Drug Review Sentiment Analysis

Sumit Mishra
New Delhi, India
sumit.mishra0432@gmail.com

***Abstract*—The Drug Review Dataset is taken from the UCI Machine Learning Repository. This Dataset provides patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting the overall patient satisfaction. The data was obtained by crawling online pharmaceutical review sites. It contains the features like the Rating of the drug according to the patient, the condition of the patient, the name of the drug, the date of the usage and the usefulcount which is the number of users who found the review useful. This data set contains different types of features which are categorical, numerical, text and date. The sentiment analysis is the main objective, in which we have to classify a review as positive or negative. The train and test set are initially merged as the main objective is to do sentiment analysis. The sentiment of the review is given according to the rating as the sentiment of the reviews are not given initially in the dataset. Exploratory data analysis is done to get good insight and engineer features for good predictions. Preprocessing is done to get the data ready for further process. The Dataset is then split into train and test set. 70% of the data is used for training and the rest 30% of the data is used for testing. Three Classification models are trained LightGBM, XGBoost, and the CatBoost and the feature importance is also plotted. Confusion Matrix and accuracy score are used for the metric.**

Keywords*—*Sentiment Analysis, Classification, EDA, Feature Engineering, Confusion Matrix, textblob, nltk, Word Cloud.

I. INTRODUCTION

The Drug Review Dataset provides patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting the overall patient satisfaction. The purpose of this study is to do the sentiment analysis on the drug reviews given by the patients.

Initially, the dataset was segregated into two parts which are train and test set but as we want to do the sentiment analysis so we merged both the dataset to get more data to train and test overall (In this study we'll talk about the merged dataset only). Exploratory Data Analysis is done on the dataset with different features to gain insight about the data and the correlation between them which will help in feature engineering. The features like 'uniqueID' are not of much use as they are just the identity given to each of the data point or patient review. some preprocessing is done before the EDA so the data is ready for the Exploratory Data Analysis. Different bar graphs, Histogram and Word Clouds are plotted. The Reviews then are cleaned so that unnecessary words and elements are removed and the features are generated. Feature Engineering is done on both the uncleaned and cleaned reviews. Textblob module is also used to give the polarity to both the cleaned and uncleaned reviews and use it as a feature as well. The classification models are trained and their performances are compared on the basis of the confusion matrix and their accuracy score but the confusion matrix is given more weight is given to the confusion matrix as the accuracy score is sometimes misleading and don't give a proper metric for the performance of the ML model. The Classification algorithms that are used are LightGBM, XGBoost, and the CatBoost.

II. DATASET & EDA

A. Dataset

The Drug Review Dataset is taken from the UCI Machine Learning Repository. This Dataset provides patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting the overall patient satisfaction. The data was obtained by crawling online pharmaceutical review sites. The Drug Review Data Set is of shape (215063, 7) i.e. It has 7 features including the review and 215,063 Data Points or entries. The features are 'drugName' which is the name of the drug, 'condition' which is the condition the patient is suffering from, 'review' is the patients review, 'rating' is the 10-star patient rating for the drug, 'date' is the date of the entry and the 'usefulcount' is the number of users who found the review useful. Here the sentiment of the review is the target variable that needs to be predicted. here we can notice that the sentiment of any review is not given, so we have to give the sentiment to the rating first and then use it as the target variable. The drugName and condition are categorical features, the date is date object, rating and usefulcount are numerical features, and the review is text. These reviews are from the year 2008 to 2017.

B. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is done to get the insight about the data and summarize the main characteristics. To understand dependency or correlation of the features.

