CS6029 SOCIAL NETWORK ANALYSIS
MINI PROJECT DOCUMENTATION

DISEASE DRUG RECOMMENDATION SYSTEM USING COLLABORATIVE FILTERING

Submitted by,

Anika Arivarashan S	2019103506
Rajeshwari N	2019103566
Ria Bas Len	2019103050

INDEX

ABSTRACT

Data Mining is a method that requires analyzing and exploring large blocks of data to extract meaningful trends and patterns. Data mining techniques can be applied to various fields including medical databases. There are thousands of people all over the world facing health and medical diagnosis problems. Hospital Information System (HIS) generate massive data but gaining useful knowledge from the diagnosis case data is a big challenge. Using the methodologies used in this project, patients can easily get information about the disease they are suffering from and the drug helpful for dealing with that disease by just entering the symptoms he/she is showing. In this project, we recommend drugs to the users based on the diseases and the reviews. The diseases are predicted using four different models. The reviews are analyzed using the Vader tool and NLP-based sentiment analysis. And finally, the drugs are recommended using probabilistic and weighted average approaches. The details of each model and approach used in this project are explained in detail. The experimental results from this paper can be further utilized for research purposes and for other various medical utilities. This model obtained an accuracy of 86% and these experimental results can be further utilized for research purposes and other medical utilities.

INTRODUCTION AND OBJECTIVE

Since coronavirus has shown up, the inaccessibility of legitimate clinical resources is at its peak, like the shortage of specialists and healthcare workers, lack of proper equipment and medicines etc. The entire medical fraternity is in distress, which results in numerous individual's demise. Due to unavailability, individuals started taking medication independently without appropriate consultation, making the health condition worse than usual. As of late, machine learning has been valuable in numerous applications, and there is an increase in innovative work for automation.

Every day a new study comes up with accompanying more drugs, tests, accessible for clinical staff every day. Accordingly, it turns out to be progressively challenging for doctors to choose which treatment or medications to give to a patient based on indications, past clinical history. With the exponential development of the web and the web-based business industry, item reviews have become an imperative and integral factor for acquiring items worldwide. Individuals worldwide become adjusted to analyze reviews and websites first before settling on a choice to buy a thing. While most of past exploration zeroed in on rating expectation and proposals on the E-Commerce field, the territory of medical care or clinical therapies has been infrequently taken care of. There has been an expansion in the number of individuals worried about their well-being and finding a diagnosis online. As demonstrated in a Pew American Research center survey directed in 2013, roughly 60% of grown-ups searched online for health-related subjects, and around 35% of users looked for diagnosing health conditions on the web. A medication recommender framework is truly vital with the goal that it can assist specialists and help patients to build their knowledge of drugs on specific health conditions. A recommender framework is a customary system that proposes an item to the user, dependent on their advantage and necessity. These frameworks employ the customers' surveys to break down their sentiment and suggest a recommendation for their exact need. In the drug recommender system, medicine is offered on a specific condition dependent on patient reviews using sentiment analysis and feature engineering. Sentiment analysis is a progression of strategies, methods, and tools for distinguishing and extracting emotional data, such as opinion and attitudes, from language.

Also one of the most commonly found concerns among patients when confronted with any medical condition is which physician to trust. It is a known fact that the health of an individual significantly affects his/her quality of life. A survey in 2013 by the Pew Internet and American Life Project found that 59% of adults have looked online for health topics and with 35% of respondents focusing on diagnosing a medical condition online. There are more people every day caring about the health and medical diagnosis problem but still many who lose their lives due to medical errors. According to the administration's report, more than 200 thousand people in China and over 100 thousand in the USA, die each year due to medication errors. More than 42% medication errors are caused by doctors because they write prescriptions based on their experience which is quite limited. Hence, finding appropriate physicians to diagnose and treat medical conditions is one of the most important decisions a patient must make.

Advancements in Data mining and Recommender Technologies allow us to explore possibilities of potential knowledge from diagnosis history records and reviews and ratings on drugs to help doctors prescribe the correct medication and decrease the medication errors effectively.

The objective of this project is to design and implement a universal Disease Prediction and Drug Recommendation System that applies various Data Mining technologies to the recommendation system. By combining information from different sources we are using various prediction algorithms along with NLP for sentiment analysis and recommendation.

