INTERNSHIP: PROJECT REPORT

Dear Intern

Project report is an inherent component of your internship. We are enclosing a reference table of content for the project report. Depending on the internship project (IT/Non-IT, Technical/Business Domain), you may choose to include or exclude or rename sections from the table of content mentioned below. You can also add additional sections. The key objective of this report is for you to systemically document the project work done.

Internship Project Title	Classification Model - Build a Model that Classifies the Side Effects of a Drug
Name of the Company	TCS iON
Name of the Industry Mentor	Himalaya Aashish
Name of the Institute	ICT Academy of Kerala

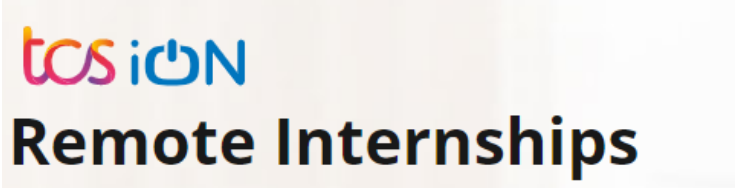
Start Date	End Date	Total Effort (hrs.)	Project Environment	Tools used
21-06-2021	20-9-2021	125	Python	Google Colab

Classification of side effect of a Drug-
Analysis & Modelling with Supervised Learning & Sentiment Analysis

A project report submitted to TCS iON

submitted by

Divya Sadanandan



July 2021

I. OBJECTIVE

Drugs are chemical substances for treating diseases, but may induce adverse reactions or side effects. Drug discovery is time-consuming and labor-intensive, and candidate drugs suffer from potential side effects. As far as we know, lots of approved drugs were withdrawn from the market because of unexpected side effects. Since drug side effects are great concern of the public health, the identification of drug side effects helps to reduce risks in drug discovery. With the increase of drug data, researchers collected information about approved drugs, and identified potential side effects of new candidate drugs. Traditional methods analyzed the drug structure-activity relationship or drug quantitative structure-property relationship. In recent years, machine learning methods were applied to the drug side effect prediction.

II. INTRODUCTION

The objective of this project is to build a classification model that classifies the side effect of a particular drug by age, gender and race.

The dataset in the center of this project provides user reviews on specific drugs along with related conditions they are used for, and ratings reflecting overall patient satisfaction. It also has columns for the medication's side effects that can possibly occur, data about the patient's age, date it was reviewed and also two other types of rating: effectiveness and ease of use.

The details of the dataset selected for the project is as follows:

Name of the dataset	WebMD.csv
Reference link for the dataset	https://www.kaggle.com/rohanharode07/webmd-drug-reviews-dataset
Total no: of instances	362806
Total no: of attributes	12
	1. Drug (categorical): name of drug 2. DrugId (numerical): drug id 3. Condition (categorical): name of condition 4. Review (text): patient review 5. Side (text): side effects associated with drug (if any) 6. EaseOfUse (numerical): 5-star rating 7. Effectiveness (numerical): 5-star rating 8. Satisfaction (numerical): 5-star rating 9. Date (date): date of review entry 10. Useful Count (numerical): number of users who found review useful. 11. Age (numerical): age group range of user 12. Sex (categorical): gender of user
Missing values	37 missing values under ‘Review ‘column

III. INTERNSHIP ACTIVITIES

1. Importing libraries and loading the dataset

2. Feature Description
   - Listing the data head
   - Listing the tail of the dataset
   - Check the shape of the dataset
   - Checking the datatypes in the dataset
   - Checking the column wise unique values

3. Exploratory Data Analysis
   - Distribution of age in the dataset -bar graph
   - The top 50 drugs in the data set -bar graph.
   - Top 30 conditions that people faced -bar graph.
   - Top 20 the number of drugs per condition- bar graph
   - Grouping the drug ID based on the conditions treated and viz.
   - Distribution of effectiveness in the data set
   - Pie chart showing the distribution of satisfaction level
   - Count plot showing the distribution of sex
   - Word cloud showing the most occurring words in the review column
   - Word cloud for reviews with high satisfaction ratings
   - Word cloud for reviews with low satisfaction ratings

4. Data preprocessing
   - Checking missing values.
   - Dropping the rows with null values in reviews

5 Performing sentiment analysis on target – 'Reviews'
   - Text preprocessing using TextBlob library- removing stop words, punctuations, convert into lower cases, lemmatize, spell check.
   - Generating positive and negative sentiments based on the polarity score obtained through VADER sentiment analysis

6 Encoding
   - Replacing the positive reviews with 1 and negative reviews with 0
   - Converting categorical data to numeric in the features with pandas factorize

7 Splitting the data set into features and target

8 Applying Standard scaler to features

9 Splitting the data set into training set and testing set

10 Creating the model

- Decision Tree Classifier
- Random Forest Classifier
- K - Nearest Neighbor Classifier
- Gradient Boosting Classifier
- XG Boost Classifier
- Support Vector Machine Classifier
- Logistic Regression
- Gaussian Naive Bayes Classifier
- Bernoulli Naive Bayes Classifier

12. Evaluating the Model

IV. APPROACH OR METHODOLOGY

1. Importing libraries and loading the dataset

The dataset is loaded after importing the necessary libraries.

The snapshot of the dataset is as follows:

	Age	Condition	Date	Drug	DrugId	EaseofUse	Effectiveness	Reviews	Satisfaction	Sex	Sides	UsefulCount
0	75 or over	Stuffy Nose	9/21/2014	25dph-7.5peh	146724	5	5	I'm a retired physician and of all the meds I ...	5	Male	Drowsiness, dizziness , dry mouth /nose/thro...	0
1	25-34	Cold Symptoms	1/13/2011	25dph-7.5peh	146724	5	5	cleared me right up even with my throat hurtin...	5	Female	Drowsiness, dizziness , dry mouth /nose/thro...	1
2	65-74	Other	7/18/2012	warfarin (bulk) 100 % powder	144731	2	3	why did my PTINR go from a normal of 2.5 to ov...	3	Female		0
3	75 or over	Other	9/23/2010	warfarin (bulk) 100 % powder	144731	2	2	FALLING AND DON'T REALISE IT	1	Female		0
4	35-44	Other	1/8/2009	warfarin (bulk) 100 % powder	144731	1	1	My grandfather was prescribed this medication ...	1	Male		1

2. Feature Description

The various features of the data set such as the shape, datatypes, unique values are being captured in this step.

```
df.shape
```

```
(362806, 12)
```

```
df.dtypes
```

```
Age              object
Condition        object
Date             object
Drug             object
DrugId            int64
EaseofUse         int64
Effectiveness     int64
Reviews          object
Satisfaction      int64
Sex              object
Sides            object
UsefulCount       int64
dtype: object
```

```
df.nunique()
```

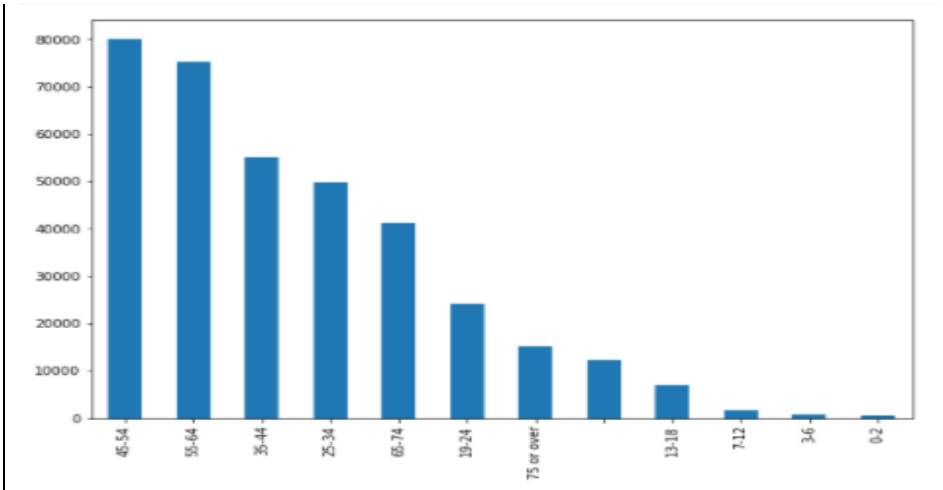
```
Age                12
Condition        1806
Date             4524
Drug             7093
DrugId           6572
EaseofUse           7
Effectiveness       7
Reviews        250167
Satisfaction        7
Sex                 3
Sides            1651
UsefulCount       148
dtype: int64
```