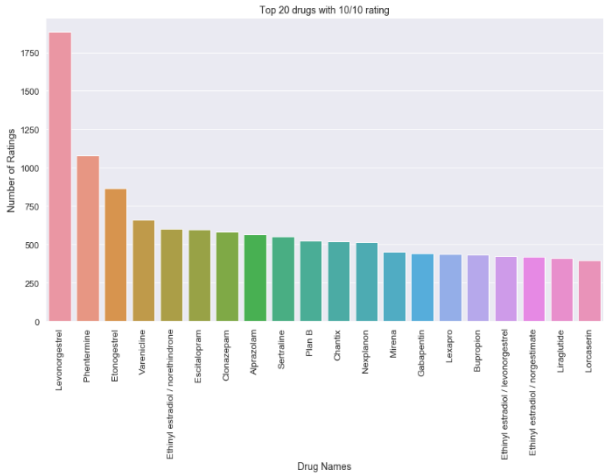

Fig – 1.0

The fig – 1.0 is a bar graph which shows the top 20 drugs given in the data set with a rating of 10/10. 'Levonorgestrel' is the drug with the highest number of 10/10 ratings, about 1883 Ratings in the data set for 'Levonorgestrel'. It's followed by 'Phentermine' with 1079 ratings.

Levonorgestrel (LNG) is a synthetic progestogen similar to Progesterone used in contraception and hormone therapy. Also known as Plan B, it is used as a single agent in emergency contraception, and as a hormonal contraceptive released from an intrauterine device, commonly referred to as an IUD.

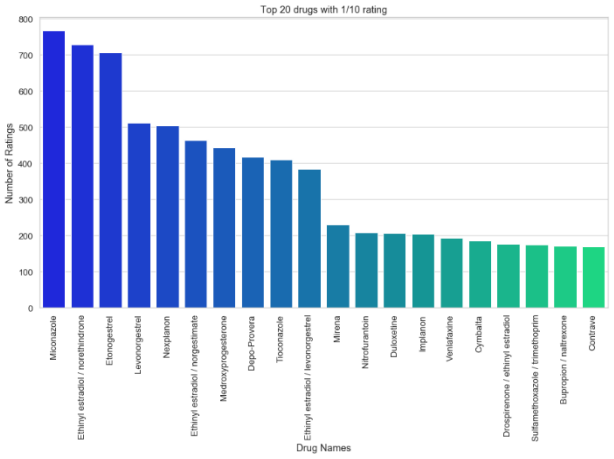

Fig – 2.0

The fig-2.0 is a bar graph that shows the top 20 drugs given in the data set with a rating of 1/10. 'Miconazole' is the drug with the highest number of 1/10 ratings, about 767. It's followed by 'Ethinyl estradiol / norethindrone' and 'Etonogestrel'.

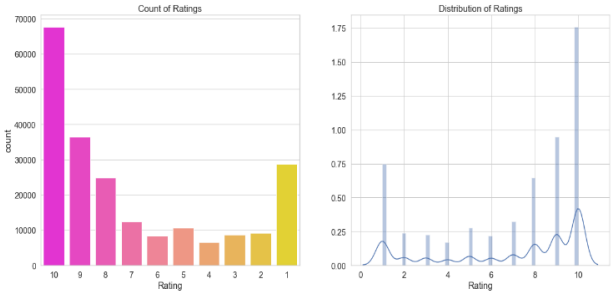

Fig – 3.0

The fig-3.0 shows a distribution plot on the right hand side and a bar graph of the same on the left hand side. This shows the distribution of the ratings from 1 to 10 in the data set. It can be inferred that mostly it's 10/10 rating and after that 9 and 1. The data points with rating of the drugs from 2 to 7 is pretty low.

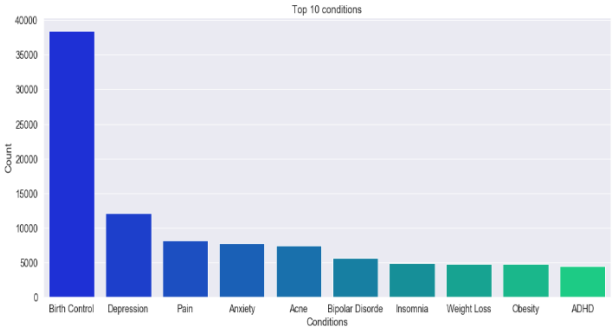

Fig – 4.0

The fig-4.0 is a bar graph which exhibits the top 10 conditions the people are suffering from. In this data set 'Birth Control' is the most prominent condition by a very big margin followed by Depression and pain. The 'Birth Control' condition has occurred about 38,436 and the depressions has occurred about 12,164. It can easily be noticed that the 'Birth Control' is more than 3 time the depression in the whole data set. In top 10 conditions ADHD is on the 10th Position, ADHD stands for Attention deficit hyperactivity disorder.

