \log(\frac{total\ number\ of\ documents}{number\ of\ documents\ containing\ the\ word})$$

TF-IDF: Term Frequency-Inverse Document Frequency

$$TF\_IDF = TF * IDF$$

If the TF-IDF score is high, it implies that the words are fairly rare and is good at distinguishing between documents. This could be more useful than Count Vectorizer, which only counts how many times a word occurs.

4.2.2 Count Vectorization

Count Vectorization comprises of calculating the number of occurrences of each word in a document Python's Sci-kit learn library has a tool called Count Vectorizer to accomplish this. Example sentence: "The side effects outweigh the benefits as there were more side effects." You can infer from this that the words "the", "side" and "effects" occurred twice while others occurred once, which is what Count Vectorization achieves. Count Vectorization in scikit learn has a parameter called n-gram range. As discussed above we used this parameter. So, for each model we texted our data using only count vectorization and count vectorization with n-grams.

4.2.3 Word Embeddings

NLP based feature learning technique, Word Embeddings maps words into vectors of real numbers that capture the context of the underlying words in relation to other words in the sentence. This conversion outcomes in words that can be depicted as a vector in an n-dimensional vector space and the distance between two such vectors depicts the level of resemblance between these words. This is compared to the thousands of dimensions necessary for sparse word representations, like one-hot encoding. For instance, we can embed the words "effects" and "taking" as dimensional vectors as [0.8 1.0 4.2 7.5 3.6] , [8.3 5.7 7.8 4.6 2.5] respectively.

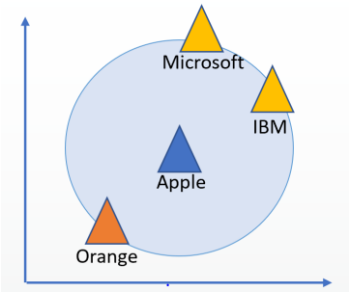

Figure 30: Representation of word vectors

For example, consider the figure 30 shows the representation of words from two categories fruits and companies. Words “IBM”, “Microsoft”, “Apple”, “Orange” are taken. The words IBM and Microsoft are closer as these both are company names which is the similarity between these two words and orange is located further away as this word is not similar to company names, as it is a fruit. The keep focus here should be on “apple” as this can be a fruit or a company, this has similarity with both orange and IBM, Microsoft so it is located in between the two categories in dimensional space.

Now, let’s see the input into these embedding with respect to our deep learning models used for this drug datasets. Firstly, the text from reviews must be tokenized by fitting Tokenizer class on the data set. We used “lower = True” argument to transform the text into lowercase ensuring uniformity of the data. Later, we mapped our list of words (tokens) to a list of unique integers for each unique word applying texts_to_sequences class. the original reviews convert into a sequence of integers after applying preprocessing. Then, we use pad_sequences class on the list of integers to make sure that all reviews are of same length, which is a very crucial step for preparing data for deep learning models. Using this class will either cut the reviews to 100 integers or pad them with 0’s if they are smaller. As embedding needs the length of input sequences and size of the vocabulary, setting vocabulary size equal to the number of words in Tokenizer dictionary + 1 and input length as 100 (maximum words), where value of the last parameter must be the same as for padding. Embedding size parameter indicates how many dimensions will be used to depict each word.

4.3 Machine Learning Algorithms

Normally, sentiment analysis for text data can be processed on numerous stages, including an individual sentence stage, paragraph stage, or the complete document at once. Regularly, sentiment is processed on the document at once or some clusters are done after processing the sentiment for individual sentences. The two general methodologies to sentiment analysis are Supervised Machine Learning and Deep Learning.

4.3.1 Supervised Machine Learning

Supervised learning is extremely popular. The data set is a set of labelled examples. Each x_i is a feature (attribute) vector with D dimensions. x_k^j is the value of the feature j of the example k, for $j \in 1 \dots D$ and $k \in 1 \dots D$. The label y_i is either a class, taken from a finite list of classes, $\{1,2, \dots, C\}$ or a real number, or a more complex object (vector, matrix, tree, graph, etc.). The Problem is: given the data set as input, create a "model" that can be used to predict the value of y for an unseen x. Regression and classification are categorized under supervised machine learning. The machine learning techniques used for this dataset are classification.

We can see from table 1, that the polarity for each aspect of review is in three levels for Side effects, ratings and effectiveness. So our classification of these reviews into the different polarity is required and this defines it to be a multi label classification. The classifiers we use for this dataset are K-Nearest Neighbors, logistic Regression, Random Forest, Bagging Classifier.