3. Data Visualizations/EDA

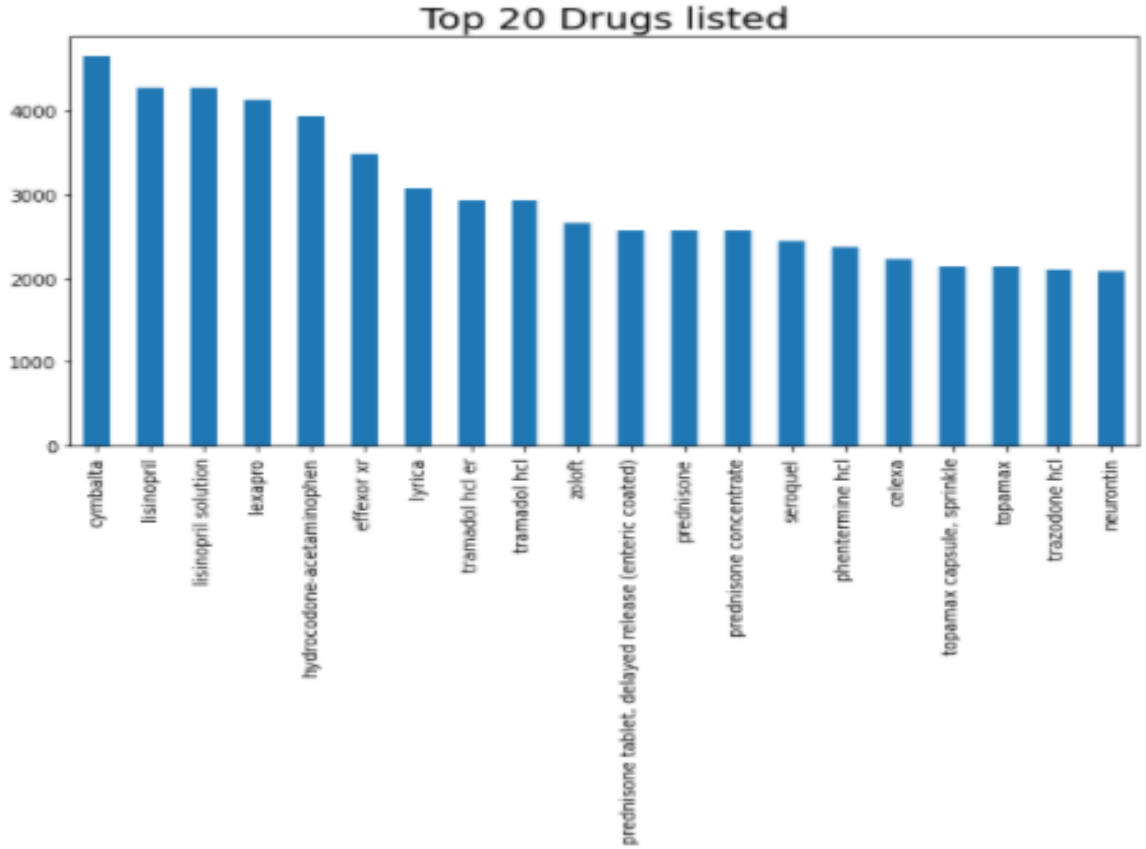
- **Distribution of age in the dataset**

```
df['Age'].value_counts()
```

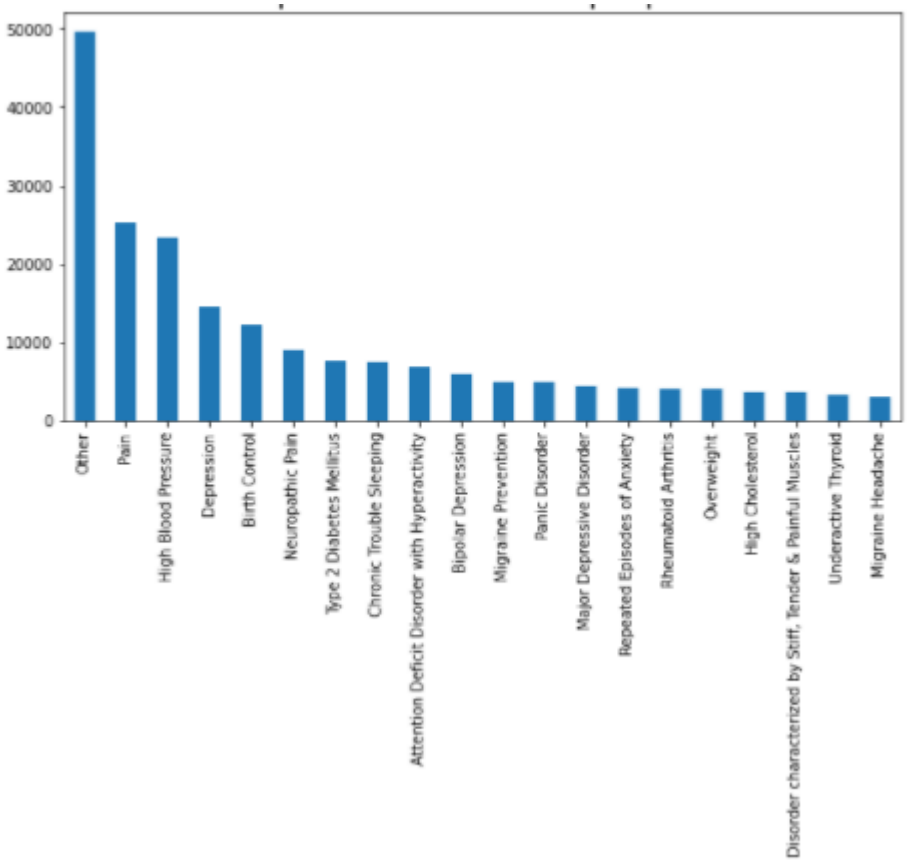
```
45-54          80043
55-64          75136
35-44          55011
25-34          49718
65-74          41216
19-24          24230
75 or over     15226
               12202
13-18           7045
7-12            1644
3-6              838
0-2              497
Name: Age, dtype: int64
```