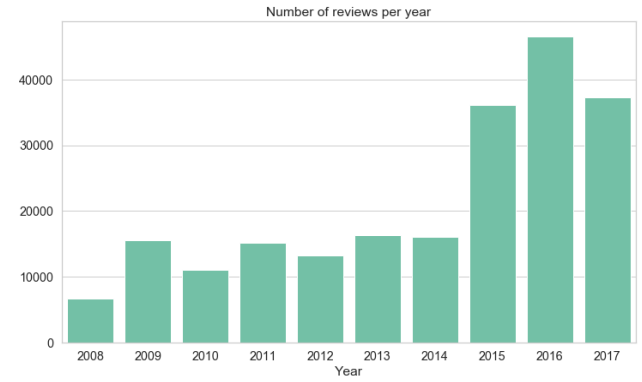

Fig – 5.1

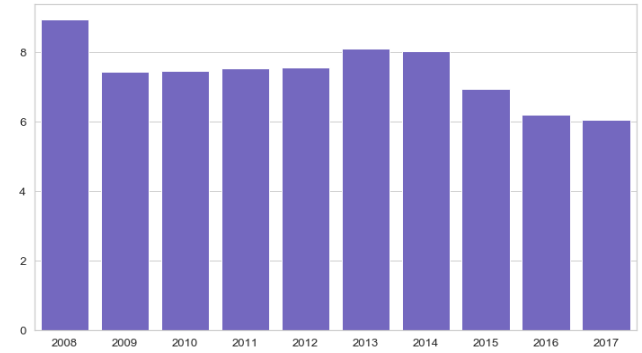

Fig – 5.2

The fig-5.1 is a Bar graph that shows the number of reviews in the data set per year. It can be inferred that most ratings are given in 2016 and 2008 has the least number of reviews. 2016 have 46507 reviews whereas 2008 have 6700 reviews.

The fig-5.2 shows the mean rating of the drugs per year. 2008 have a mean rating of 8.92 which is the highest but it can also infer from the above bar graph(fig-5.1) that it has the lowest number of reviews by the patients. 2017 has a mean rating of 6.04 which is the lowest in the graph. The average rating per year is not below 5 in any given year from 2008 to 2017.

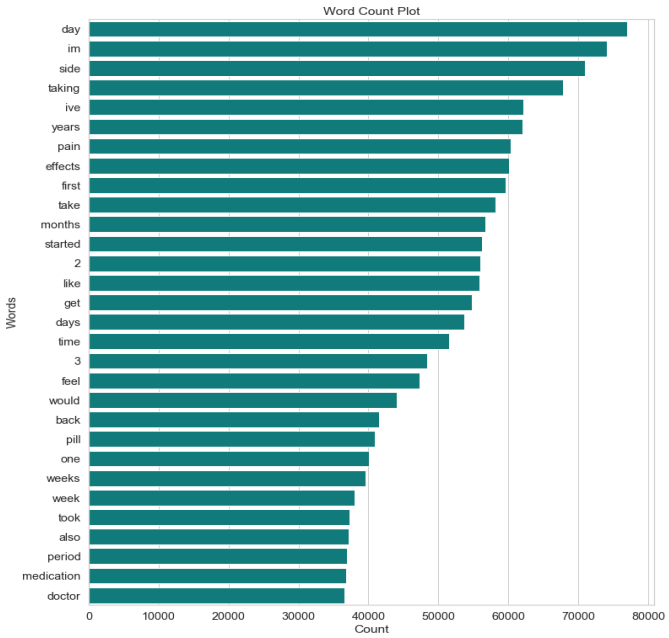

Fig – 6.0

This fig-6.0 is a Word Count Plot and in this top 30 most common words in the drug reviews are plotted. These are plotted after removing all the English stopwords. The most occurring word is ‘day' followed by 'im'. 'day' has occurred 76,947 times and 'im' occurred 74,032 times in the reviews. There are also several words in the top 30 words like 'medication', 'effects', 'period', 'doctor’, ‘pill' which gives a direct context of the reviews related to health.