SUMMARY OF RELATED WORKS

With a sharp increment in AI advancement, there has been an exertion in applying machine learning and deep learning strategies to recommender frameworks. These days, recommender frameworks are very regular in the travel industry, e-commerce, restaurant, and so forth. Unfortunately, there are a limited number of studies available in the field of drug proposal framework utilizing sentiment analysis on the grounds that the medication reviews are substantially more intricate to analyze as it incorporates clinical wordings like infection names, reactions, a synthetic names that used in the production of the drug.

Diego Galeano et al, proposed that drug side effects represent one of the leading causes of morbidity and mortality in health care. Many side effects are not detected until the drug hits the market. In this paper, They investigated the use of collaborative filtering models for predicting side effects of marketed drugs. Drugs might not be able to produce any possible side effect. For instance, blood system drugs causes few sensory and endocrine-related side effects, whereas anti-cancer drugs produce side effects in almost all human systems.

Satvik Garg proposed that during pandemic situation, the inaccessibility of legitimate clinical resources is at its peak, like the shortage of specialists and healthcare workers, lack of proper equipment and medicines etc. Due to this unavailability, individuals started taking medication independently without appropriate consultation, making the health condition worse than usual. This paper intends to present a drug recommender system that can drastically reduce specialists heap. They build a medicine recommendation system that uses patient reviews to predict the sentiment using various vectorization processes like Bow, TF-IDF, Word2Vec, and Manual Feature Analysis, which can help recommend the top drug for a given disease by different classification algorithms.

T. Venkat Narayana Rao et al, proposed that sentiment analysis is done using Lightgbm, It is a gradient-boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following

advantages: Faster training speed, higher efficiency, and Lower memory usage. The Lightgbm uses XGBoost as a baseline and outperforms it in training speed and the dataset sizes it can handle. It finds out the correlation among the variables and later calculates predicted sentiment using cleaned reviews and recommends all the medicines/drugs with their mean predicted value. the medicine is more accurate when the mean predicted value is more. so, doctors prescribe the best medicine by considering the highest mean value predicted.

Kamaraj, K.Gomathi et al, proposed that they have demonstrated results of Decision Tree and Naive Bayes models in predicting three diseases, Heart Disease,Diabetes and Breast Cancer.

M.A.Nishara Banuet al, proposed, they have predicted heart diseases. By applying K-Mean on the medical dataset, they have clustered the relevant data, upon which MAFIA(Maximal Frequent Itemset Algorithm) is applied to generate rules and identification of frequent pattern which is fed to the C4.5 (Decision Tree) model to classify patterns.

H Wang et al, proposed paper, determine novel drug indications and side effects in one integrated framework. This strategy provides a complementary method to medical genetics-based drug repositioning, which reduces the occurrence of false positives in medical genetics-based drug repositioning, resulting in a ranked list of new candidate indications and/or side effects with different confidence levels.

Xiaohong Jiang et al. examined three distinct algorithms, decision tree algorithm, support vector machine (SVM), and backpropagation neural network on treatment data. SVM was picked for the medication proposal module as it performed truly well in each of the three unique boundaries - model exactness, model proficiency, model versatility. Additionally, proposed the mistake check system to ensure analysis, precision and administration quality.

Yin Zhang et al, proposed a novel cloud-assisted drug recommendation (CADRE), which can recommend users with top-N related medicines according to symptoms. In CADRE, we first cluster the drugs into several groups according to the functional description information and design a basic personalized drug recommendation based on user collaborative filtering.

Jiugang Li et al, constructed a hashtag recommender framework that utilizes the skip-gram model and applied convolutional neural networks (CNN) to learn semantic sentence vectors. These vectors use the features to classify hashtags using LSTM RNN. Results depict that this model beats the conventional models like SVM, Standard RNN. This exploration depends on the fact that it was undergoing regular AI methods like SVM and collaborative filtering techniques; the semantic features get lost, which has a vital influence in getting a decent expectation.

Youjun Bao et al, proposed a design and implement a universal medicine recommender system framework that applies data mining technologies to the recommendation system. The medicine recommender system consists of a database system module, data preparation module, recommendation model module, model evaluation, and data visualization module.