An ensemble meta-estimator that fits base classifiers each on random subsets of the initial dataset and then cumulative of their individual predictions, by either voting or averaging, to form a final prediction is called Bagging classifier. Such a meta-estimator is usually used as a way to decrease the variance of a black-box estimator, by initiating randomization into its development process and then producing an ensemble out of it. Bagging decreases overfitting, this leads to a rise in bias, that is compensated by the decrease in variance though. The base classifier for our dataset is Decision Tree Classifier.

4.3.2 Deep Learning

The models that are trained on large sets and neural network architectures that learn features straight from the data with no need for manual feature extraction are called Deep learning models. We are going to use two types of Recurrent Neural Network, known as Long Short Term Memory (LSTM) and Bidirectional Long Short Term Memory (BiLSTM). The main idea of LSTM is that it is able to remember the sequence of past data i.e. words in our case in order to make a decision on the sentiment of the word. This is the reason it is widely used in time series analysis. The idea of BiLSTM is it includes duplicating the first recurrent layer in the network to create two layers side-by-side, then giving the input sequence as input to the first layer and giving a reversed copy of the input sequence to the second layer. Hence, BiLSTM memorizes the past and future data i.e. words in order to decide on the polarity of the word.

We are going to create the network using Keras. Keras is built on TensorFlow and can be used to build most types of deep learning models. We are going to specify the layers of the model as below.

5 Experiments and Results

Now let's see the results of each Supervised learning model using three methods on the dataset which are Count vectorization, Count vectorization with N-grams and TD-IDF. We will see the classification report and confusion matrix for the each model.

Confusion matrix: is a synopsis of prediction results on a classification problem. The count of correct and incorrect predictions are encapsulated and divided by each class. The confusion matrix reveals the ways in which your classification model is puzzled when it makes predictions. It gives us vision on the errors being made by a classifier and the types of errors that are being made. The classification report reveals a depiction of the main classification metrics on a per-class basis. This gives a greater insight on the classifier performance over overall accuracy which can conceal functional weaknesses in one class of a multiclass problem. Visual classification reports are used to analyze classification models to choose models. The components of classification report are explained below in detail.
Precision: is the capability of a classifier not to label a negative instance positive. For each class it is identified as the ratio of true positives to the sum of true and false positives.
Recall: is the capability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.
F_1 score: is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. In general, F1 scores are lower than accuracy measures as they implant precision and recall into their calculation.
Support: is the number of actual occurrences of the class in the specified dataset.
Accuracy: is the ratio of number of correct predictions to the total number of input samples.
micro: Calculate metrics globally by counting the total true positives, false negatives and false positives.
macro: Calculate metrics for each label and find their unweighted mean, does not take label imbalance into account.
weighted: Calculate metrics for each label, and find their average weighted by support, does take label imbalance into account.

From figure 2-5 we can infer that our data is imbalanced and for imbalanced data the best metric is accuracy and weighted F1 Score (weighted), so we conclude which is the best classifier for each approach using these metrics. The best performing classifiers confusion matrix and learning curves are shown below, rest are not listed here to keep the report concise, but can refer to the code provided in .pdf format for all the results. We did not show the results for bagging classifier using Count Vectorizer with n grams as this was consuming huge running time more than 2 days.

5.1 In-Domain Sentiment Analysis on Drugs.com

Table 2: Results of Drug.com Data

		Rating	
		Accuracy	**F1 score**
KNN	**CV**	0.69	0.67
	TF-IDF	0.66	0.54
	CV and n-grams	0.65	0.63
Logistic Regression	**CV**	0.8	0.79
	TF-IDF	0.79	0.76
	CV and n-grams	**0.91**	**0.92**
Random forest	**CV**	**0.89**	**0.89**
	TF-IDF	**0.89**	**0.89**
	CV and n-grams	0.88	0.88
Bagging Classifier	**CV**	0.87	0.87
	TF-IDF	0.87	0.88
LSTM	**Embedding and LSTM**	0.82	0.75
	Embedding and BiLSTM	**0.81**	**0.81**