C. N-grams

In the fields of computational linguistics and probability, an *n*-gram is a contiguous sequence of *n* items from a given sample of text or speech.

I have used seaborn and 'ngrams' from Natural Language Toolkit to plot the unigrams, bigrams and the trigrams of the drug review after a cleaning them a little bit. I have plotted the top 20 n-grams on the basis of the ratings i.e. n-grams for ratings less than or equal to 5 and ratings greater than 5.

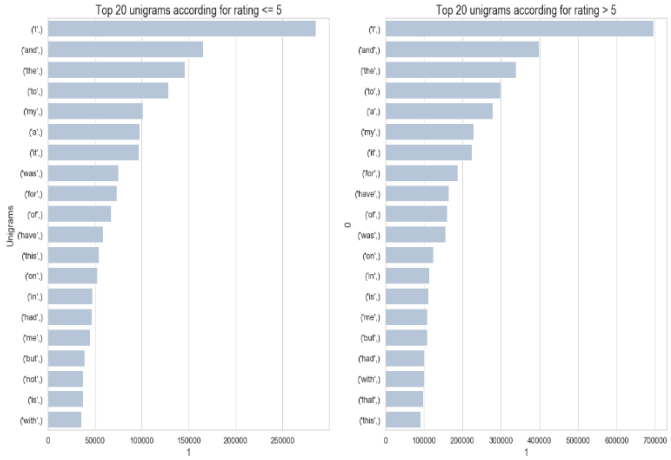

Fig – 7.1

This fig-7.1 depicts the top 20 **unigrams** on the basis of the ratings. Here "I" is the most occurring word in both the scenarios. For the rating less or equal to 5 “I” has occurred 285,406 times and for rating greater than 5 it has occurred 695,107times.

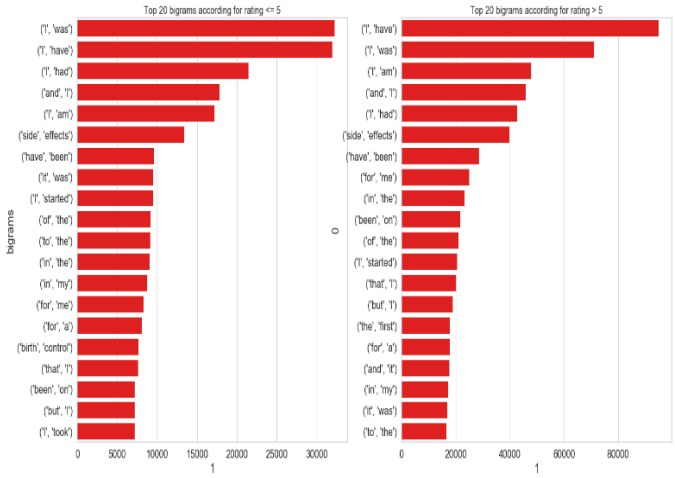

Fig – 7.2

This fig-7.2 depicts the top 20 **Bigrams** on the basis of the ratings. In the top 20 Bigrams for the rating<=5, the number of times the bigram ('I’, ‘was') and ('I', 'have') occurred is 32,246 and 31,928 respectively. And for the rating greater than 5 ('I', 'have') is the most occurring bigram which occurred 94,984 times.

