DRUG REVIEW ANALYSIS REPORT

PG-DBDA

GROUP NO-21

IACSD, Pune

INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT AKURDI, PUNE

Documentation On

"Drug Review Analysis"

PG-DBDA AUG 2020

Group No: 21

Submitted By:

Shweta Hepat: 1348

Riya Raghuwanshi: 1340

Mr. Prashant Karhale
Centre Coordinator

Mr. Akshay Tilekar
Project Guide

INDEX

Table of Contents

Chapter1

SYNOPSIS

1.1 Project Title

Drug Review analysis.

1.2 Project Option

Internal Project

1.3 Internal Guide

Mr. AKASHY TILEKAR

1.4 Sponsorship and External Guide

NA

1.5 Technical Keywords

Keywords: Machine Learning, Python**.**

1.6 Problem Statement

Performing sentiment analysis so people get easily choose preferable drug.

1.7 Abstract

The dataset provides patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting overall patient satisfaction. Due to the huge amount of biological and medical data available today, along with well-established machine learning algorithms, the design of largely automated drug development pipelines can now be envisioned. These pipelines may guide, or speed up, drug discovery; provide a better understanding of diseases and associated biological phenomena; help planning preclinical wet-lab experiments, and even future clinical trials. This automation of the drug development process might be key to the current issue of low productivity rate that pharmaceutical companies currently face. In this survey, we will particularly focus on two classes of methods: sequential learning and recommend systems, which are active biomedical fields of research.

1.8 Goals and Objectives

Our objective was to analyze the drugs, based on the reviews.
So, that we could help patients make a more suitable decision making it easier for them in the future.

Chapter2

INTRODUCTION

2.1 Purpose

- People post rating about drugs on based on their experience
- People also post 300-character review about drugs on portal
- Portal show top rating of drugs
- It attracts more people to visit the portal

2.2 Scope of The Project:

2.2.1 Initial Functional Requirement Will Be: -

• Selecting the algorithm meeting requirement.

• Choosing the optimum algorithm form set of algorithm.

• Testing it on the datasets.

• After getting the result if the result is low change the hyper parameters.

• Out of all result get best of all.

2.2.2 Initial non-functional requirement will be: -

• Getting the large datasets can provide developer enough data to train the model.

• Maintain the minimum variance bias so the model successfully works.

• Avoid the under-fitting and over-fitting conditions.

Terms	Definitions
Dataset	Data for training and testing for the model.
Variance	Difference between the training and testing accuracy.
Bias	Both learning and training accuracy is

	low.
Over fitting	Model is very complex
Under fitting	Model is bias
Developer	Who is developing the model
Review	A written recommendation about the appropriateness of an Product/Drugs for selling and buying may include suggestions for improvement.
Reviewer	A person that examines an Product/Drugs and has the ability to recommend approved Product/Drugs for buying or to request that changes be made in the Product/Drugs.
Software Requirement Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document
User	Reviewer

2.3 Software Life Cycle Model

In order to make this Project we are going to use Classic LIFE CYCLE MODEL. Classic life cycle model is also known as WATERFALL MODEL. The life cycle model demands a Systematic sequential approach to software development that begins at the system level and progress through analysis design coding, testing and maintenance.

2.3.1 The Classic Life Cycle Model

The waterfall model is sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception initiation, Analysis, Design (validation), construction. Testing and maintained.

1. System Engineering and Analysis:

Because software is always a part of larger system work. Begins by establishing requirement for all system elements and Then allocating some subset of these requirement to the software system Engineering and analysis encompasses the

requirement gathering at the system level with a small amount of top level design and analysis.

2. Software requirement Analysis:

The requirement gathering process is intensified and focused specifically on the software Requirement for the both system and software are discussed and reviewed with the customer. The customer specifies the entire requirement or the software and the function to be performed by the software.

3. Design:

Software design is actually a multi-step process that focuses on distinct attributes of the program data structure, software architecture, procedural detail and interface of the software that can be assessed or quality before coding begins. Like requirement the design is documented and becomes part of the software.

4. Coding:

The design must be translated into a machine readable form. The coding step performs this task. If design is programmed in a detailed manner, coding can be accomplished mechanically.

5. Testing:

Once code has been generated programmed testing begins. The testing process focuses on the internals of the software ensuring that all statement have been tested and on the functional externals hat is conducting tests to uncover the errors and ensure that defined input will produce the results that agree with the required results.

- **Unit testing:**

In computer programming, Unit testing is software Verification and validation method where the programmer gains confidence that individual units of source code are fit to use a unit is the smallest testable part of an application. In procedural programming a unit may be an individual programmed, function, procedure, etc. while in object-oriented programming, the smallest Unit is a class, which may belong to a base/super class abstract class or derived/child class.

- **Benefits:**

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict written contract that the piece of code must satisfy.

- **Documentation:**

Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit and how to use it can look at the tests to gain a basic understanding of the unit API

- **Limitation of unit testing:**

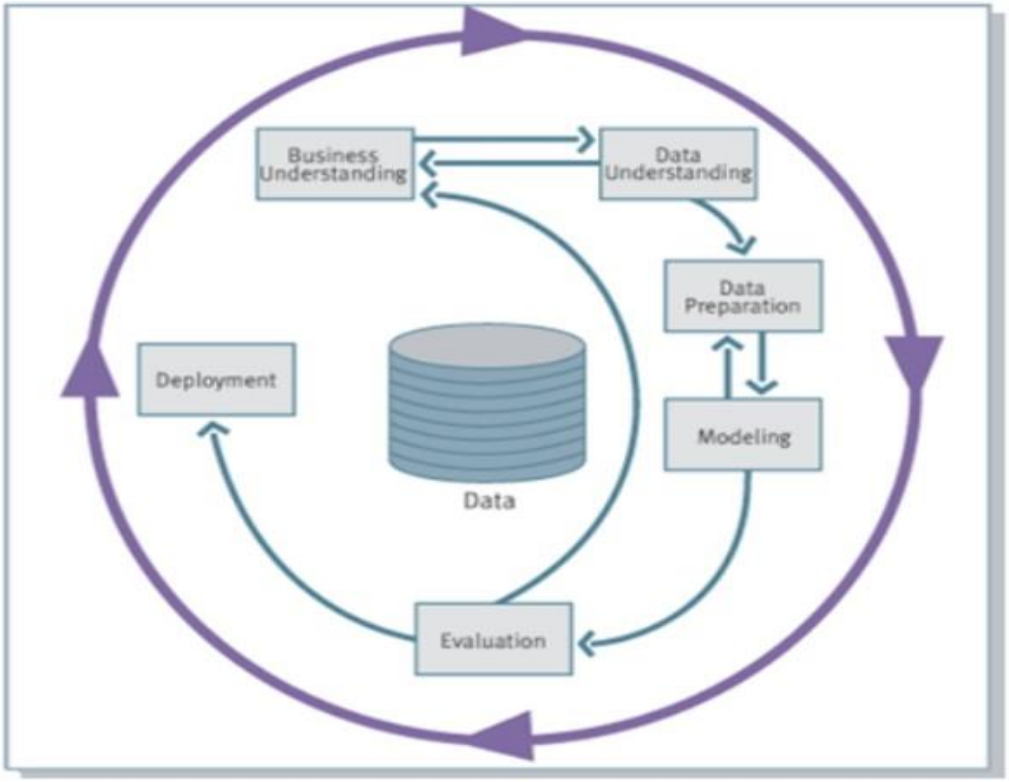
Testing cannot be expected to catch error in the program. It is impossible to evaluate all execution paths for all but the most trivial programs. The same is true for unit testing. Additionally, by unit testing only types the functionality if the units themselves.