A few limitations of the existing works were tuning some of the required parameters is very time-consuming and our results are very dependent on these parameters, Algorithms that used patient reviews to predict sentiment using various vectorization processes were not optimized leading to incorrect results and errors. There is still room for improvement in predicting and understanding drug side effects in the emerging field of system pharmacology.

ARCHITECTURE DIAGRAM AND DETAILED DESIGN

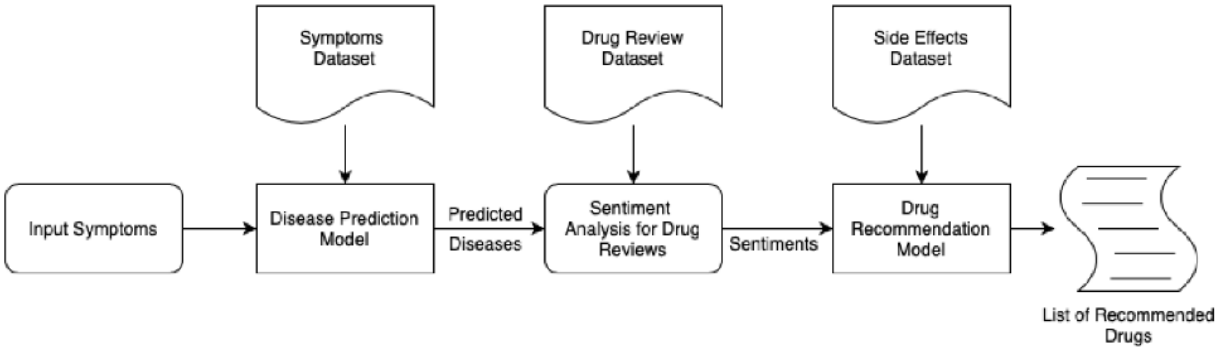

Fig. Overall architecture diagram

Module 1 : Disease Prediction Model

For classification of the symptoms based on diseases, the dataset has been converted it into a new CSV format file which now has symptoms as the columns and diseases as rows. Using one hot encoding, we have mapped every symptom with all the diseases and adding value 1 if it is present for disease and 0 otherwise. The decision tree classifier then predicts the disease based on the symptoms.

INPUT: Symptoms

OUTPUT: Predicted Disease

Module 2: Sentiment Analysis for Drugs Review

We obtain the list of possible drugs the next task is to be able to recommend the best drug for the patient. In accordance, we adopted two different approaches for this. NLP-based approach to analyze the sentiments using a neural net model in order to obtain positive or negative sentiment predictions for the reviews. we have used the VADER tool. It is a simple rule-based model for general sentiment analysis.

INPUT: merged dataset(Symptoms and drug review)

OUTPUT: classification of sentiment

Module 3: Drug Recommendation Model

There are multiple drugs for a single disease. Hence by using sentiment analysis we were able to filter out the negative and neutral reviews leaving us with only the positive ones. We used the weighted average approach for drug recommendation taking note of useful count and average rating offered by the users.

INPUT: Drugs Review (including ratings and useful counts)

OUTPUT: Recommended best drugs.

Datasets Description

1. **Symptoms dataset:**

This dataset is used to take symptoms as input and predict the disease as an output. Dataset is obtained from *the Disease-Symptom Knowledge Database* which is a knowledge database of disease-symptom associations generated by an automated method based on information in textual discharge summaries of patients at New York-Presbyterian Hospital admitted during 2004. There are a total of 149 unique diseases in this dataset and 405 symptoms. Each disease contains 4-5 symptoms corresponding to it.
This dataset contains 3 columns:

1. Disease
2. Count of Disease Occurrence
3. Symptom

2. **Drug Review Dataset:**

This dataset is used in order to take the predicted disease as input and recommend appropriate drugs based on reviews and ratings (Sentiment Analysis). The dataset is gathered from the UCI Machine Learning Repository for Drug Review which provides patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting overall patient satisfaction.

The Repository had two datasets (Test and Train) which are combined for analysis and visualization purposes as they had the same number of columns.
It contains 7 columns and 215063 rows, There are 3671 unique Drug names and 916 unique Conditions (Disease) in this dataset along with the rating and reviews corresponding with the drug names.