- **The top 20 drugs in the data set -bar graph**



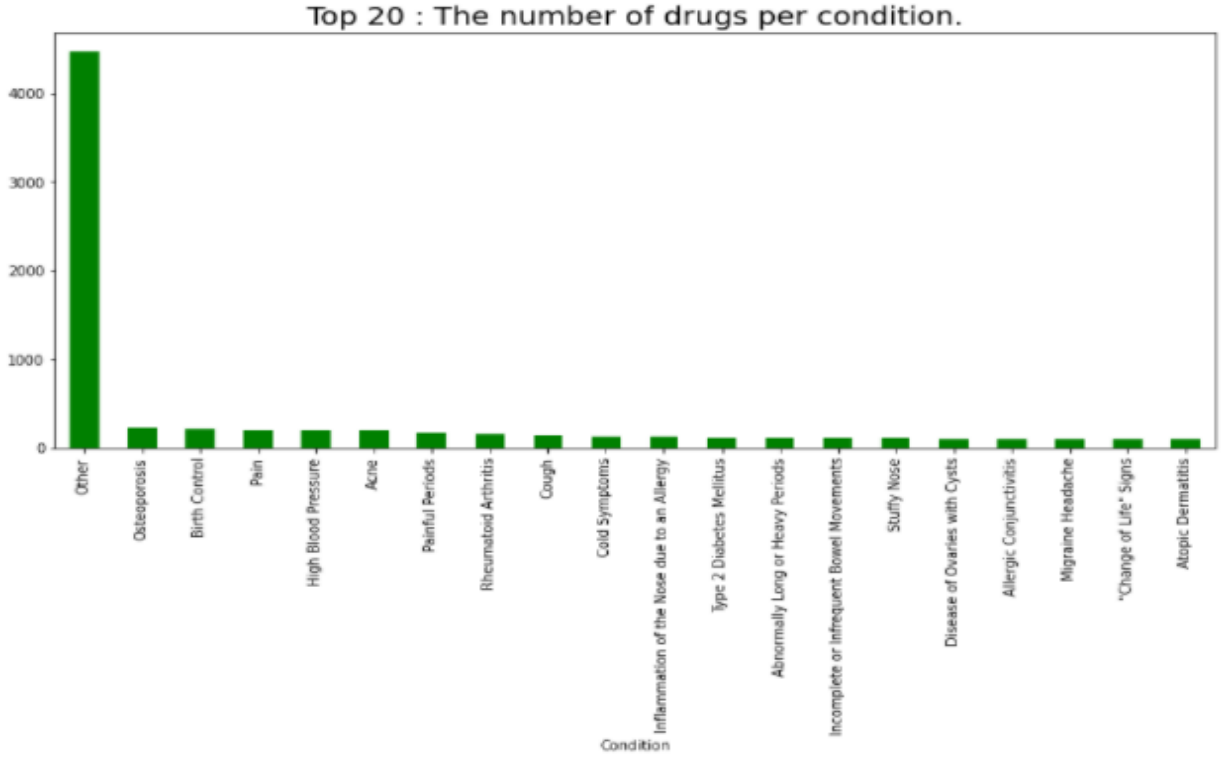
- **Top 20 conditions listed -bar graph.**



The topmost condition is not listed. Or multiple conditions could be treated with several other drugs.

The length of the dataset is 362806 and there are 6572 unique drug ID's & 1806 conditions listed

Top 20 the number of drugs per condition- bar graph

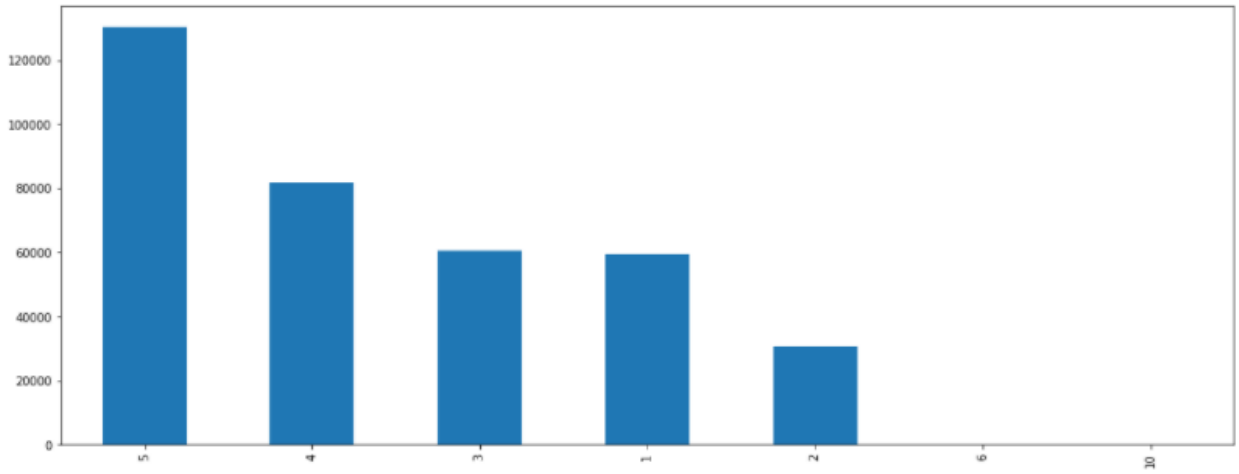


- **Grouping the drug ID based on the conditions treated and viz.**

Here we can see that a single drug is used to treat multiple conditions	Drug prednisone tablet, delayed release (enteric coated) 92 prednisone concentrate 92 prednisone 92 cipro suspension, microcapsule reconstituted 59 cipro 59 methylprednisolone 52 ciprofloxacin 48 levaquin 46 kenalog-40 vial 43 doxycycline hyclate tablet, delayed release (enteric coated) antimalarial drugs 42 doxycycline hyclate tablet tetracyclines 42 azithromycin tablet macrolide antibiotics 40 azithromycin tablet 40 azithromycin packet macrolide antibiotics 40 azithromycin 40 metronidazole 39 levofloxacin solution 37 levofloxacin 37 doxycycline calcium syrup 36 amoxicillin tablet, chewable 34 Name: Condition, dtype: int64

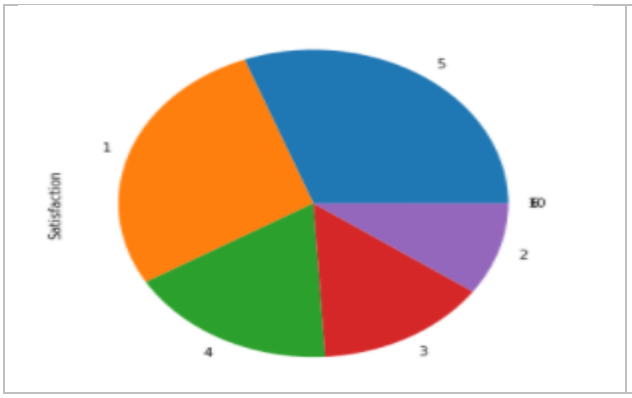
Here we can also find that multiple drugs are being used to treat a certain condition.	Condition Other 4469 Osteoporosis 229 Birth Control 204 Pain 202 High Blood Pressure 200 Acne 197 Painful Periods 161 Rheumatoid Arthritis 148 Cough 137 Cold Symptoms 129 Inflammation of the Nose due to an Allergy 122 Type 2 Diabetes Mellitus 117 Abnormally Long or Heavy Periods 111 Incomplete or Infrequent Bowel Movements 109 Stuffy Nose 108 Disease of Ovaries with Cysts 103 Allergic Conjunctivitis 102 Migraine Headache 95 "Change of Life" Signs 92 Atopic Dermatitis 92 Name: Drug, dtype: int64

- **Distribution of effectiveness in the data set**



We can observe that most of the drugs were effective from the effective rating distribution

- **Pie chart showing the distribution of satisfaction level**

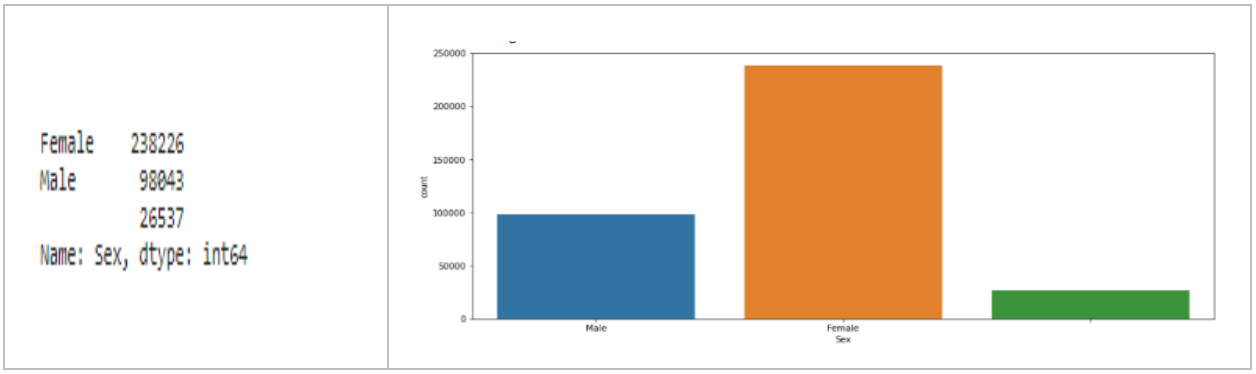



```
df['Satisfaction'].value_counts()
```

```
5     111550
1     100901
4      63158
3      51852
2      35342
6          2
10         1
Name: Satisfaction, dtype: int64
```

There is an equal distribution between users who gave minimum rating 1 and maximum rating 5 .