6. Maintenance:

Software will undoubtedly undergo change after it is Delivered to the customer Change will occur because errors have been encountered because the software must be able adopted to accommodate changes in its external environment because the customer requires functional or performance enhancement enhancements. The classic life cycle is the oldest and most widely used paradigm or software engineering.

2.4 CRISP-DM Method Life Cycle

CRISP-DM methodology that stands for Cross Industry Standard Process for Data Mining, is a cycle that describes commonly used approaches that data mining experts use to tackle problems in traditional BI data mining. It is still being used in traditional BI data mining teams. Take a look at the following illustration. It shows the major stages of the cycle as described by the CRISP-DM methodology and how they are interrelated.



Let us now learn a little more on each of the stages involved in the CRISP-DM life cycle-

• Business Understanding:

This initial phase focuses on understanding the project objectives and requirements from a business perspective, and then converting this knowledge into a data mining problem definition. A preliminary plan is designed to achieve the objectives. A decision model, especially one built using the Decision Model and Notation standard can be used.

• Data Understanding:

The data understanding phase starts with an initial data collection and proceeds with activities in order to get familiar with the data, to identify data quality problems, to discover first insights into the data, or to detect interesting subsets to form hypotheses for hidden information.

• Data Preparation:

The data preparation phase covers all activities to construct the final dataset (data that will be fed into the modelling tool(s)) from the initial raw data. Data preparation tasks are likely to be performed multiple times, and not in any prescribed order. Tasks include table, record, and attribute selection as well as transformation and cleaning of data for modelling tools.

• Modelling

In this phase, various modelling techniques are selected and applied and their parameters are calibrated to optimal values. Typically, there are several techniques for the same data mining problem type. Some techniques have specific requirements on the form of data. Therefore, it is often required to step back to the data preparation phase.

• Evaluation

At this stage in the project, you have built a model (or Models) that appears to have high quality, from a data analysis perspective. Before proceeding to final deployment of the model, it is important to evaluate the model thoroughly and review the steps executed to construct the model, to be certain it properly achieves the business objectives. A key objective is to determine if there is some important business issue that has not been sufficiently considered. At the end of this phase, a decision on the use of the data mining results should be reached.

• Deployment

Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that is useful to the customer. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data scoring (e.g. segment allocation) or data mining process.

2.5 Overview of Document

• **Extracting Data:** Dataset is gathered from UCI Machine Learning website.

• **Objective:** Our objective was to analyses the drugs, based on the reviews. So, that we could help patients make a more suitable decision making it easier for them in the future.

• **Imports:** This section describes way to connect external analysis requirements to python.

• **Cleaning and prep:** Section describes procedure to cleaning the data to make it analysis ready.

• **Analysis:** This is most important section of project describing all analysis done and its visualization.

• **Conclusion:** Conclusion concludes the project.

Chapter3

OVERALL DESCRIPTION

3.1 Data

The dataset provides patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting overall patient satisfaction. The data was obtained by crawling online pharmaceutical review sites. The intention was to study

The data is split into a train (75%) a test (25%) partition (see publication) and stored in two. tsv (tab-separated-values) files, respectively.

Attribute Information:

1. drug Name (categorical): name of drug

2. condition (categorical): name of condition

3. review (text): patient review

4. rating (numerical): 10-star patient rating

5. date (date): date of review entry

6. useful Count (numerical): number of users who found review useful

We found an interesting dataset at UCI Machine Learning Repository, called *"Drug Review Dataset (Drugs.com) Dataset"* [7]. The dataset was obtained by crawling online pharmaceutical review sites, *Drugs.com*. The dataset contains 215063 number of samples. As shown in Table 1, the dataset
provides patient ratings from one to ten showing overall patient satisfaction, patient reviews on specific drugs along with related conditions and the date of recording, and finally
the number of other users who found that such review is useful

```
   uniqueid                          drugName                     condition  \
0    206461                         Valsartan  Left Ventricular Dysfunction
1     95260                        Guanfacine                          ADHD
2     92703                            Lybrel               Birth Control
3    138000                        Ortho Evra               Birth Control
4     35696       Buprenorphine / naloxone           Opiate Dependence
5    155963                            Cialis  Benign Prostatic Hyperplasia
6    165907                    Levonorgestrel       Emergency Contraception
7    102654                      Aripiprazole             Bipolar Disorde
8     74811                           Keppra                    Epilepsy
9     48928  Ethinyl estradiol / levonorgestrel               Birth Control

                                              review  rating  \
0  "It has no side effect, I take it in combinati...     9.0
1  "My son is halfway through his fourth week of ...     8.0
2  "I used to take another oral contraceptive, wh...     5.0
3  "This is my first time using any form of birth...     8.0
4  "Suboxone has completely turned my life around...     9.0
5  "2nd day on 5mg started to work with rock hard...     2.0
6  "He pulled out, but he cummed a bit in me. I t...     1.0
7  "Abilify changed my life. There is hope. I was...    10.0
8  " I Ve had  nothing but problems with the Kepp...     1.0
9  "I had been on the pill for many years. When m...     8.0

                date  usefulCount
0       May 20, 2012           27
1      April 27, 2010          192
2   December 14, 2009           17
3    November 3, 2015           10
4   November 27, 2016           37
5   November 28, 2015           43
6      March 7, 2017            5
7     March 14, 2015           32
8     August 9, 2016           11
9    December 8, 2016            1
Shape of train : (161297, 7)
Shape of test : (53766, 7)
```

3.2 Imports

• **Matplotlib:** Matplotlib is a collection of command style functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure. We have used it to show visualizations of analysis.