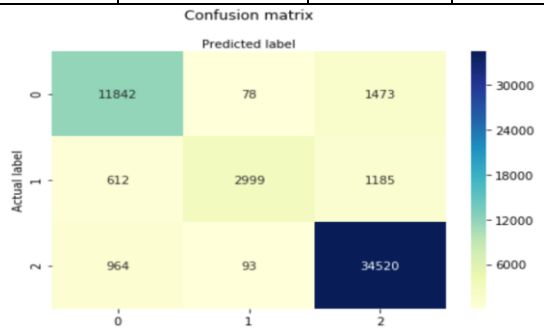

Figure 31: Count Vectorizer with N-grams and Logistic Regression

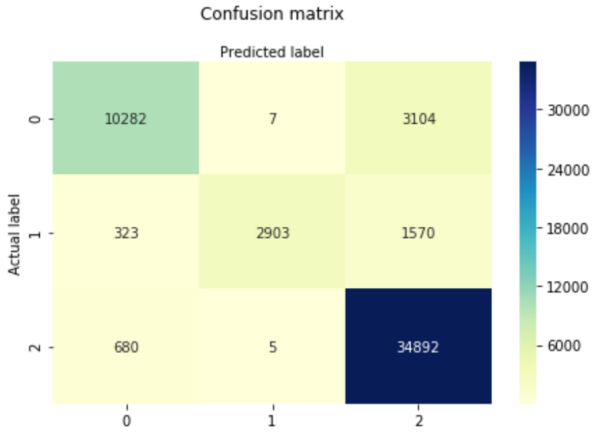

Figure 32: Count Vectorizer and Random Forest

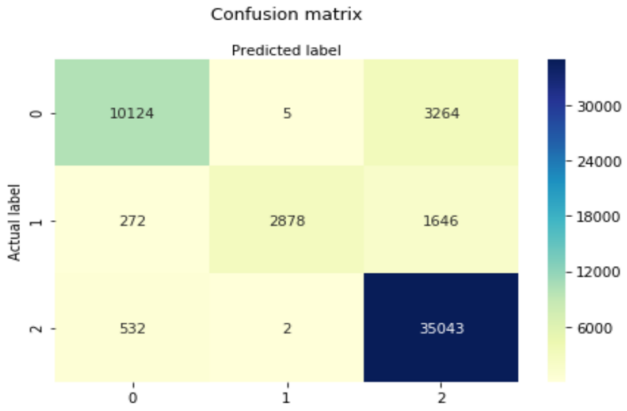

Figure 33:TD-IDF and random forest

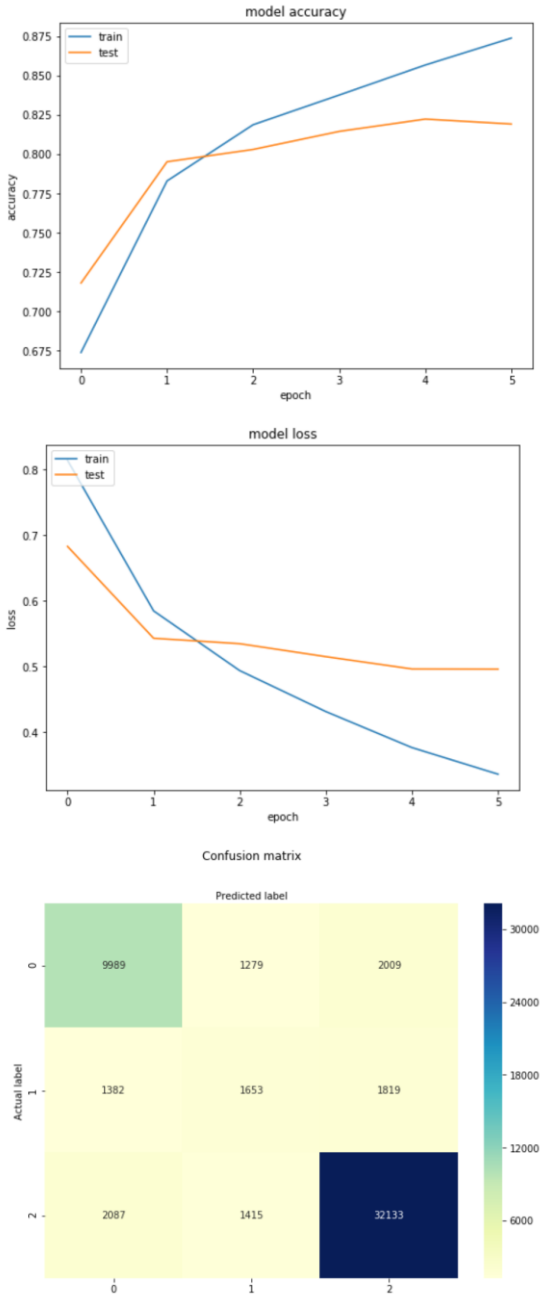

Figure 34: BiLSTM

5.2 In-Domain Sentiment Analysis on Druglib.com

Table 3: Results of Druglib.com Data

		effectiveness		side-effects		rating	
		Accuracy	**F1 score**	**Accuracy**	**F1 score**	**Accuracy**	**F1 score**
KNN	**CV**	0.62	0.63	0.58	0.56	0.56	0.54
	TF-IDF	0.72	0.68	0.34	0.24	0.67	0.62
	CV and n-grams	0.35	0.36	0.44	0.38	0.61	0.54
Logistic Regression	**CV**	0.74	0.72	0.74	0.74	0.73	0.72
	TF-IDF	**0.75**	**0.68**	**0.75**	**0.74**	0.71	0.63
	CV and n-grams	0.75	0.71	**0.75**	**0.74**	**0.76**	**0.71**
Random forest	**CV**	0.74	0.71	0.72	0.7	0.7	0.63
	TF-IDF	0.75	0.7	0.73	0.71	0.7	0.62
	CV and n-grams	0.75	0.69	0.7	0.66	0.69	0.59
Bagging Classifier	**CV**	0.69	0.68	0.68	0.68	0.71	0.67
	TF-IDF	0.72	0.7	0.71	0.71	0.67	0.64
	CV and n-grams	0.72	0.69	0.71	0.71	0.71	0.66
LSTM	**Embedding and LSTM**	0.72	0.61	0.56	0.51	0.68	0.56
	Embedding and BiLSTM	**0.81**	**0.61**	**0.79**	**0.62**	**0.78**	**0.68**