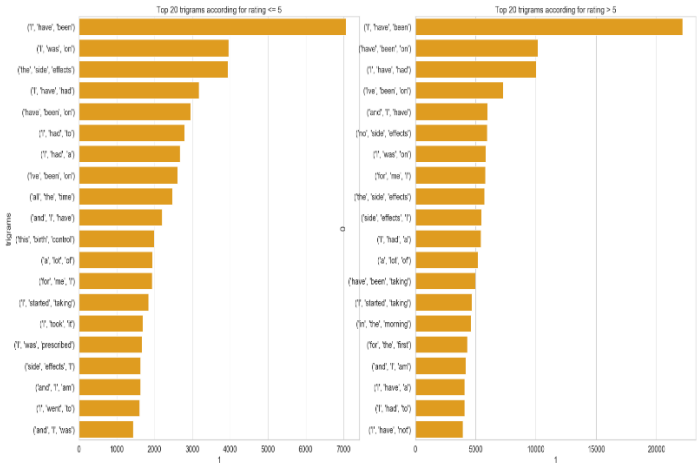

Fig – 7.3

This fig-7.3 depicts the top 20 **Trigrams** on the basis of the ratings. In the top 20 Trigrams for the rating<=5, the number of times the trigram ('I', 'have’, ‘been') and ('I', 'was’, ‘on') occurred is 7060 and 3951 respectively. And for the rating greater than 5, ('I', 'have’, ‘been') is the most occurring trigram which occurred 22,130 times. Here it can be seen that the trigram ('I', 'have’, ‘been') is occurred the most in both the scenarios.

D. Word Cloud

A tag cloud (**word cloud**, or weighted list in visual design) is a novelty visual representation of text data, typically used to depict keyword metadata (tags) on websites, or to visualize free form text.

Fig – 8.1

This is a word cloud for the Drug reviews which has a rating of 10/10.

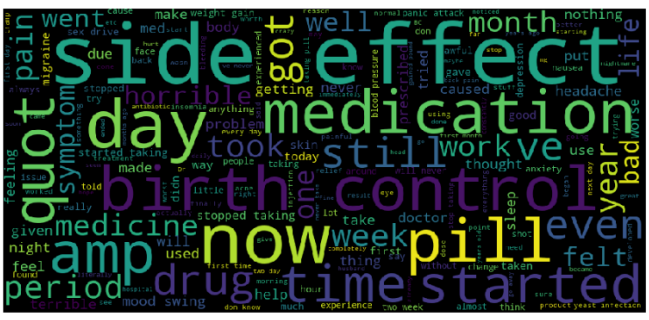

Fig – 8.2

This is a word cloud for the drug reviews with a rating of 1/10.

III. PREPROCESSING AND FEATURE ENGINEERING

A. *Preprocessing*

Drug Review Dataset contains 6 Features about the drugs given which are 'drugName', 'condition', 'rating', 'date', 'usefulcount' and the 'review' itself. The sentiments of the reviews are not given so we have to generate them on the basis of the ratings and use it as a target value which is to be predicted.

The train and test sets are merged so that we have a large combined dataset as the sentiments were not given in either of the sets. The size of the dataset is 215,063 Rows and 7 columns. The data set is then sorted on the basis of the unique ID of the drugs (data points). The Data points with the null values in any of the given features are dropped as the dropped rows were only 0.55% of the total data. The shape of the dataset after dropping is (213,869, 7). The dates given in the dataset are not in the datetime format so I have changed it to the datetime64 format for further processing. The sentiment for the reviews is given on the basis of the rating. If the rating is greater than 5 then it's a positive sentiment and if the rating is less than or equal to 5 then it's a negative sentiment.

The reviews are cleaned before the feature engineering. Regular expressions are used to clean the reviews. The reviews are changed into lower case firstly so that there's a uniformity. After analyzing the reviews, I found there's a repeating pattern "'" which is occurring in most of the reviews hence I removed it. All the Special characters are removed. Some special characters were still left hence all the non-ASCII characters are removed. Trailing and leading whitespaces are removed from the reviews. Multiple whitespaces are replaced with a single space for more clarity. The stopwords are also removed from the reviews as it'll be not very useful in the modelling. Only English stopwords are removed. The words in the review are also stemmed using the snowball stemmer. For example, the word running will be replaced with run.