• **Pandas:** Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. We have used pandas for perform operation on data.

• **Numpy:** Numpy is used to for mathematical operations. This package provides easy use of mathematical function.

• **Seaborn:** Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

• **Sklearn:** Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines.

Chapter4

REQUIREMENT SPECIFICATION

4.1 External Requirement Specification

4.1.1 Hardware Interfaces

• Processor: i3 Processor or above
• RAM: 4 GB (min)
• Hard Disk: 20 GB(free)

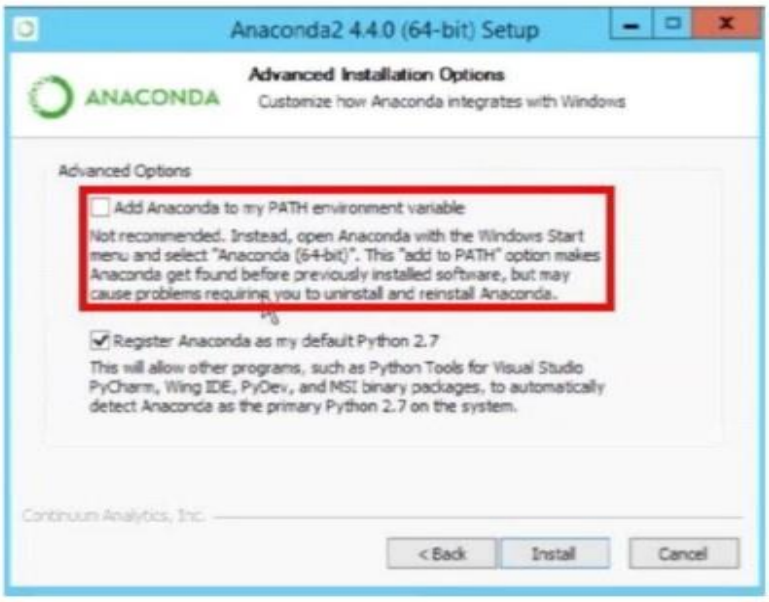
4.1.2 Software Interfaces

Functional:

First of all, model should be successfully trained by developer. Installing Anaconda on Windows

1. Download and install Anaconda (windows version).

2. Select the default options when prompted during the installation of Anaconda.



3. After you finished installing, open Anaconda Prompt. Type the command below to see that you can use a Jupyter (IPython) Notebook.

4. If you didn't check the add Anaconda to path argument during the installation process, you will have to add python and conda to your environment variables.

5. This step gives two options for adding python and conda to your path.

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda R distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS and Linux. To get Navigator, get the Navigator cheat sheet and install Anaconda. The Navigator Getting started with Navigator section shows how to start Navigator from the shortcuts or from a terminal window. Preferable in case data is small or can be performable in Anaconda.

Google Colab

In case of very large data where Anaconda takes time use Google Colab for better performance. Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs. Colab is free to use.

Spyder:

It is an open-source cross-platform integrated development environment(IDE) for scientific programming in the Python language Spyder integrates with a number of prominent packages in the scientific Pythonstack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open-source software. It is released under the MIT license**.**

Chapter5

DATA PRE-PROCESSING AND ANALYSIS

5.1 Data Cleaning

Data cleaning is the process of preparing data for analysis by removing or changing data that is wrong, incomplete, unrelated/unimportant, copied, or improperly formatted. This data is usually not necessary or helpful when it comes to carefully studying data because it may interfere with the process or provide incorrect results. There are (more than two, but not a lot of) methods for cleaning data depending on how it is stored along with the answers being searched for/tried to get.

5.1.1 Screenshot before cleaning

In this figure, the data has NaN values and also data gets over fitted and because of that the accuracy getting is wrong for model.

```
Name: drugName, Length: 884, dtype: int64
condition
Not Listed / Othe                          214
Pain                                       200
Birth Control                              172
High Blood Pressure                        140
Acne                                       117
Depression                                 105
Rheumatoid Arthritis                        98
Diabetes, Type 2                            89
Allergic Rhinitis                           88
Bipolar Disorde                             80
Osteoarthritis                              80
Insomnia                                    78
Anxiety                                     78
Abnormal Uterine Bleeding                   74
Migraine                                    59
Psoriasis                                   58
Endometriosis                               57
3</span> users found this comment helpful.  57
ADHD                                        55
Asthma, Maintenance                         54
Chronic Pain                                53
Migraine Prevention                         50
Irritable Bowel Syndrome                    49
Major Depressive Disorde                    49
Urinary Tract Infection                     47
4</span> users found this comment helpful.  45
ibromyalgia                                 45
Bronchitis                                  44
Postmenopausal Symptoms                     44
2</span> users found this comment helpful.  43
Name: drugName, dtype: int64
```

5.1.2 Screenshot after cleaning

In this figure, the data has NaN values gets removed and also data gets under sampled and because of that the accuracy getting is more better than over fitted model.

```
Name: drugName, Length: 810, dtype: int64
condition
Pain                         200
Birth Control                172
High Blood Pressure          140
Acne                         117
Depression                   105
Rheumatoid Arthritis          98
Diabetes, Type 2              89
Allergic Rhinitis             88
Bipolar Disorde               80
Osteoarthritis                80
Anxiety                       78
Insomnia                      78
Abnormal Uterine Bleeding     74
Migraine                      59
Psoriasis                     58
Endometriosis                 57
ADHD                          55
Asthma, Maintenance           54
Chronic Pain                  53
Migraine Prevention           50
Irritable Bowel Syndrome      49
Major Depressive Disorde      49
Urinary Tract Infection       47
ibromyalgia                   45
Postmenopausal Symptoms       44
Bronchitis                    44
GERD                          43
HIV Infection                 43
Bacterial Infection           43
Sinusitis                     42
Name: drugName, dtype: int64
```