- **Count plot showing the distribution of sex**



Majority of the users were of female gender in the dataset. Unspecified gender field is also observed.

4. Text Preprocessing on the target- 'Reviews'

- **Text Preprocessing (TextBlob)**
  Text preprocessing is done by using the TextBlob Library.

  TextBlob is built upon NLTK and provides an easy-to-use interface to the NLTK library. various tasks can be performed like part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

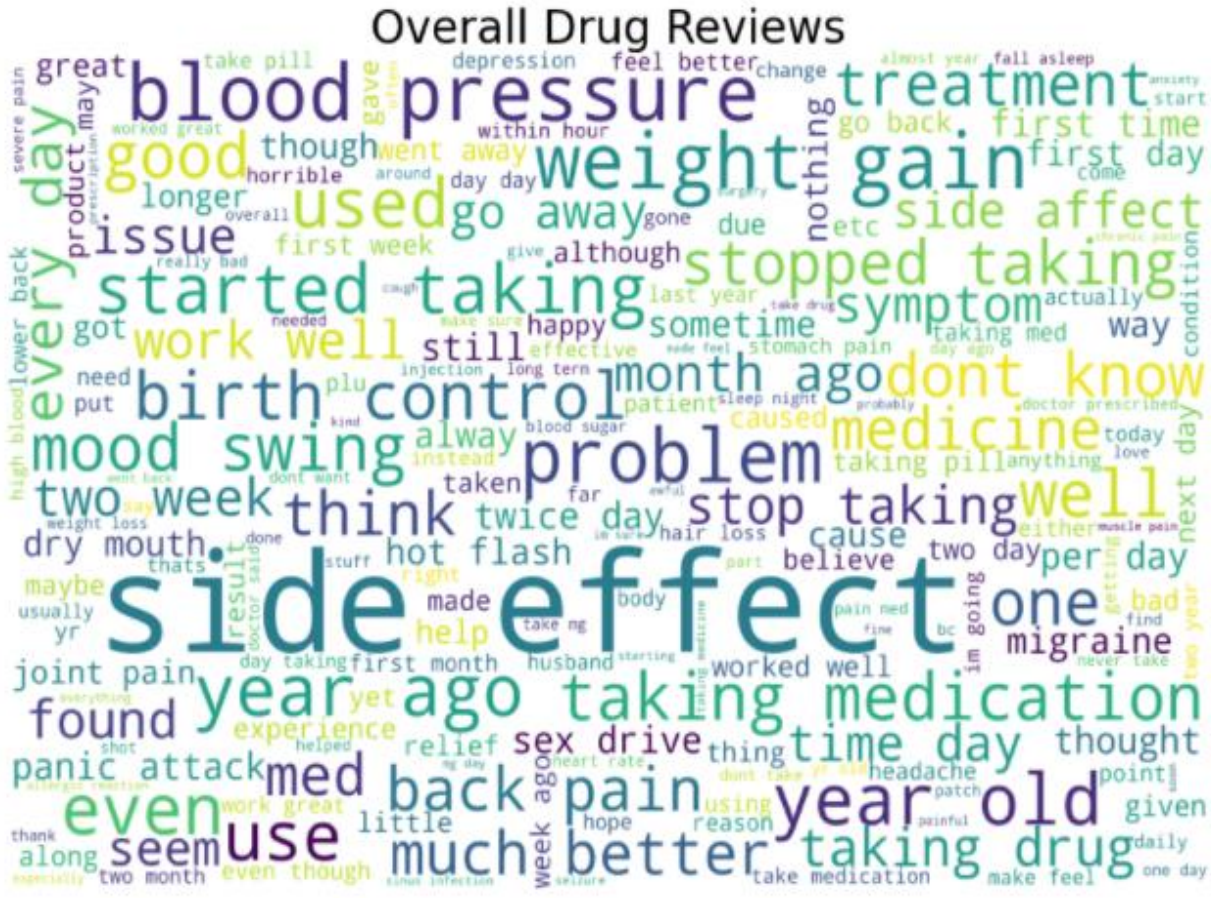
  The following steps were performed on the 'Reviews' column
  - The text is converted to lower case.
  - Stop words and punctuations were removed
  - Lemmatization is performed.
  - A Wordcloud is generated for the 'Review'.

  A Wordcloud (or Tag cloud) is a visual representation of text data. It displays a list of words, the importance of each being shown with font size or color. This format is useful for quickly perceiving the most prominent term. This represents the essence of the data. It displays the most repeated words in the column such that the word with the highest count appears in larger font. Less popular words are shown in small fonts.

  Therefore, the wordcloud is generated for the target 'Reviews'.

- **Word cloud showing the most occurring words in the review column**

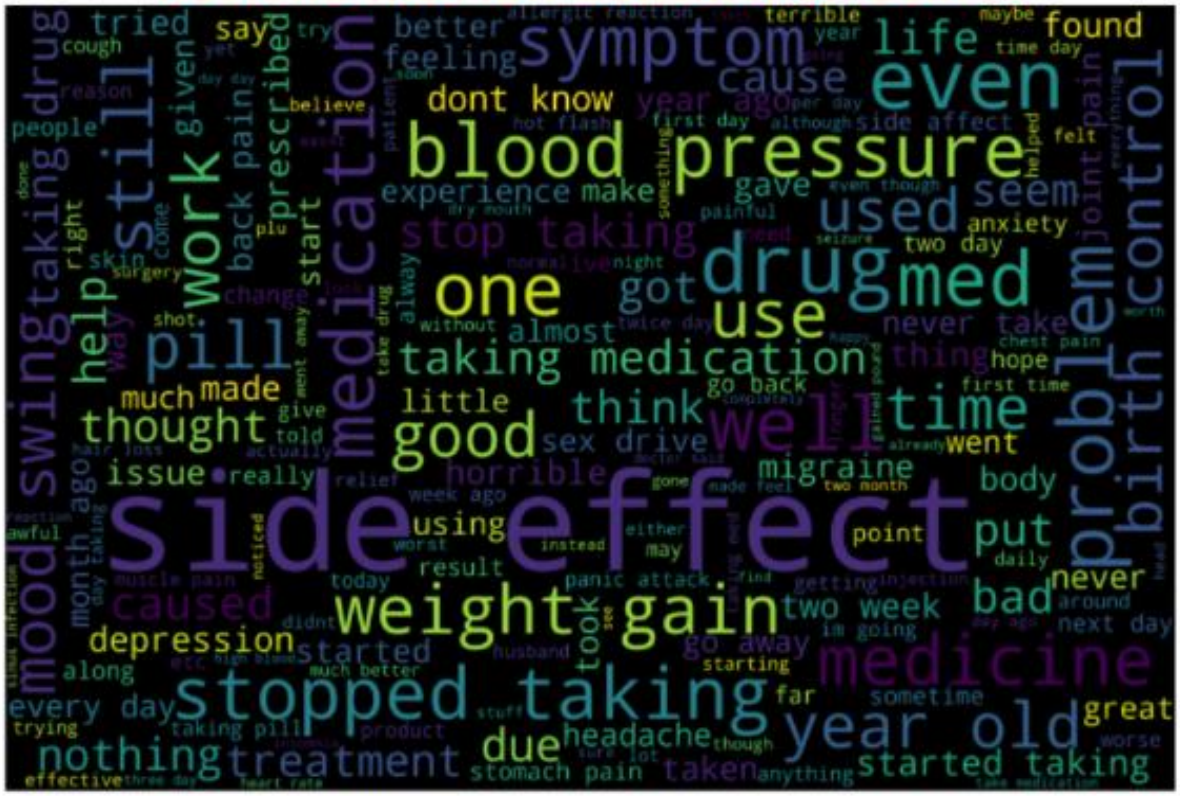
Review being the target and since it is in the form of textual data, we could try finding the most occurred words in the column by generating the word cloud.