B. *Feature Engineering*

There are 7 initial features given in the dataset which are 'drugName', 'condition', 'rating', 'date', 'usefulcount' and the 'review'. The heatmap of the correlation matrix of the numerical features are plotted before the feature engineering which is given in the fig-9.1.

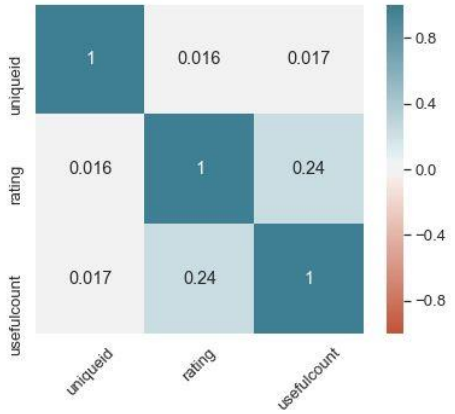

Fig – 9.1

It's plotted with seaborn. It can be inferred that the correlation between the 'usefulcount' and 'rating' is significant that is 0.24. 'uniqueID' is just the Unique ID given to each and every data point that is the consumer of the drug.

I have used textblob module to give the sentiment polarity of the review. This polarity is given to both the cleaned and uncleaned review. The interesting fact is that the correlation coefficient of the rating and the uncleaned review is 0.3481753 and with cleaned reviews is 0.23328393 hence it's greater for uncleaned review so, I have dropped the cleaned review columns and Cleaned it again but this time without removing the stopwords and stemming the words. Now the correlation coefficient of the cleaned review with the rating is 0.34600369 which is very good when compared with the last result.

The new features engineered are 'count_word' which is the number of words in each review, 'count_unique_word' which is the number of the unique words in the reviews. 'count_letters' is the letter count, 'punctuation_count' is the punctuation count, 'count_words_upper' is the upper case word count,'count_words_title' is the title case word counts, 'count_stopwords' is the number of stop words in the review, and the 'mean_word_len' is the average length of the words in the review. The date is also divided into three columns which are day, month and year for separate features for training.

A new correlation heatmap is plotted using seaborn which contains all the new features engineered and the old features. It's given in the fig-9.2

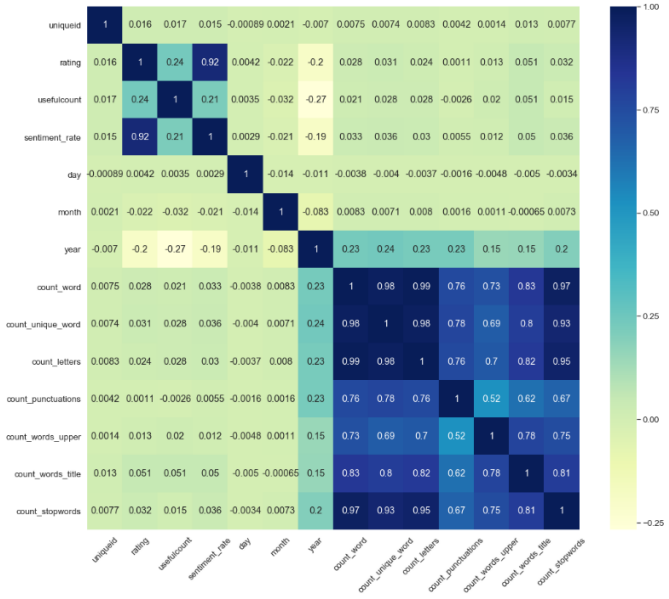

Fig – 9.2

The Label Encoder is used to change the categorical values of Drug Names and the conditions in to numerical values for the machine learning modelling. There are 3,667 unique drugs in the dataset that's why One hot encoder is not used as it would generate 3,667 new features and it would be very computationally expensive.