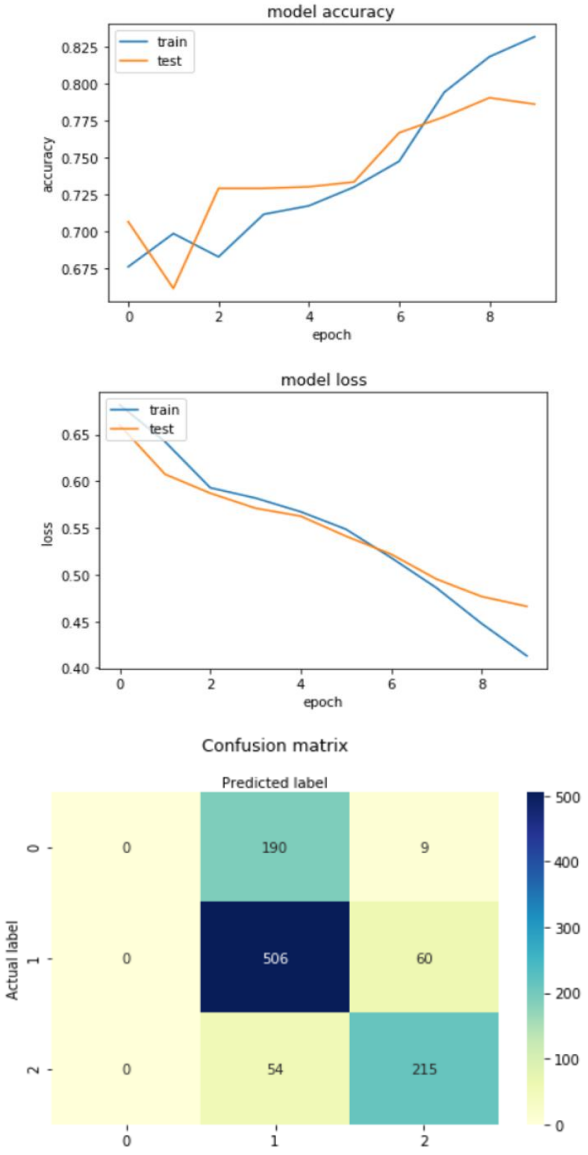

Figure 35: Effectiveness – BiLSTM

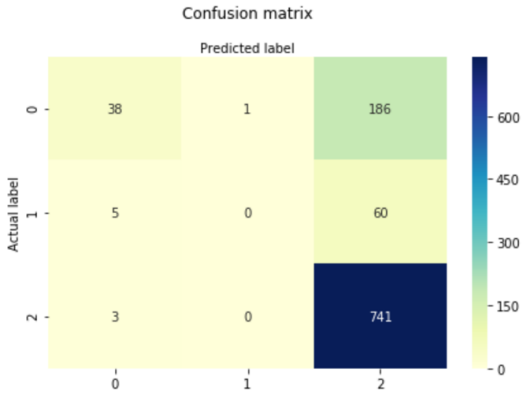

Figure 36: Effectiveness Class- TD-IDF and Logistic Regression

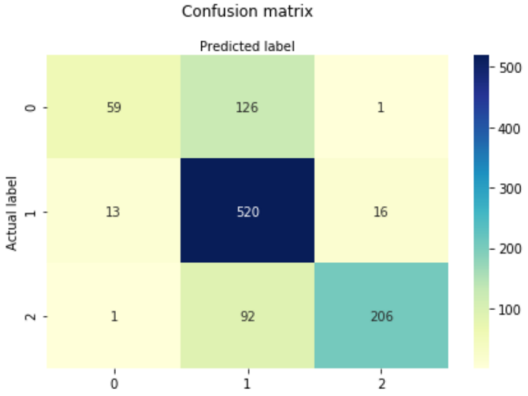

Figure 37: Side Effects - TD-IDF and Logistic Regression

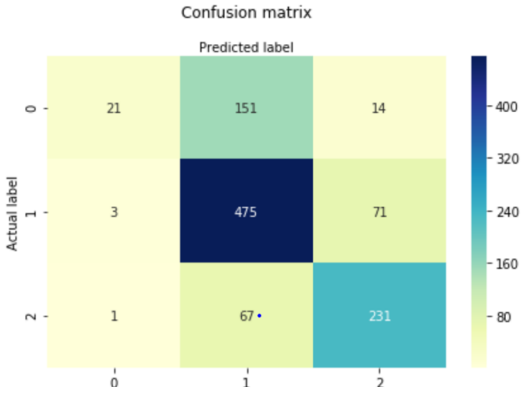

Figure 38: Side Effects - Count Vectorizer with N-grams and Random Forest