5.2 Assessing columns

i) Assessing columns

- Visualizing missing counts per columns

ii) Dropping Unwanted columns

- *Unique ID*
- *Date*

• The URL

https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29

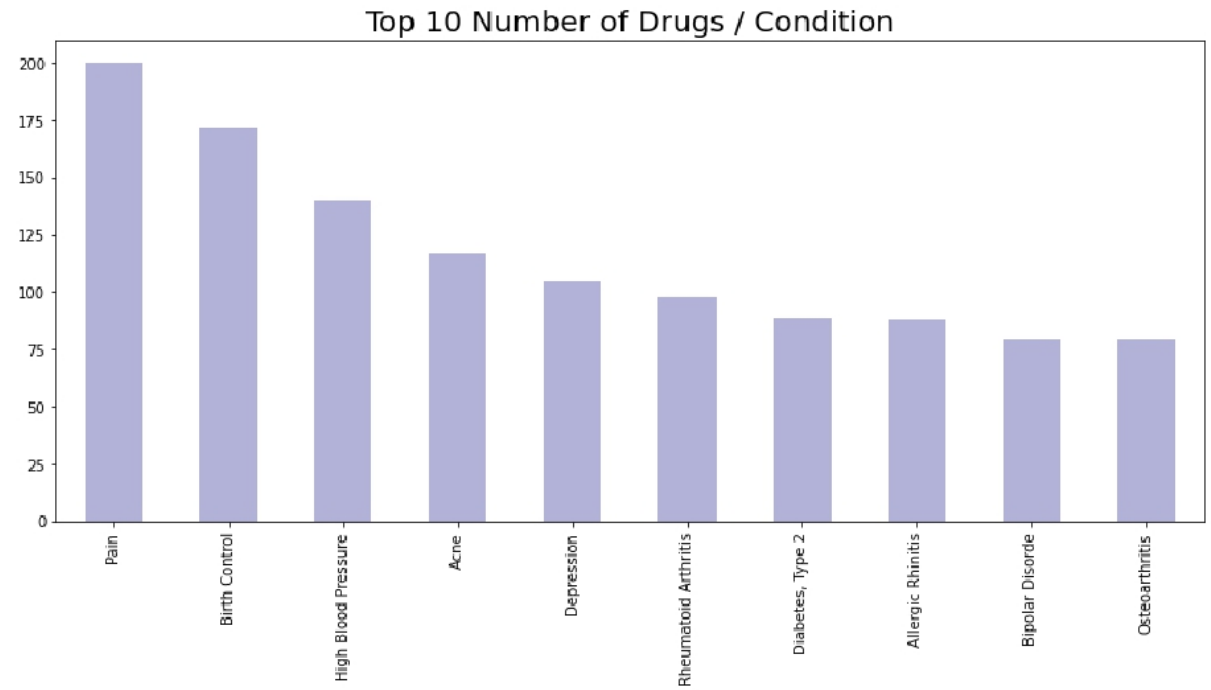
Dropping missing rows

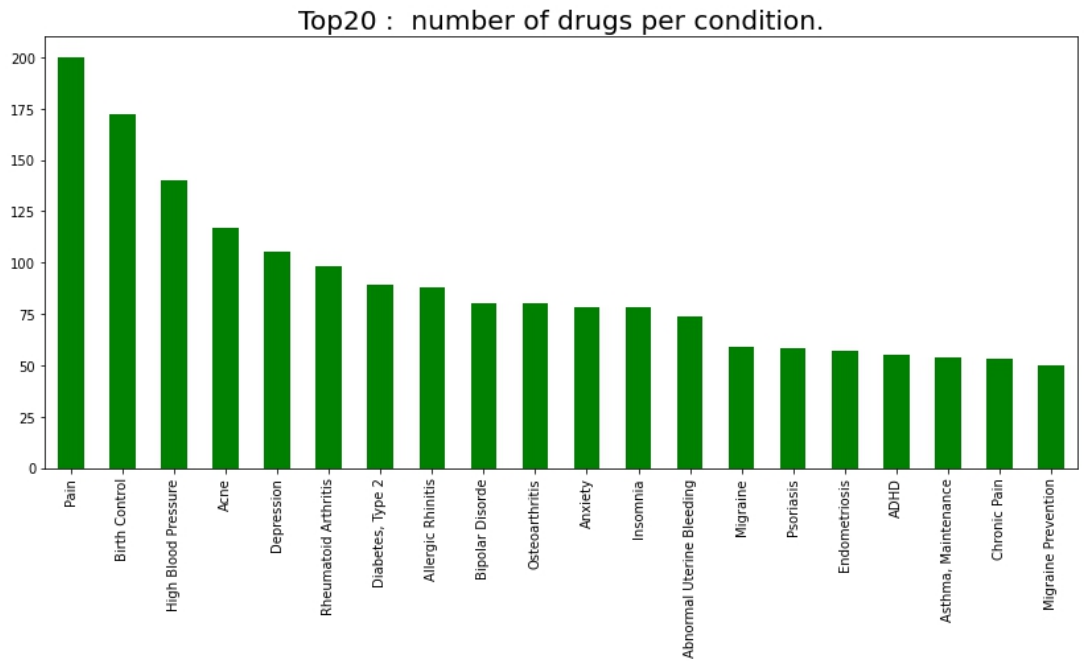
Replacing missing condition with particular condition of drugs indicating it