IV. SPLITTING DATASET AND MODELLING

The shape of the dataset after the deletion of the null values is (213,869, 7). 70% of the dataset is used for the training and the rest of the data i.e. 30% is used for the testing purpose. The shape of the training set is (149708, 15) and the shape of

the test set is (64161, 15). Three Machine learning models are trained which are LightGBM, XGBoost, and the CatBoost. The feature importance is also plotted for LightGBM and the description of these algorithms and their hyper parameters are given below.

A. LGBM

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It's designed to be distributed and efficient. It has many advantages like faster training speed and higher efficiency, lower memory usage, better accuracy and support of parallel and GPU learning, since it is based on decision tree algorithms, it splits the tree leaf wise with the best fit. The LightGBM paper uses XGBoost as a baseline and outperforms it in training speed and the dataset sizes it can handle. The parameters of the LightGBM used are,

LGBMClassifier (n_estimators=10000, learning_rate=0.10, num_leaves=30, subsample=.9, max_depth=7, reg_alpha=.1, reg_lambda=.1, min_split_gain=.01, min_child_weight=2, silent=-1, verbose=-1)

The fig-10.1 depicts the feature importance plot using the LightGBM. It can be inferred that the most importance feature is the mean word length and after that the condition of the patient. The least important feature of them all is the upper-case word count.

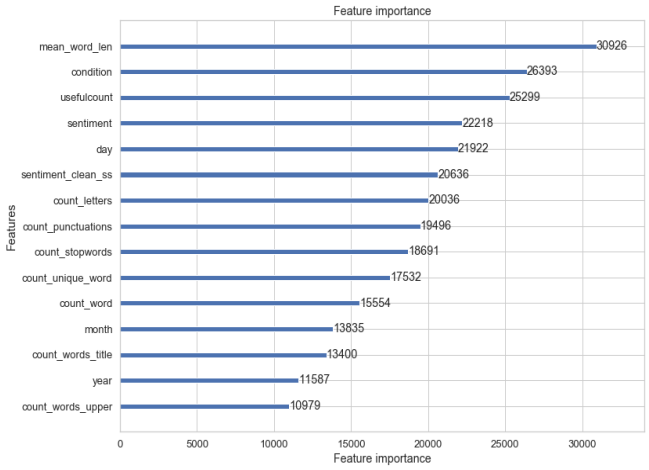

Fig – 10.1

The Classification report for the LGBM model is given below (fig-10.2), it can be seen that the precision for predicting negative sentiment is 0.85 and for the positive sentiment is 0.90. The biggest difference is between their recall value i.e. for the negative sentiment, it's is 0.76 and for the positive sentiment it's 0.94. The accuracy of the LGBM is 0.8879.

	precision	recall	f1-score	support
0	0.85	0.76	0.80	19345
1	0.90	0.94	0.92	44816
accuracy			0.89	64161
macro avg	0.88	0.85	0.86	64161
weighted avg	0.89	0.89	0.89	64161

Fig – 10.2

B. XGBoost

XGBoost stands for extreme Gradient Boosting. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting. The parameters of the XGBoost are,

XGBClassifier (n_estimators = 10000, learning_rate=0.10, num_leaves=30)

The fig-10.3 depicts the feature importance plot using the XGBoost. It can be inferred that the most important feature is the condition of the patient and it's far more important than the features following after it. The features like sentiment, usefulcount and the year are equally important for the training.