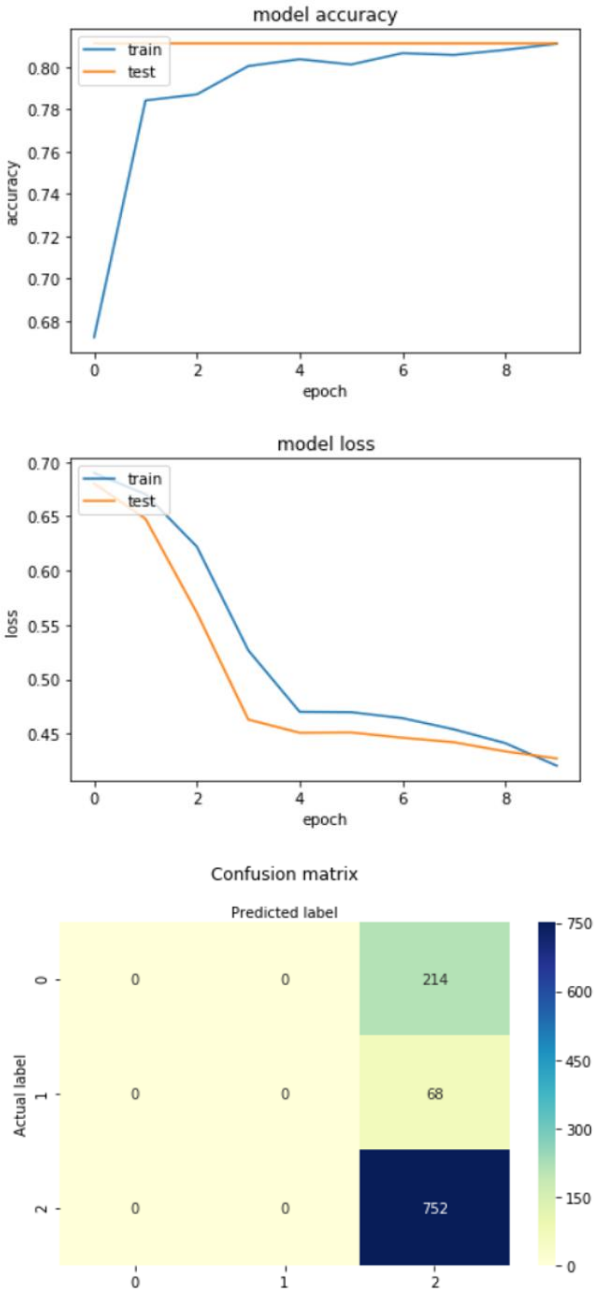

Figure 39: Side effects - BiLSTM

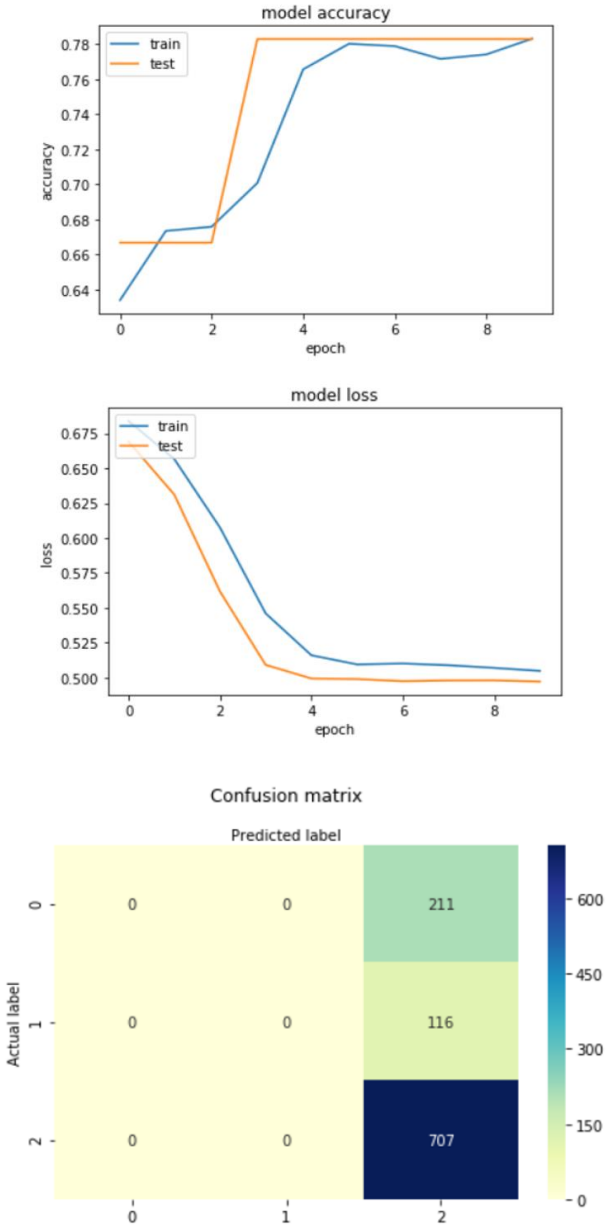

Figure 40: Rating- BiLSTM

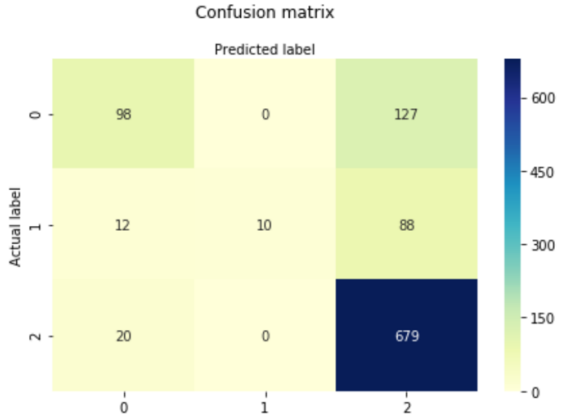

Figure 41: Rating - Count Vectorizer with N-grams and Logistic Regression

5.3 Cross-data Sentiment Analysis train on druglib.com

Here the train data is small and test data is really huge.

Table 4: Results of Train on Druglib.com Data

		rating	
		Accuracy	**F1 score**
KNN	**CV**	0.62	0.53
	TF-IDF	0.63	0.59
	CV and n-grams	**0.66**	**0.53**
Logistic Regression	**CV**	0.68	0.66
	TF-IDF	**0.69**	**0.62**
	CV and n-grams	**0.7**	**0.65**
Random Forest	**CV**	0.66	0.56
	TF-IDF	0.67	0.58
	CV and n-grams	0.66	0.53
Bagging Classifier	**CV**	0.66	0.6
	TF-IDF	0.64	0.61
	CV and n-grams	0.67	0.61