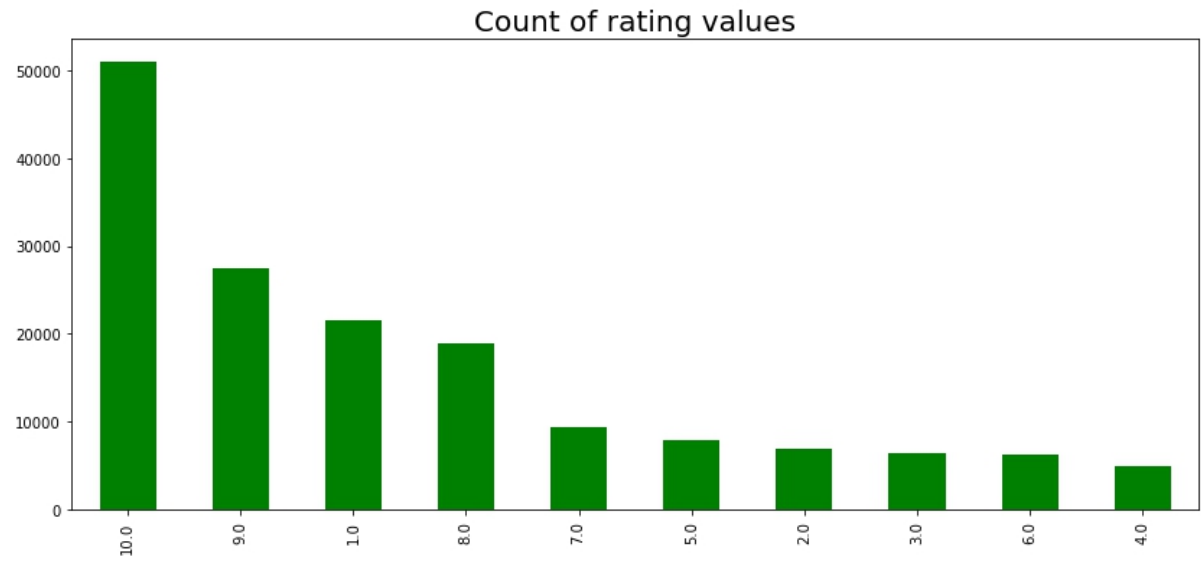
- Pre- processing columns

Exploratory Data Analysis

First of all, we would like to understand how is the distribution of the user ratings. As shown in Figure 1, rating ten is the most frequent ratings that user gave, which has 68005. samples. Then, the followings are the rating nine and rating one, which has 36708 and 28918 samples, respectively. That is to say, the total number of samples with rating one, nine, or ten is 133631, which is over half of the number of the overall samples. Hence, we can conclude that the drug users like to give the extreme ratings. Also, if users are not satisfied, they would tend to give the lowest rating, one star,
but not two or three stars. Besides, we would like to see if different dates effect the user ratings. Note that the overall user ratings' mean is 6.99 and the standard deviation is 3.28


Top 10 Number of Drugs / Condition
200
175
150
125
100
75
50
25
0
Pain
Birth Control
High Blood Pressure
Acne
Depression
Rheumatoid Arthritis
Diabetes, Type 2
Allergic Rhinitis
Bipolar Disorde
Osteoarthritis


Top20 : number of drugs per condition.
200
175
150
125
100
75
50
25
0
Pain
Birth Control
High Blood Pressure
Acne
Depression
Rheumatoid Arthritis
Diabetes, Type 2
Allergic Rhinitis
Bipolar Disorde
Osteoarthritis
Anxiety
Insomnia
Abnormal Uterine Bleeding
Migraine
Psoriasis
Endometriosis
ADHD
Asthma, Maintenance
Chronic Pain
Migraine Prevention



NLP (Natural Language Processing):

Sentiment Analysis is the process of determining whether a piece of writing is positive, negative or neutral, and then assigning a weighted sentiment score to each entity, theme, topic, and category within the document. This is an incredibly complex task that varies wildly with context. For example, take the phrase, “sick burn” In the context of video games, this might actually be a positive statement. Creating a set of NLP rules to account for every possible sentiment score for every possible word in every possible context would be impossible. But by training a machine learning model on pre-scored data, it can learn to understand what “sick burn” means in the context of video gaming, versus in the context of healthcare. Unsurprisingly, each language requires its own sentiment classification model.

Stop Words: A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

We would not want these words to take up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to stop words. NLTK (Natural Language Toolkit) in python has a list of stop words stored in 16 different languages. You can find them in the nltk_data directory. home/pratima/nltk_data/corpora/stop words is the directory address. (Do not forget to change your home directory name)

```
alphanumeric=lambda x: re.sub('[^a-zA-Z]', ' ', str(x) )#selecting only a-zA-Z

punc_lower=lambda x: re.sub('[%s]' % re.escape(string.punctuation), ' ', x.lower()) # selecting puctuation marks and lower case
split=lambda x: x.split()

df_train1['review'] = df_train1.review.map(alphanumeric).map(punc_lower).map(split)
print(df_train1)
```

Stemming:

Stemming and Lemmatization both generate the root form of the inflected words. The difference is that stem might not be an actual word whereas, lemma is an actual language word. Stemming follows an algorithm with steps to perform on the words which makes it faster. Whereas, in lemmatization, you used WordNet corpus and a corpus for stop words as well to produce lemma which makes it slower than stemming. You also had to define a parts-of-speech to obtain the correct lemma.

Lemmatizing:

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma*. If confronted with the token *saw*, stemming might return just *s*, whereas lemmatization would attempt to return either *see* or *saw* depending on whether the use of the token was as a verb or a noun. The two may also differ in that stemming most commonly collapses derivationally related words, whereas lemmatization commonly only collapses the different inflectional forms of a lemma. Linguistic processing for stemming or lemmatization is often done by an additional plug-in component to the indexing process, and a number of such components exist, both commercial and open-source.

review	rating	usefulCount
[my, son, is, halfway, through, his, fourth, w...	8.0	192
[i, used, to, take, another, oral, contracepti...	5.0	17
[this, is, my, first, time, using, any, form, ...	8.0	10
[nd, day, on, mg, started, to, work, with, roc...	2.0	43
[he, pulled, out, but, he, cummed, a, bit, in,...	1.0	5
...	...	...
[this, would, be, my, second, month, on, junel...	6.0	0
[i, was, given, this, in, iv, before, surgey, ...	1.0	34
[limited, improvement, after, months, develope...	2.0	35
[i, ve, been, on, thyroid, medication, years, ...	10.0	79
[i, ve, had, chronic, constipation, all, my, a...	9.0	116

CountVectorezier: Scikit-learn's CountVectorizer is used to convert a collection of text documents to a vector of term/token counts. It also enables the pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text.

TF-IDF: TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction. Let's take sample example and explore two different spicy sparse matrices before go into deep explanation

TOPIC MODLING: Topic modeling is a method for *unsupervised* classification of documents, similar to clustering on numeric data, which finds some natural groups of items (topics) even when we're not sure what we're looking for. A document can be a part of multiple topics, kind of like in fuzzy clustering (soft clustering) in which each data point belongs to more than one cluster.

Topic modeling provides methods for automatically organizing, understanding, searching, and summarizing large electronic archives.
It can help with the following:

- discovering the hidden themes in the collection.
- classifying the documents into the discovered themes.

- using the classification to organize/summarize/search the documents.

Latent Dirichlet Allocation: It is one of the most popular topic modeling methods. Each document is made up of various words, and each topic also has various words belonging to it. The aim of LDA is to find topics a document belongs to, based on the words in it in LDA we created dominant topics.