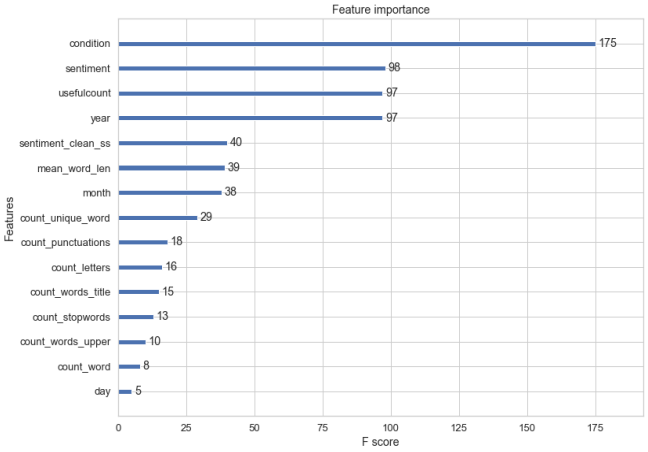

Fig – 10.3

The Classification report for the XGBoost model is given below (fig-10.4), it can be seen that the precision for predicting negative sentiment is 0.68 and for the positive sentiment is 0.78. The biggest difference is between their recall value i.e. for the negative sentiment, it's is 0.38 and for the positive sentiment it's 0.92. The accuracy of the XGBoost is 0.7582. The XGBoost is not the best model for this sentiment analysis.

	precision	recall	f1-score	support
0	0.68	0.38	0.49	19345
1	0.78	0.92	0.84	44816
accuracy			0.76	64161
macro avg	0.73	0.65	0.66	64161
weighted avg	0.75	0.76	0.73	64161

Fig – 10.4

C. CatBoost

CatBoost is an algorithm for gradient boosting on decision trees. It is developed by Yandex researchers and engineers, and is used for search, recommendation systems, personal assistant, self-driving cars, weather prediction and many other tasks at Yandex and in other companies, including CERN, Cloudflare, Careem taxi. It is in open-source. The parameters of the CatBoost are,

CatBoostClassifier (iterations = 10000, learning_rate = 0.5)

The Classification report for the CatBoost model is given below (fig-10.5), it can be seen that the precision for predicting negative sentiment is 0.83 and for the positive sentiment is 0.90. The Recall for the negative sentiment is 0.77 and for the positive sentiment it's 0.93. The accuracy of the CatBoost is 0.8837. The performance of the LGBM and

CatBoost is very comparable to each other, according to their classification report and accuracy.

	precision	recall	f1-score	support
0	0.83	0.77	0.80	19345
1	0.90	0.93	0.92	44816
accuracy			0.88	64161
macro avg	0.87	0.85	0.86	64161
weighted avg	0.88	0.88	0.88	64161

Fig – 10.5

V. Conclusion

The main aim was to predict the sentiment of the drug reviews given by the patients. Hence Exploratory Data Analysis was done to get more insight about the dataset and preprocessing was done to get the data ready for both the modelling and EDA. Initially 7 features were given, hence feature engineering was done on the basis of the EDA and reviews by the patients. The reviews were cleaned and features are generated. The features were generated by both the cleaned and uncleaned reviews. In the Machine Learning modelling, three classification models were trained which were LightGBM, XGBoost, and the CatBoost. The best performing model is the LGBM Classifier but it's accuracy and the classification report are comparable to the CatBoost Classifier. The accuracies were 0.8879 and 0.8837 respectively. The features importance is also plotted for LGBM and CatBoost. The classification report is also there for deeper analysis of the model as only accuracies doesn't tell much about a classification model.

References

[1] Aspect-Based Sentiment Analysis of Drug Reviews Applying Cross-Domain and Cross-Data Learning *(references)* https://dl.acm.org/doi/10.1145/3194658.3194677

[2] Research Paper Template https://www.ieee.org › web › org › conferences › Conference-template-A4

[3] CatBoost Open Source Library Documentation https://catboost.ai/

[4] XGBoost Documentation https://xgboost.readthedocs.io/en/latest

[5] Kaggle https://www.kaggle.com/jessicali9530/kuc-hackathon-winter-2018

[6] UCI Library Drug Review Dataset (Drug.com) https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29

[7] Drug names and their description https://www.drugs.com/drug_information.html

[8] LGBM Documentation https://lightgbm.readthedocs.io/en/latest

[9] TextBlob documentation https://textblob.readthedocs.io/en/dev

[10] Word Cloud or Tag Cloud https://en.wikipedia.org/wiki/Tag_cloud