LSTM	**Embedding and LSTM**	0.66	0.61
	Embedding and BiLSTM	0.60	0.52

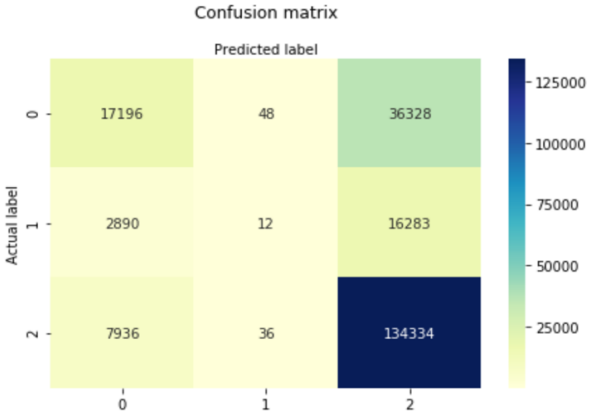

Figure 42: Rating - Count Vectorizer with N-grams and Logistic Regression

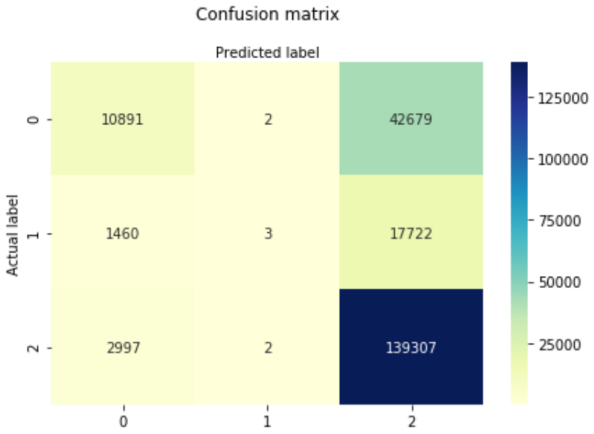

Figure 43: Rating – TDF-IDF and Logistic Regression

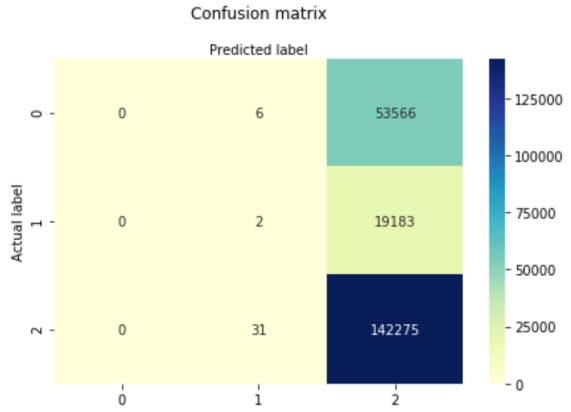

Figure 44: Rating – Count Vectorizer with N-grams and KNN

5.4 Cross-data Sentiment Analysis train on drugs.com

Here the train data is large and test data is small.

Table 5: Results of train on Drug.com Data

		Rating	
		Accuracy	**F1 score**
KNN	**CV**	0.61	0.59
	TF-IDF	0.67	0.55
	CV and n-grams	0.6	0.58
Logistic Regression	**CV**	0.72	0.69
	TF-IDF	**0.75**	**0.7**
	CV and n-grams	0.76	0.72
Random Forest	**CV**	0.7	0.63
	TF-IDF	0.69	0.61
	CV and n-grams	**0.69**	**0.59**
Bagging Classifier	**CV**	0.63	0.62
	TF-IDF	0.68	0.65

LSTM	**Embedding and LSTM**	0.56	0.51
	Embedding and BiLSTM	0.57	0.52

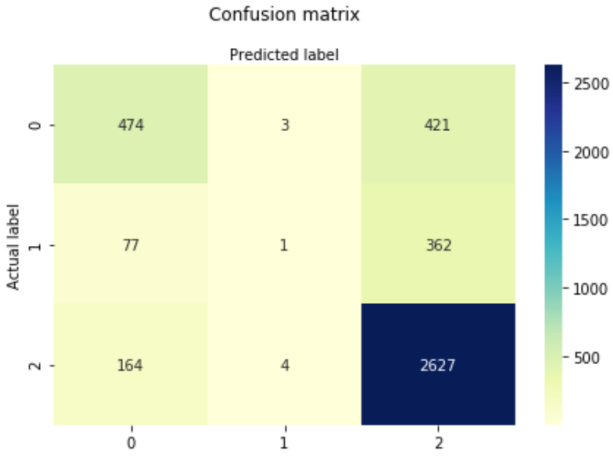

Figure 45: Rating – TDF-IDF and Logistic Regression

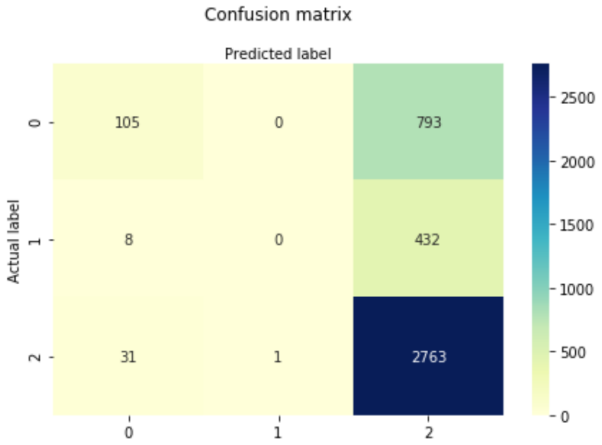

Figure 46: Rating – Count Vectorizer with n-grams and Random Forest

For the deep learning models, the train data is again split into train and validation test, to tune the hyperparameters. The deep learning model hyperparameters like epochs, optimizer, batch size have been chosen depending on the training data and validation loss. Major focus is not to increase the validation loss.