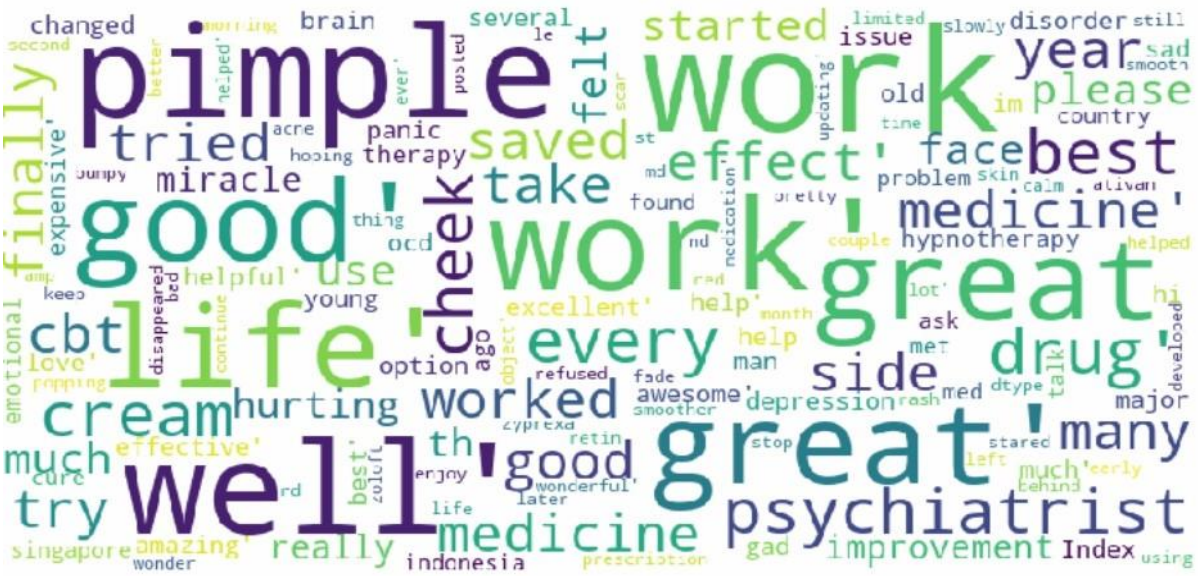
	drugName	condition	rating	usefulCount	Dominant_Topic	Perc_Contribution	Topic_Keywords
0	Guanfacine	ADHD	8.0	192	2.0	0.6443	side effect, no side, year ago, panic attack
1	Lybrel	Birth Control	5.0	17	3.0	0.6844	birth control, mood swing, weight gain, side e...
2	Ortho Evra	Birth Control	8.0	10	3.0	0.8630	birth control, mood swing, weight gain, side e...
3	Cialis	Benign Prostatic Hyperplasia	2.0	43	0.0	0.5926	blood pressure, started taking, lost lb, feel ...
4	Levonorgestrel	Emergency Contraception	1.0	5	1.0	0.9370	felt like, first time, yeast infection, took pill
...	...	...	...	...	...	...	...
148381	Tekturna	High Blood Pressure	7.0	18	3.0	0.9193	birth control, mood swing, weight gain, side e...
148382	Junel 1.5 / 30	Birth Control	6.0	0	0.0	0.6094	blood pressure, started taking, lost lb, feel ...
148383	Metoclopramide	Nausea/Vomiting	1.0	34	2.0	0.6066	side effect, no side, year ago, panic attack
148384	Orencia	Rheumatoid Arthritis	2.0	35	1.0	0.4715	felt like, first time, yeast infection, took pill
148385	Thyroid desiccated	Underactive Thyroid	10.0	79	0.0	0.6829	blood pressure, started taking, lost lb, feel ...

Top 10 Condition with Dominant Topic:

	drugName	condition	rating	usefulCount	Dominant_Topic	Perc_Contribution	Topic_Keywords
1	Lybrel	Birth Control	5.0	17	3.0	0.6844	birth control, mood swing, weight gain, side e...
2	Ortho Evra	Birth Control	8.0	10	3.0	0.8630	birth control, mood swing, weight gain, side e...
5	Aripiprazole	Bipolar Disorde	10.0	32	0.0	0.5667	blood pressure, started taking, lost lb, feel ...
7	Ethinyl estradiol / levonorgestrel	Birth Control	8.0	1	3.0	0.5531	birth control, mood swing, weight gain, side e...
9	L-methylfolate	Depression	10.0	54	2.0	0.8252	side effect, no side, year ago, panic attack

WORD CLOUD: -

We created word cloud on top words in the review:



TOP SELECTED WORDS FROM REVIEW

ALGORITHM SELECTION:

Logistic Regression:

Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

For example,

- To predict whether an email is spam (1) or (0)
- Whether the tumour is malignant (1) or not (0)

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time. From this example, it

can be inferred that linear regression is not suitable for classification problem. Linear regression is unbounded, and this brings logistic regression into picture. Their value strictly ranges from 0 to 1.