1. ID
2. Drug name
3. Condition
4. Review
5. Rating
6. Date
7. Useful count

3. Side Effects Dataset

We have successfully included this dataset containing side effects of specific drugs in order to help patients identify the risks involved in the drug that is being recommended. This dataset is again gathered from UCI Machine Learning Repository for Side Effects of Drugs.

IMPLEMENTATION DETAILS

Import all the required libraries and machine learning models from sklearn package

```
import pandas as pd
import numpy as np
import os

from sklearn.model_selection import train_test_split
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn import svm
```

```
# Reading the symptoms dataset
sym_data = pd.read_csv(path + f"/symptoms_data.csv")
```

```
sym_data.head()
```

	Disease	Heberden's node	Murphy's sign	Stahli's line	abdomen acute	abdominal bloating	abdominal tenderness	abnormal sensation	abnormally hard consistency	abortion	abscess bacterial	absences finding	achalasia
0	Alzheimer's disease	0	0	0	0	0	0	0	0	0	0	0	0
1	Alzheimer's disease	0	0	0	0	0	0	0	0	0	0	0	0
2	Alzheimer's disease	0	0	0	0	0	0	0	0	0	0	0	0
3	Alzheimer's disease	0	0	0	0	0	0	0	0	0	0	0	0

Storing disease and symptoms as lists, and splitting them into train and test sets.

```
sym_data = pd.concat([sym_data]*2, ignore_index=True)
cols = sym_data.columns
cols = cols[1:]
```

```
x = sym_data[cols]
y = sym_data['Disease']
```

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=42)
```

Using Machine learning models to predict disease given symptoms.

```
features = cols
feature_dict = {}
for i,f in enumerate(features):
    feature_dict[f] = i
```

```
print ("DecisionTree")
dt = DecisionTreeClassifier(max_depth=120)
clf_dt=dt.fit(x_train,y_train)
print ("Acurracy: ", clf_dt.score(x_test,y_test))
```

```
DecisionTree
Acurracy:  0.8864144803742119
```

```
input_features = [feature_dict['fever'], feature_dict['cough'], feature_dict['drool']]
input_features
```

```
[122, 70, 89]
```

```
arr =[]
for i in range(len(features)):
  if i == input_features[0]:
    i = int(i/input_features[0])
  elif i == input_features[1]:
    i = int(i/input_features[1])
  elif i == input_features[2]:
    i = int(i/input_features[2])
  else:
    i = 0
  arr.append(i)
```

```
arr = np.array(arr).reshape(-1,len(arr))
```

```
predicted_disease = (mnb.predict(arr))
print("The disease predicted based on given symptoms is : " + predicted_disease[0])
```

```
The disease predicted based on given symptoms is : Alzheimer's disease
```

Importing the preprocessed and merged dataset containing the drug names for every disease, the reviews, symptoms, rating and useful count of all the drugs.

```
from google.colab import drive
drive.mount("/content/drive")


merged_data = pd.read_csv(path + f"/Merged_Dataset.csv")
merged_data.head()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

	Drug	Disease	Review	Rating	UsefulCount	Symptoms
0	Aspirin	transient ischemic attack	"No side effects, easy to take, no more sympt...	10.0	10	['speech slurred', 'dysarthria', 'facial pares...
1	Clopidogrel	transient ischemic attack	"I've been taking this medicine for a lit...	10.0	8	['speech slurred', 'dysarthria', 'facial pares...
2	Clopidogrel	transient ischemic attack	"I took ibuprofen (2 caps at night for severe ...	6.0	13	['speech slurred', 'dysarthria', 'facial pares...
3	Clopidogrel	transient ischemic attack	"After my VAD Stroke I am on plavix. I have a...	5.0	9	['speech slurred', 'dysarthria', 'facial pares...
4	Bayer Children's Aspirin	transient ischemic attack	"No side effects, easy to take, no more sympt...	10.0	10	['speech slurred', 'dysarthria', 'facial pares...