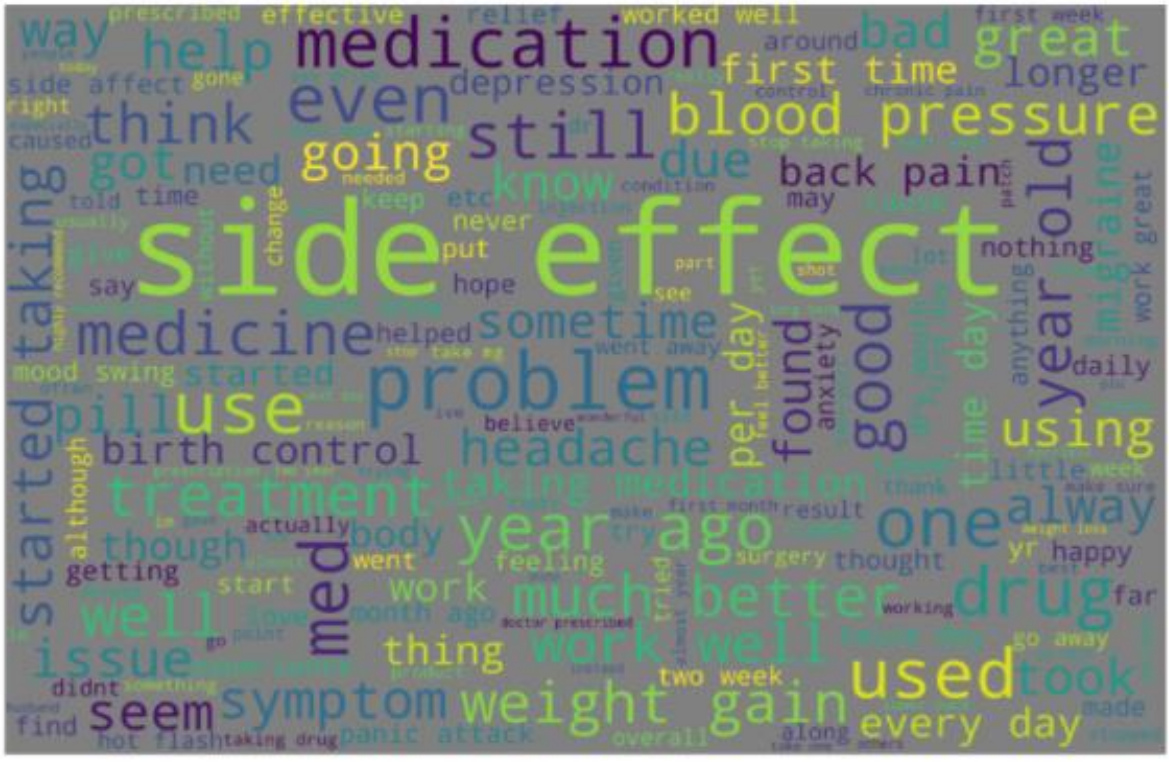
WordCloud of Reviews

In order to check if there is a relation between the satisfaction level and the reviews the word cloud is been generated for reviews made by the highly satisfied users and the users who had given low satisfaction score.

Reviews with low Satisfaction Scores



Reviews with Positive Satisfaction Scores



5. Sentiment Analysis with VADER

- **VADER**

VADER Sentiment Analysis. VADER (Valence Aware Dictionary and sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only talks about the Positive and Negative score but also tells us about how positive or negative a sentiment is.

Polarity of the sentences were generated for the 'Reviews' column.

	neg	neu	pos	compound
0	0.124	0.734	0.142	0.1027
1	0.243	0.631	0.126	-0.3182
2	0.000	1.000	0.000	0.0000
3	0.444	0.556	0.000	-0.1531
4	0.086	0.771	0.143	0.3818

The above data frame was generated that shows the negative, neutral, positive and the compounded score for the sentiments.

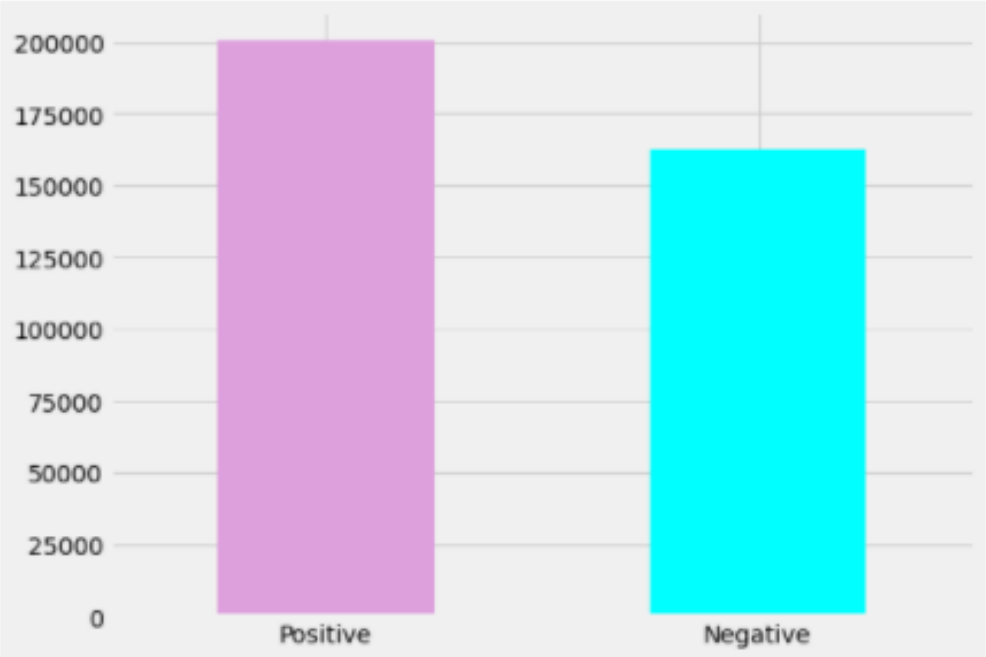
Converting the scores into positive and negative sentiments based on the compounded values.

- **Positive Sentiment: compound score >= 0**
- **Negative Sentiment: compound score < 0**

Negative compounded value shows negative sentiments &

Positive compounded value shows positive sentiments.

A bar graph is plotted to show the overall distribution of positive and negative sentiments in Reviews




```
df_c['Sentiment'].value_counts()
```

```
Positive    200428
Negative    162341
Name: Sentiment, dtype: int64
```

We can see that there are 200428 positive reviews & 162341 negative reviews.

The positive reviews could be possibly given by users who might not have had any side effects & negative sentiments could be given by users who had side effects/difficulties with the drug.