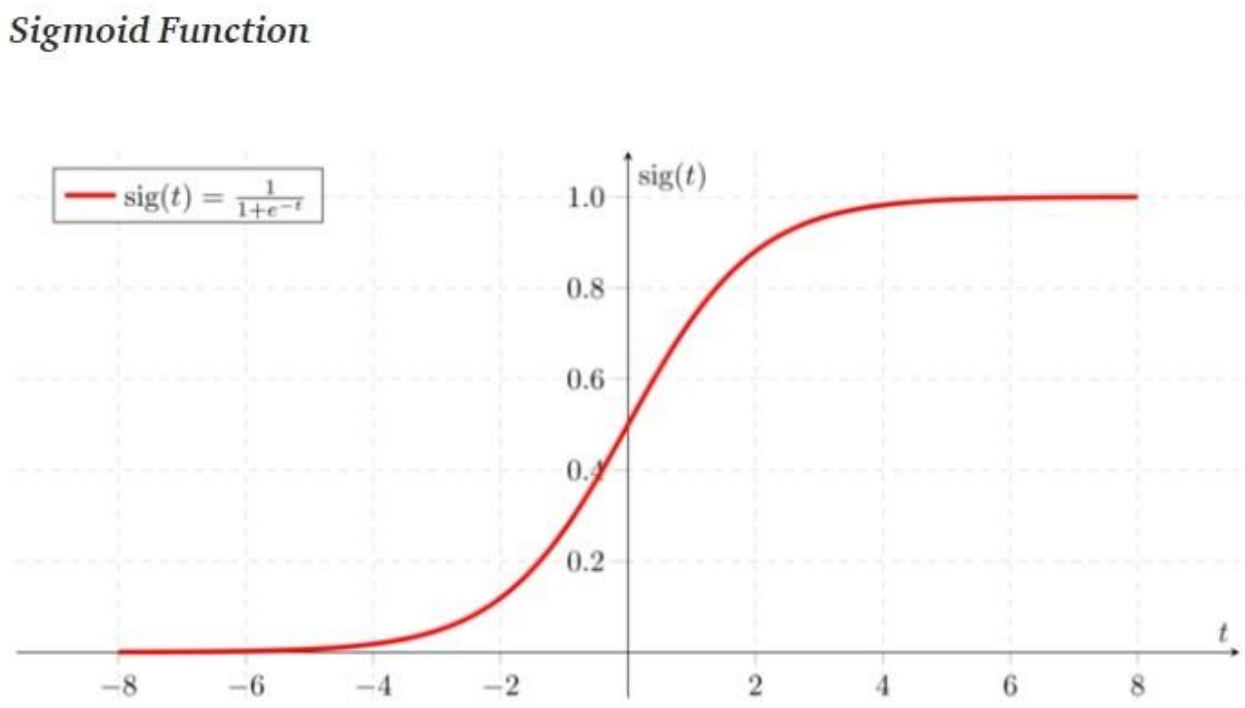


Figure 2: Sigmoid Activation Function

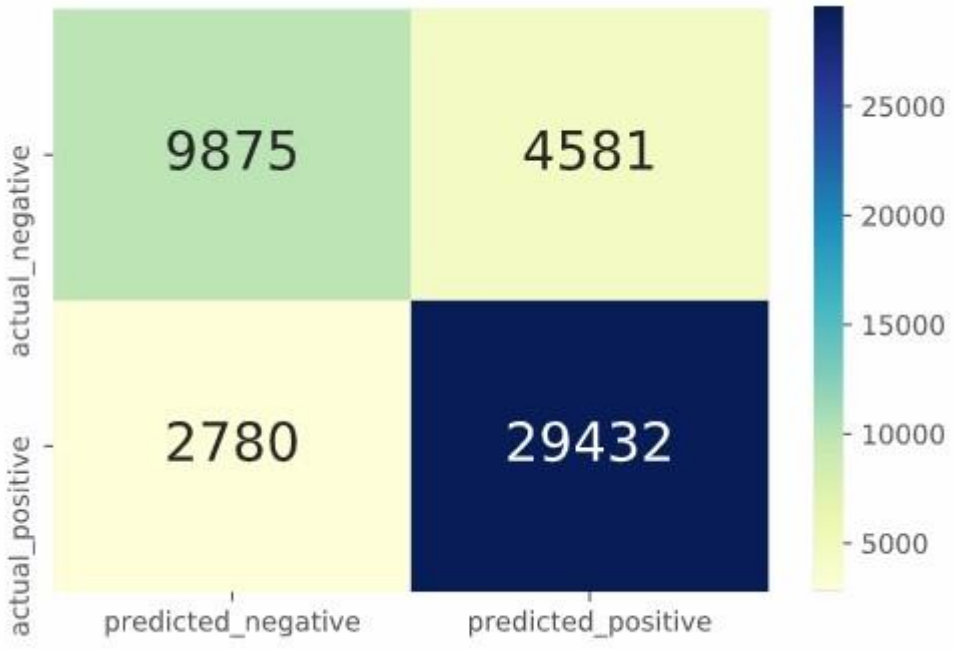
We Created Corpus Matrix of unigram and bigram to fit our model:

	aa	aarp	ab	abacavir	abate	abated	abcess	abd	abdomen	abdominal	...	zovirax	zpack	zpak	zumba	zutripro	zyban	zyclara	zyprexa	zyrtec	zyvox
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

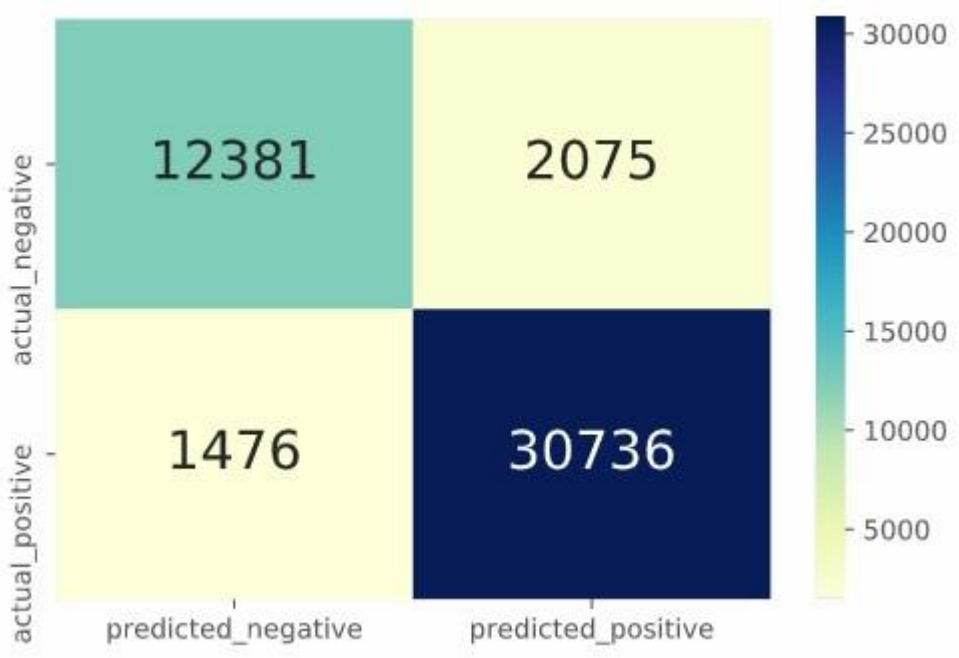
5 rows × 10700 columns

abdomen area	...	zutripro	zyban	zyban quit	zyclara	zyprexa	zyprexa made	zyprexa year	zyrtec	zyrtec claritin	zyv
0	...	0	0	0	0	0	0	0	0	0	0
0	...	0	0	0	0	0	0	0	0	0	0
0	...	0	0	0	0	0	0	0	0	0	0
0	...	0	0	0	0	0	0	0	0	0	0
0	...	0	0	0	0	0	0	0	0	0	0