Performing sentiment analysis of drug review using VADER.

```
pip install vaderSentiment
```

```
Collecting vaderSentiment
  Downloading https://files.pythonhosted.org/packages/44/a3/1218a3b5651dbcba1699101c84e5c84c36cbba360d9dbf29f2ff18482982/vaderSentim
     |████████████████████████████████| 133kB 2.6MB/s
Installing collected packages: vaderSentiment
Successfully installed vaderSentiment-3.3.1
```

```
#Importing and installing the necessary library for VADER.

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyser = SentimentIntensityAnalyzer()
```

```
#Making a list of reviews to give it as input for vader analysis.
review_list = list(merged_data.Review)

#Initializing the necessary files.
sentiments = []
positive = []
negative = []
neutral = []
compound = []
```

```
for text in review_list:

  #Performing Vader Analysis on each review.
  com = analyser.polarity_scores(text)["compound"]
  pos = analyser.polarity_scores(text)["pos"]
  neu = analyser.polarity_scores(text)["neu"]
  neg = analyser.polarity_scores(text)["neg"]

  #Adding each value to the corresponding array
  positive.append(pos)
  negative.append(neg)
  neutral.append(neu)
  compound.append(com)
  sentiments.append({"Review":text,
                     "Positive": pos,
                     "Negative": neu,
                     "Neutral": neg,
                     "Compound": com})

sentiments_data = pd.DataFrame.from_dict(sentiments)
```

```
sentiments_data
```

	Review	Positive	Negative	Neutral	Compound
0	"No side effects, easy to take, no more sympt...	0.206	0.638	0.156	0.1779
1	"I've been taking this medicine for a lit...	0.098	0.762	0.140	-0.4092
2	"I took ibuprofen (2 caps at night for severe ...	0.000	0.721	0.279	-0.8176
3	"After my VAD Stroke I am on plavix. I have a...	0.055	0.945	0.000	0.6757
4	"No side effects, easy to take, no more sympt...	0.206	0.638	0.156	0.1779
...	...	...	...	...	...
10055	"Reminds me of Adderall but only a natural way...	0.081	0.919	0.000	0.5023
10056	"My husband has Vascular Dementia for appox. 4...	0.177	0.674	0.149	0.3272
10057	"I've only been using this medication (pa...	0.164	0.638	0.198	-0.1531
10058	"Makes me feel ill."	0.000	0.517	0.483	-0.4215
10059	"Trouble sleeping,stomach pain,headache,consti...	0.000	0.426	0.574	-0.4019

```
#Adding the sentiment analysis columns to the merged dataset.
merged_data["Positive"] = positive
merged_data["Negative"] = negative
merged_data["Neutral"] = neutral
merged_data["Compound"] = compound
merged_data["Review_Sentiment"] = ''
```

```
#Visualizing the merged dataset with the sentiment analysis results.
merged_data.head()
```

	Drug	Disease	Review	Rating	UsefulCount	Symptoms	Positive	Negative	Neutral	Compound	Review_Sentiment
0	Aspirin	transient ischemic attack	"No side effects, easy to take, no more sympt...	10.0	10	['speech slurred', 'dysarthria', 'facial pares...	0.206	0.156	0.638	0.1779	
1	Clopidogrel	transient ischemic attack	"I've been taking this medicine for a lit...	10.0	8	['speech slurred', 'dysarthria', 'facial pares...	0.098	0.140	0.762	-0.4092	
2	Clopidogrel	transient ischemic attack	"I took ibuprofen (2 caps at night for severe ...	6.0	13	['speech slurred', 'dysarthria', 'facial pares...	0.000	0.279	0.721	-0.8176	

Based on the compound value we can determine whether the overall sentiment of the review is positive, negative, or neutral. Below will be the threshold we shall be used for classifying the review sentiment class:

Positive sentiment: compound score >= 0.05

Neutral sentiment: -0.05 < compound score < 0.05

Negative sentiment: compound score <= -0.05

```
merged_data.loc[merged_data['Compound'] >= 0.05, 'Review_Sentiment'] = 'Positive'
merged_data.loc[merged_data['Compound'] <= -0.05, 'Review_Sentiment'] = 'Negative'
merged_data["Review_Sentiment"].replace('', 'Neutral', inplace = True)
```

```
#Dropping the columns Positive, Negative, Neutral and Compound.
merged_data = merged_data.drop(columns = ["Positive", "Negative", "Neutral", "Compound"])
```

```
# Rearranging the columns in different order.
merged_data = merged_data[["Disease", "Drug", "Symptoms", "Review", "Review_Sentiment", "Rating", "UsefulCount"]]
```

```
#Visualizing the merged dataset after adding the sentiment analysis results.
merged_data.head()
```