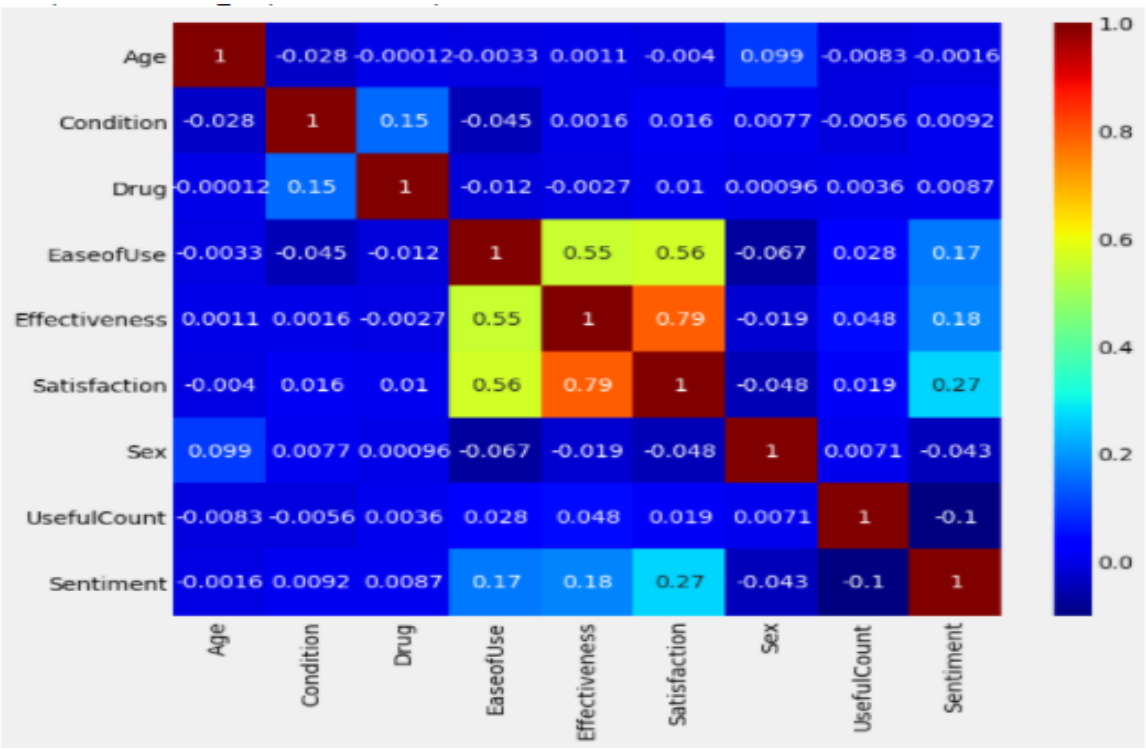
6. Encoding

- **Replacing the positive reviews with 1 and negative reviews with 0.**
- **Converting categorical data to numeric in the features**
  Using pandas factorize the categorical features Age, Condition, Drug & Sex is converted to numerical values

	Age	Condition	Drug	EaseofUse	Effectiveness	Satisfaction	Sex	UsefulCount	Sentiment
0	0	0	0	5	5	5	0	0	1
1	1	1	0	5	5	5	1	1	0
2	2	2	1	2	3	3	1	0	1
3	0	2	1	2	2	1	1	0	0
4	3	2	1	1	1	1	0	1	1

The encoded Dataframe

A heatmap is plotted with the numerical columns to check for correlation



7. Splitting the dataset into features and target

The dataset is then split into features and target
Columns such as 'Date','DrugId', 'Reviews','Sides','neg', 'neu', 'pos', 'compound' & Sentiments were dropped from Feature's 'X'

Target 'y' = 'Reviews'

8. Applying Standard scaler to features

- **Standard scaling**

Standardization scales each input variable separately by subtracting the mean (called centering) and dividing by the standard deviation to shift the distribution to have a mean of zero and a standard deviation of one.

After checking the statistical data, Standard Scaling is performed on features (X)

	0	1	2	3	4	5	6	7
count	3.627690e+05	3.627690e+05	3.627690e+05	3.627690e+05	3.627690e+05	3.627690e+05	3.627690e+05	3.627690e+05
mean	6.688582e-15	7.261534e-14	-8.332813e-14	8.814549e-16	-7.796846e-15	-1.479807e-14	-3.153879e-14	-4.506013e-15
std	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00
min	-1.871698e+00	-7.955879e-01	-1.471110e+00	-2.291706e+00	-1.739086e+00	-1.325390e+00	-1.454956e+00	-7.234226e-01
25%	-8.381886e-01	-6.803718e-01	-8.923612e-01	-7.739390e-01	-3.654331e-01	-1.325390e+00	-1.454956e+00	-6.161781e-01
50%	1.953209e-01	-3.643507e-01	-3.514902e-02	7.438279e-01	3.213933e-01	-8.401710e-02	3.571589e-01	-2.944446e-01
75%	7.120757e-01	1.755189e-01	8.196913e-01	7.438279e-01	1.008220e+00	1.157355e+00	3.571589e-01	2.417777e-01
max	3.812605e+00	5.142977e+00	1.893223e+00	4.538245e+00	4.442352e+00	4.260787e+00	2.169273e+00	2.662392e+01

9. Splitting the data set into training and testing set

Here, since the dataset is large, there is no requirement to have a k-fold or similar cross-validation techniques.The data set is being split to training and testing set with test size 0.2 and random state 101

```
X_train, X_test, y_train, y_test = train_test_split(X, y,random_state=101,test_size=0.2)
```

10. Creating the model

Multiple Machine Algorithms were used to create the models & the performance of the model is evaluated with evaluation metrics like precision, recall, f1 score & confusion matrix

- Decision Tree Classifier
- Random Forest Classifier
- K - Nearest Neighbor Classifier
- Gradient Boosting Classifier
- XGBoost Classifier
- Support Vector Machine Classifier
- Logistic Regression
- Gaussian Naive Bayes Classifier
- Bernoulli Naive Bayes Classifier

V. ASSUMPTIONS

The objective of this project is to build a classification model that classifies the side effect of a particular drug by age, gender and race.

The dataset (webMD.csv) for this project provides user reviews on specific drugs.

The column 'Sides' at a glance appears as the column that details the side effects occurred in users.

But while taking a closer look in to the data the 'Sides' column details the side effect that may occur and not the side effects that had occurred after the drug usage.

Sides
Drowsiness, dizziness , dry mouth /nose/throat, headache , upset stomach , constipation , or trouble sleeping may occur.
Drowsiness, dizziness , dry mouth /nose/throat, headache , upset stomach , constipation , or trouble sleeping may occur.

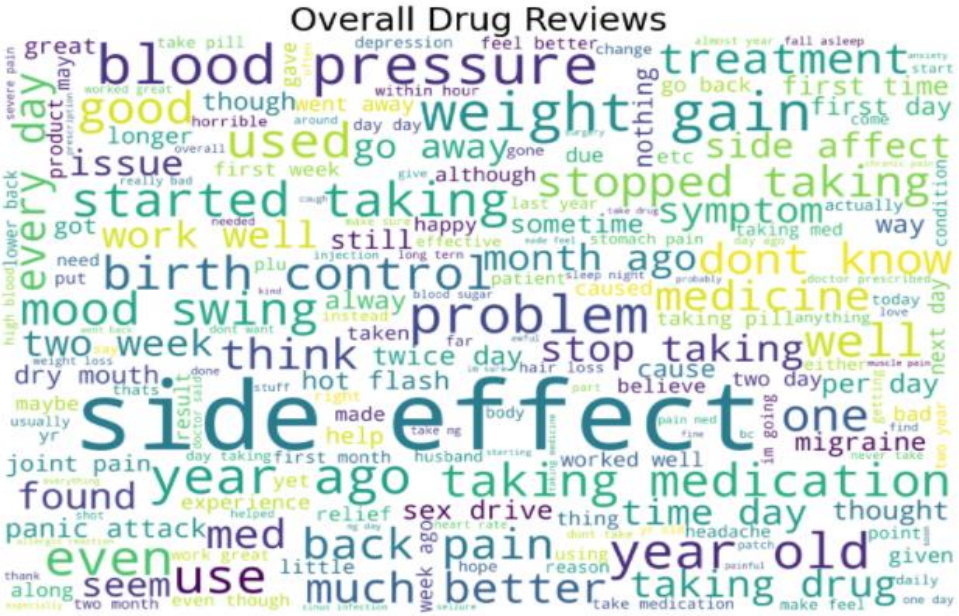
Hence, we could consider the target variable as 'Reviews'

As a result of sentiment analysis performed on 'Reviews' we see that there are 200428 positive reviews & 162341 negative reviews.

Assumption

The positive reviews could be possibly given by users who might not have had any side effects using the drug & negative sentiments could be given by users who had side effects/difficulties with the drug.

We also can infer from the word cloud generated on the 'Reviews' that the data mostly speaks about the side effects.



VI. ALGORITHMS

• Logistic Regression

In Logistic regression, it is used to model the probability of a finite number of outcomes, typically two. In essence, a logistic equation is
created in such a way that the output values can only be between 0 and 1.