6 Conclusion

Within this work, we analyzed the application of machine learning based sentiment analysis of patient created drug reviews. On only patient satisfaction but sentiment aspects regarding effectiveness and suffered side effects were analyzed. Depending on aspect and data source, promising classification results were achieved. In-domain Sentimental analysis, training and evaluation shows very good classification results but Cross-data Sentimental analysis are not much satisfactory as in-domain sentiment analysis. Additionally, the results evidently indicate that especially aspect-based sentiment analysis requires more massive data sets to extract features with adequate generalization abilities as druglib.com data is more organized, but had better results of drugs.com. The deep learning models were one of the best models for in-domain sentiment analysis but not on cross-domain sentiment analysis. However, we believe that this work contributes to open up future research using sampling techniques with more data.

7 Appendix

7.1 Stop words in English

{'about', "you're", 'that', 'after', 'yourself', 'yours', "wouldn't", 'should', 'be', 'themselves', 'if', 'to', 'didn', 'there', 'my', "you'll", 'for', 'into', 'all', 'more', 've', 'ours', 'other', 'couldn', "mightn't", 'her', 'did', "you'd", 'ma', 'down', 'the', 'doesn', 'because', 'both', 'through', "doesn't", 'me', 'o', 'not', 'too', 'most', 't', 'they', 'won', 'will', "hasn't", 'you', 'any', 'just', 'll', 'shan', 'yourselves', 'below', 'nor', 'we', 'when', "it's", "mustn't", 'those', 'or', 'isn', 'them', 'here', 'such', 'haven', 'is', 'do', "don't", 'at', 'under', "should've", 'then', 'hasn', 'against', 'ourselves', "wasn't", 'himself', 'your', "that'll", 'don', "hadn't", 'am', 're', 'weren', 'were', "aren't", 'his', 'few', 'its', "you've", 'up', 'than', "weren't", 'before', 'itself', 'with', 'again', 'these', "haven't", 'mightn', 'what', 'how', 'whom', 'doing', 'being', "shan't", 'wasn', "isn't", "didn't", 'some', 'it', 'above', 'only', 'an', 'each', 'i', 'further', 'she', 'a', 'own', 'which', 'has', 'been', 'does', 'no', 'their', 'd', 'who', 'of', 'having', 'as', 'where', 'needn', 'y', "shouldn't", 'by', 'hadn', 'why', 'had', 'out', 'now', 'mustn', 'was', 'but', 'myself', 'same', "couldn't", 'theirs', 'until', 'off', "she's", 'very', "won't", 'ain', 'hers', 'so', 's', 'can', 'our', 'once', 'while', 'this', 'are', 'from', 'aren', 'he', "needn't", 'during', 'wouldn', 'between', 'have', 'and', 'herself', 'm', 'on', 'him', 'in', 'shouldn', 'over'}

References

1. National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov
2. https://www.nltk.org/data.html - to download nltk
3. Using Regex for Text Manipulation in Python from stackabuse.com
4. https://towardsdatascience.com/
5. Beginner's guide to Web Scraping in Python using Beautiful Soup by analyticsvidya
6. Pattern Mining for Extraction of mentions of Adverse Drug Reactions from User Comments by Azadeh Nikfarjam, MSI, Graciela H. Gonzalez, PhD1 1 Biomedical Informatics Department, Arizona State University, Phoenix, AZ.
7. Xavier Glorot, Antoine Bordes, and Yoshua Bengio.2011. Domain Adaptation for Large-scale Sentiment Classification: A Deep Learning Approach. In Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML'11). Omnipress,USA,513–520.
8. Diana Cavalcanti and Ricardo Prudêncio.2017. Aspect-Based Opinion Mining in Drug Reviews. In Progress in Artificial Intelligence, Eugénio Oliveira, João Gama, Zita Vale, and Henrique Lopes Cardoso(Eds.). Springer International Publishing, Cham,815–827.
9. Sentiment Analysis of Twitter Data Apoorv Agarwal Boyi Xie Ilia Vovsha Owen Rambow Rebecca Passonneau Department of Computer Science Columbia University New York, NY 10027 USA
10. Research on text sentiment analysis based on CNNs and SVM by Yuling Chen Zhi Zhang . 2018 IEEE
11. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches ,Qiang Ye , Ziqiong Zhang , Rob Law.
12. Yang, P., & Chen, Y. (2017). A survey on sentiment analysis by using machine learning methods. 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC).
13. Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani , Veselin Stoyanov, SemEval-2016 Task 4: Sentiment Analysis in Twitter
14. 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology, Natural Language Processing for Sentiment Analysis An Exploratory Analysis on Tweets Wei Yen Chong , Lay-Ki Soon.
15. Patient opinion mining to analyze drugs satisfaction using supervised learning. Journal of Applied Research and Technology 15, 4 (2017), 311 – 319.