Accuracy of Logistic Regression for unigram:



Accuracy of Logistic Regression for bigram:



	LogReg1	LogReg2
Accuracy	0.842	0.924
Precision	0.865	0.937
Recall	0.914	0.954
F1 Score	0.889	0.945

Naive Bayes:

Naive Bayes is a supervised learning algorithm used for classification tasks. Hence, it is also called Naive Bayes Classifier. As other supervised learning algorithms, naive Bayes uses features to make a prediction on a target variable. The key difference is that naive Bayes assumes that features are independent of each other and there is no correlation between features. However, this is not the case in real life. This naive assumption of features being uncorrelated is the reason why this algorithm is called "naive". In this post, I will first cover some basic concepts on probability and show how Bayes' Theorem, the core of naive Bayes classifier, is derived. Then I will show how naive Bayes classifier builds up on Bayes' Theorem as well as advantages/disadvantages of naive Bayes and its implementation on scikit-learn. Naive Bayes is a supervised learning algorithm for classification so the task is to find the class of observation (data point) given the values of features. Naive Bayes classifier

calculates the probability of a class given a set of feature values (i.e. p (yi | x1, x2 , … , xn)). Input this into Bayes' theorem:

$$p(yi \mid x1, x2, \ldots, xn) = \frac{p(x1, x2, \ldots, xn \mid yi).p(yi)}{p(x1, x2, \ldots, xn)}$$

Again, scikit learn (python library) will help here to build a Naive Bayes model in Python. There are three types of Naive Bayes model under the scikit-learn library:

Gaussian: It is used in classification and it assumes that features follow a normal distribution.

Multinomial: It is used for discrete counts. For example, let's say, we have a text classification problem. Here we can consider Bernoulli trials which is one step further and instead of "word occurring in the document", we have "count how often word occurs in the document", you can think of it as "number of times outcome number x is observed over the n trials".

Bernoulli: The binomial model is useful if your feature vectors are binary (i.e. zeros and ones). One application would be text classification with 'bag of words' model where the 1s & 0s are "word occurs in the document" and "word does not occur in the document" respectively.

Accuracy Score using Count Vectorizer: -

	LogReg1	LogReg2	NB1(multinomial)	NB2(Bernoulli)
Accuracy	0.842	0.919	0.797	0.846
Precision	0.864	0.932	0.857	0.895
Recall	0.914	0.952	0.848	0.880
F1 Score	0.888	0.942	0.852	0.887

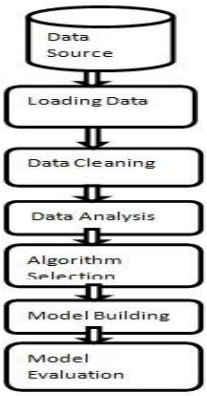
Accuracy Score using TFIDF:

	LR1-TFIDF	LR2-TFIDF	NB1-TFIDF	NB2-TFIDF
Accuracy	0.845	0.877	0.792	0.846
Precision	0.862	0.884	0.782	0.895
Recall	0.924	0.946	0.968	0.880
F1 Score	0.892	0.914	0.865	0.887

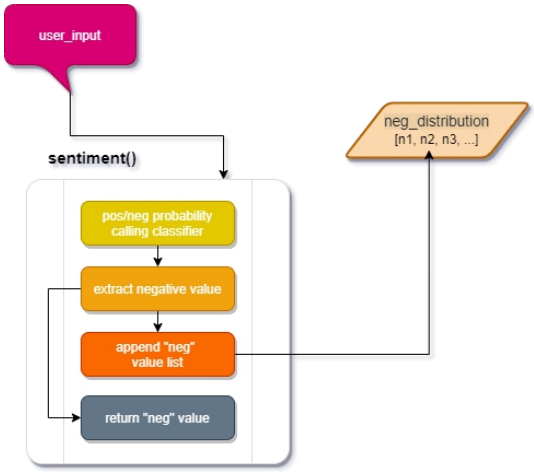
Chapter6

SYSTEM DESIGN

6.1 Flowchart of the System:



The above flowchart describes the working flow of the project. We are collecting our dataset from kaggle then load our data in jupyter notebook. After loading data, we had done EDA of data (Data cleaning, Data Analysis)



Chapter7

MODEL EVALUATION

Summary

Most of works in the same dataset and similar dataset focus on sentiment analysis [7][2][4][6] [11]. The works [7]and [11] employed sentiment analysis to predict users' satisfactory, while our work would like to predict exact rating based on reviews and other features such as useful count on the reviews. [11] differs from ours in a way that they broke sentences in a review and applied sentiment method to detect sentiment related with the sentences in a review, while we use the whole review text. The work [6] gave a clear sentiment classification word cloud. It seems that their method can preserve more positive and negative words. But since our work and [6] use different dataset, it cannot be a fair comparison. The state-of-the-art methods currently employed to study this type of data would be deep learning like [9]. A deep learning technique learn categories incrementally through its hidden layer architecture, defining low-level categories like letters first then little higher level categories like words and then higher level categories like sentences. Because deep learning is beyond the course's scope, we did not try this method. From the Section 1, we find that TF-IDF and useful count have a great influence on rating, but none of existing works used TF-IDF and useful count to predict users' satisfactory. On the other hand, we find that using mixed n-gram method can improve result a lot, which is the same as the work's [7] finding. We believe that the original work [7] simplified the problem a lot, since they divided all rating into three categories without scientific reason or proof. That's why they can achieve 91.89% accuracy.

Chapter8

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 pre-processing 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 Logistic Regression and Naïve Bayes (MultinomialNB and BernoulliNB). The best performing model is the LOGISTIC Regression Model but it's accuracy and the classification report are comparable to the Naïve Bayes Classifier. The accuracies were 0.919 and 0.846 respectively. The features importance is also plotted for Logistic Regression and Naïve Bayes Classifier. The classification report is also there for deeper analysis of the model as only accuracies doesn't tell much about a classification model.

Chapter9

FUTURE SCOPE

- Build a Recommender System
    - Help patients Decision Making
    - Provide benchmark to drug provider

- Topic Modelling for reviews across different conditions

REFERENCES

[1] Aspect-Based Sentiment Analysis of Drug Reviews Applying CrossDomain 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-winter2018
[6] UCI Library Drug Review Dataset (Drug.com) https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Dr ugs.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