• k Nearest Neighbors Algorithm(kNN)

K Nearest Neighbours is a basic algorithm that
stores all the available and predicts the classification of unlabelled data based on a similarity measure. Like linear geometry when two parameters are plotted on the 2D Cartesian system and we identify the similarity measure by calculating the distance between the points, the same applies here, KNN algorithm works on the assumption those similar things exist in proximity, simply we can put into the same things stay close to each other.

• Decision Tree Classifier Algorithm

This is an ensemble method. Decision Tree is a tree-like
graph where sorting starts from the root node to the leaf node until the target is achieved. It is the most popular one for decision and classification based on supervised algorithms. It is constructed by recursive partitioning where each node acts as a test case for some attributes and each edge, deriving from the node, is a possible answer in the test case. Both the root and leaf nodes are two entities of the algorithm. Decision Tree Analysis is done via an algorithmic approach where a data set is split into subsets as per conditions. The name itself says it is a tree-like model in the form of if-then-else statements. The deeper is the tree and more are the nodes, the better is the model.

• Random Forest Classifier Algorithm

The random forest algorithm is based on supervised
learning. It can be used for both regression and classification problems. It can be viewed as a collection of multiple decision trees algorithms with random sampling. Random forest is a combination of Breiman's "bagging" idea and random selection of features. The idea is to make the prediction precise by taking the average or mode of the output of multiple decision trees. The greater the number of decision trees is considered the more precise output will be. It also comes under ensemble methods.

• Gradient Boosting Algorithm

This is also an ensemble method which performs
Boosting, a special type of Ensemble Learning technique that works by combining several predictors with poor accuracy into a model with strong accuracy. This works by each model paying attention to its predecessor's mistakes. There is an improved version to this which is the XGBoost.

• XGBoost Algorithm

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. XGBoost falls under the category of Boosting techniques in Ensemble Learning. In Boosting technique, the errors made by previous models are tried to be corrected by succeeding models by adding some weights to the models.

• Naive-Bayes Classifier Algorithm

This is a probabilistic classifier which is used when each of
the features are independent. The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

• Support vector machine (SVM) Algorithm

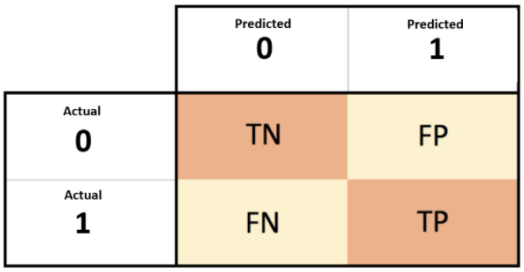
This is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text. Compared to newer algorithms like neural networks, they have two main advantages: higher speed and better performance with a limited number of samples (in the thousands). This makes the algorithm very suitable for text classification problems, where it's common to have access to a dataset of at most a couple of thousands of tagged samples.

VII. METRICS FOR EVALUATION

The predicted values of output for the test data are obtained and the efficiency of classification is studied in terms of different performance metrics. The performance parameters to evaluate the classifier models are;

• Confusion Matrix

Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.



Confusion Matrix
There are 4 important terms:
True Positives (TP): The cases in which we predicted YES and the actual output was also YES.
True Negatives (TN): The cases in which we predicted NO and the actual output was NO.
False Positives (FP): The cases in which we predicted YES and the actual output was NO.
False Negatives (FN): The cases in which we predicted NO and the actual output was YES.

• Accuracy score

It is the ratio of number of correct predictions to the total number of input samples. It works well only if there are an equal number of samples belonging to each class.If you are working on a classification problem, the best score is 100% accuracy. If you are working on a regression problem, the best score is 0.0 error.

$$Accuracy = \frac{TruePositive + TrueNegative}{TotalSample}$$

• Precision score

It is the number of correct positive results divided by the number of

positive results predicted by the classifier. In the simplest terms, Precision is the ratio between the True Positives and all the Positives.

$$\frac{TruePositives}{TruePositives + FalsePositives}$$

- **Recall score**

It is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

$$\frac{TruePositives}{TruePositives + FalseNegatives}$$

- **F1 score**

F1 Score is used to measure a test's accuracy. It is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances). The greater the F1 Score, the better is the performance of our model.

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Testing of Algorithms

Once a model is fitted on to the training data, it is then evaluated based on the test data. The performance metrics for the classifier is estimated and it gives the best model for the given dataset. The performance metrics for the different algorithms are summarised below. The performance of the model is expressed as a summary using the Classification Report which is obtained using the classification_report() function.

ALGORITHM	CLASSIFICATION REPORT
Decision Tree Classifier Model	CPU times: user 1.77 s, sys: 10.5 ms, total: 1.78 s Wall time: 1.78 s Confusion Matrix for Decision Tree: [[21191 11372] [11905 28086]] Score: 67.92 Classification Report: precision recall f1-score support 0 0.64 0.65 0.65 32563 1 0.71 0.70 0.71 39991 accuracy 0.68 72554 macro avg 0.68 0.68 0.68 72554 weighted avg 0.68 0.68 0.68 72554
Random Forest Classifier Model	CPU times: user 47.3 s, sys: 2.87 s, total: 50.2 s Wall time: 50.1 s Confusion Matrix for Random Forest Classifier: [[21689 10874] [9511 30480]] Score: 71.9 Classification Report: precision recall f1-score support 0 0.70 0.67 0.68 32563 1 0.74 0.76 0.75 39991 accuracy 0.72 72554 macro avg 0.72 0.71 0.71 72554 weighted avg 0.72 0.72 0.72 72554
XGBoost Classifier Model	CPU times: user 13.7 s, sys: 122 ms, total: 13.9 s Wall time: 13.9 s Confusion Matrix for XGBoost Classifier: [[16884 15679] [9400 30591]] Score: 65.43 Classification Report: precision recall f1-score support 0 0.64 0.52 0.57 32563 1 0.66 0.76 0.71 39991 accuracy 0.65 72554 macro avg 0.65 0.64 0.64 72554 weighted avg 0.65 0.65 0.65 72554
Gradient Boosting Classifier Model	CPU times: user 37.7 s, sys: 129 ms, total: 37.9 s Wall time: 37.8 s Confusion Matrix for Gradient Boosting Classifier: Score: 65.49 Classification Report: precision recall f1-score support 0 0.64 0.52 0.58 32563 1 0.66 0.76 0.71 39991 accuracy 0.65 72554 macro avg 0.65 0.64 0.64 72554 weighted avg 0.65 0.65 0.65 72554

Gaussian Naive Bayes Classifier Model	CPU times: user 71.6 ms, sys: 940 µs, total: 72.5 ms Wall time: 74.6 ms Confusion Matrix for Gaussian Naive Bayes Classifier: [[15240 17323] [9923 30068]] Score: 62.45 Classification Report: precision recall f1-score support 0 0.61 0.47 0.53 32563 1 0.63 0.75 0.69 39991 accuracy 0.62 72554 macro avg 0.62 0.61 0.61 72554 weighted avg 0.62 0.62 0.62 72554
Bernoulli Naive Bayes Classifier Model	CPU times: user 116 ms, sys: 4.96 ms, total: 121 ms Wall time: 112 ms Confusion Matrix for Bernoulli Naive Bayes Classifier: [[37767 22395] [20781 27888]] Score: 60.33 Classification Report: precision recall f1-score support 0 0.65 0.63 0.64 60162 1 0.55 0.57 0.56 48669 accuracy 0.60 108831 macro avg 0.60 0.60 0.60 108831 weighted avg 0.60 0.60 0.60 108831
KNN Classifier Model	Confusion Matrix for K Neighbors Classifier: [[19715 12848] [11725 28266]] Score: 66.13 Classification Report: precision recall f1-score support 0 0.63 0.61 0.62 32563 1 0.69 0.71 0.70 39991 accuracy 0.66 72554 macro avg 0.66 0.66 0.66 72554 weighted avg 0.66 0.66 0.66 72554
Logistic Regression Model	CPU times: user 799 ms, sys: 452 ms, total: 1.25 s Wall time: 679 ms Confusion Matrix for Logistic Regression: [[16963 15600] [10396 29595]] Score: 64.17 Classification Report: precision recall f1-score support 0 0.62 0.52 0.57 32563 1 0.65 0.74 0.69 39991 accuracy 0.64 72554 macro avg 0.64 0.63 0.63 72554 weighted avg 0.64 0.64 0.64 72554

Support Vector Machine Classifier Model	CPU times: user 4h 58min 36s, sys: 39.4 s, total: 4h 59min 15s Wall time: 4h 57min 54s Confusion Matrix for Support Vector Machines: [[15346 17217] [8529 31462]] Score: 64.51 Classification Report: precision recall f1-score support 0 0.64 0.47 0.54 32563 1 0.65 0.79 0.71 39991 accuracy 0.65 72554 macro avg 0.64 0.63 0.63 72554 weighted avg 0.64 0.65 0.64 72554

VIII. OUTCOME

Listed below are the comparison of the results from different algorithms, ordered by accuracy:

Random Forest Classifier = 72%

Decision Tree Classifier = 68%

K - Nearest Neighbor Classifier = 66.13%

Gradient Boosting Classifier = 65.49%

XGBoost Classifier = 65.43%

Support Vector Machine Classifier = 65%

Logistic Regression = 64.17%

Gaussian Naive BAyes Classifier = 62.45%

Bernoulli Naive Bayes Classifier = 61.07%

If we order the different models by the values for accuracy as metrics, we can see that Random Forest Classifier gives the best results of 72% accuracy.

SVM has had the longest execution time (4h 48min).

IX. ENHANCEMENT SCOPE

In the recent years, we have observed a growing integration of multi-scale data, from molecular databases to clinical datasets, in conjunction with a democratization of DL models to leverage these different data types. Neural nets have been used so far mostly for NLP applications in PV, but they have integrated the most recent state-of-the-art concepts such as attention mechanisms and multi tasks learning. Their applications are starting to be used beyond that scope, both in chemoinformatics and with clinical observational data. It noted that most of the approaches in the recent years that aim at predicting ADEs have been using annotated datasets. This almost exclusive use of supervised models has its limits, as the prediction of novel and unknown drug effects cannot rely on labeled data. This is only the dawn of AI, and numerous questions remain such as how to address class imbalances in supervised modeling tasks, and how to incorporate unsupervised approaches in PV studies (Outstanding Questions). Techniques such as GANs hold promise in addressing some of these concerns. For example, novel unsupervised approaches using GANs that can generate in silco molecules with desired chemical properties are starting to emerge, showing great promise for drug safety. While academic research has witnessed a drastic increase in the use of ML and DL, the community will begin to see these approaches entering into practice at a growing rate. For example, the FDA recently released plans for a new regulatory framework to promote the development of safe medical devices using AI algorithms. We expect that this will extend to drug development and safety in the future. Appropriate regulatory frameworks will need to be established to control for the risk of false positives. Overall, the risk of implementing AI approaches for PV is low and the opportunity high as it may have a positive impact on healthcare.

X. LINK TO CODE AND EXECUTABLE FILE

- ✓ Link to GitHub
  https://github.com/DivyaSadanandan/Side-effect-of-Drug
- ✓ Link to Loom video
  https://www.loom.com/share/9ad0c92fc97149d684150b367e370659
- ✓ Link to the dataset webmd.csv:
  https://www.kaggle.com/rohanharode07/webmd-drug-reviews-dataset
- ✓ Link to the code and executable file (Google Colab)
  https://colab.research.google.com/drive/1UJbyuIPvq96h05Mydt3vTFlTgP1oi3Io?usp=sharing

XI. REFERENCES

- *Dataset*

UCI -
https://archive.ics.uci.edu/ml/datasets.php?format=&task=&att=&area=life&numAtt=&numIns=greater1000&type=&sort=nameUp&view=table
Kaggle-
https://www.kaggle.com/datasets?search=Drug&datasetsOnly=true
https://www.kaggle.com/rohanharode07/webmd-drug-reviews-dataset
https://g25.tcsion.com/LX/contents/content_home?content_player=true&LaunchFrom=iHUB&newWindow=Y&org_id=1016&User-Agent=Computer&mtop_sec_key=vLrO8%2FXQvzyzcoiAgKjerpSx9UbUuZo%2FQMO%2B9tknAczz3YFoDKoIWWTWWuepY0PI&c_id=rio-125-classification-model-build-a-model-that-cl-785-3298&eForm=Y&usrid=16729032&TargetOrgid=3298&usrorgid=1016&serviceid=13&app_id=9519
https://www.kaggle.com/sszokoly/webmd-part1

- *Accessing large files in colab*

https://www.youtube.com/watch?v=aTHtJhdrKKQ
https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

- *Data cleaning*

https://www.kdnuggets.com/2020/07/data-cleaning-secret-ingredient-success-data-science-project.html

- *Data Preprocessing-*

https://www.youtube.com/watch?v=NBm4etNMT5k
https://campus.datacamp.com/courses/dealing-with-missing-data-in-python/the-problem

- *ML approach –*

https://www.sciencedirect.com/science/article/abs/pii/S1476927116302195

- *Implementation-*

https://www.youtube.com/watch?v=gYsU2VTESE8

- *Encoding categorical data-*

https://medium.com/wicds/label-and-one-hot-encoding-61525a32b99c
https://www.sciencedirect.com/science/article/abs/pii/S1476927116302195
https://towardsdatascience.com/how-machine-learning-can-help-identify-effectiveness-and-adverseness-of-a-drug-e23c7933c233
https://www.kaggle.com/ayushggarg/covid19-vaccine-adverse-reactions
https://www.sciencedirect.com/science/article/abs/pii/S1476927116302195

- *Road map-NLP*

https://www.youtube.com/watch?v=fM4qTMfCoak&list=PLZoTAELRMXVMdJ5sqbCK2LiM0HhQVWNzm

- *Text mining & Sentiment Analysis*

https://www.youtube.com/watch?v=lMQzEk5vht4
https://www.youtube.com/watch?v=AzjIu4YjHCo&t=600s
https://www.youtube.com/watch?v=HsI5YF6uxGs&t=1s
http://github.com/manan904/Twitter-Sentiment-Analysis
http://github.com/marcosan93/Stock-Analyzer
http://github.com/samiramunir/Simple-Sentiment-Analysis-using-NLTK

- *Word cloud*

https://amueller.github.io/word_cloud/auto_examples/a_new_hope.html
https://www.datacamp.com/community/tutorials/wordcloud-python

- *Sentiment Analysis with VADER*

https://www.youtube.com/watch?v=Alu_cCXNS-k
https://www.youtube.com/watch?v=HsI5YF6uxGs&t=1s

- *Count vectorizer*

https://www.youtube.com/watch?v=lBO1L8pqR9s
https://www.youtube.com/watch?v=lBO1L8pqR9s&t=53s

- *TF IDF*

https://www.youtube.com/watch?v=D2V1okCEsiE

- *Bag of words*

https://www.youtube.com/watch?v=IKqBLTeQQL8&list=PLZoTAELRMXVMdJ5sqbCK2LiM0HhQVWNzm&index=7

- *Stemming & Lemmatization*

https://www.youtube.com/watch?v=JpxCt3kvbLk&list=PLZoTAELRMXVMdJ5sqbCK2LiM0HhQVWNzm&index=4

- *Training and testing data*

https://www.youtube.com/watch?v=fwY9Qv96DJY
https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623

- *Multi label classification-*

https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/
https://www.youtube.com/watch?v=HsI5YF6uxGs
https://www.youtube.com/watch?v=jEnVBHYF2bQ&t=1374s
https://www.kaggle.com/sszokoly/webmd-part1

- *Classification algorithm*

https://www.youtube.com/watch?v=ppXFoltcX7A

- *More references*

https://towardsdatascience.com/how-machine-learning-can-help-identify-effectiveness-and-adverseness-of-a-drug-e23c7933c233
https://www.sciencedirect.com/science/article/abs/pii/S1476927116302195