	Disease	Drug	Symptoms	Review	Review_Sentiment	Rating	UsefulCount
0	transient ischemic attack	Aspirin	['speech slurred', 'dysarthria', 'facial pares...	"No side effects, easy to take, no more sympt...	Positive	10.0	10
1	transient ischemic attack	Clopidogrel	['speech slurred', 'dysarthria', 'facial pares...	"I've been taking this medicine for a lit...	Negative	10.0	8
2	transient ischemic attack	Clopidogrel	['speech slurred', 'dysarthria', 'facial pares...	"I took ibuprofen (2 caps at night for severe ...	Negative	6.0	13
3	transient ischemic attack	Clopidogrel	['speech slurred', 'dysarthria', 'facial pares...	"After my VAD Stroke I am on plavix. I have a...	Positive	5.0	9
4	transient ischemic attack	Bayer Children's Aspirin	['speech slurred', 'dysarthria', 'facial pares...	"No side effects, easy to take, no more sympt...	Positive	10.0	10

Weighted average of rating and useful count,

```
#Reading the dataset with sentiment analysis of reviews
data = pd.read_csv(path + f'/Sentiment_analysis.csv')
```

```
data.head()
```

	Disease	Drug	Symptoms	Review	Review_Sentiment	Rating	UsefulCount
0	transient ischemic attack	Aspirin	['speech slurred', 'dysarthria', 'facial pares...	"No side effects, easy to take, no more sympt...	Positive	10.0	10
1	transient ischemic attack	Clopidogrel	['speech slurred', 'dysarthria', 'facial pares...	"I've been taking this medicine for a lit...	Negative	10.0	8
2	transient ischemic attack	Clopidogrel	['speech slurred', 'dysarthria', 'facial pares...	"I took ibuprofen (2 caps at night for severe ...	Negative	6.0	13
3	transient ischemic attack	Clopidogrel	['speech slurred', 'dysarthria', 'facial pares...	"After my VAD Stroke I am on plavix. I have a...	Positive	5.0	9
4	transient ischemic attack	Bayer Children's Aspirin	['speech slurred', 'dysarthria', 'facial pares...	"No side effects, easy to take, no more sympt...	Positive	10.0	10

```
#Sorting the data based on drug name
data = data.sort_values(['Drug'])
```

```
# Creating a function to calculate weighted average
def wavg(group, avg_name, weight_name):
    d = group[avg_name]
    w = group[weight_name]
    try:
        return (d * w).sum() / w.sum()
    except ZeroDivisionError:
        return d.mean()
```

```
data.groupby(["Disease", "Drug"]).apply(wavg, "Rating", "UsefulCount")
```

```
Disease                   Drug
alzheimer's disease       Aricept                    5.807916
                          Aricept ODT                9.000000
                          Donepezil                  5.503089
                          Exelon                     7.526027
                          Galantamine                8.510638
                                                       ...
schizophrenia             Zyprexa Zydis              6.304348
transient ischemic attack Aspirin                   10.000000
                          Bayer Children's Aspirin  10.000000
                          Clopidogrel                6.766667
                          Plavix                     5.000000
Length: 403, dtype: float64
```

```
# Creating a dataframe of Drug and its average rating
data_wavg = pd.DataFrame(data.groupby(["Drug"]).apply(wavg, "Rating", "UsefulCount").reset_index())
```

```
data_wavg = data_wavg.rename(columns={0: "Rating_Wavg"})
```

```
data_wavg.head()
```

	Drug	Rating_Wavg
0	Abilify	7.350122
1	Abilify Maintena	8.714286
2	Acetaminophen / chlorpheniramine	8.000000
3	Acetaminophen / phenyltoloxamine	10.000000
4	Acetazolamide	6.376068

```
merged_wavg = pd.merge(data_wavg, data, on='Drug')
```

```
# Merging the weighted average column with the dataset
merged_wavg.drop(columns=['Symptoms','Rating'], inplace=True)
merged_wavg = merged_wavg[['Disease','Drug', 'Review', 'Review_Sentiment', 'Rating_Wavg', 'UsefulCount']]
```

```
merged_wavg.head()
```

	Disease	Drug	Review	Review_Sentiment	Rating_Wavg	UsefulCount
0	schizophrenia	Abilify	"This medication has helped my son a great dea...	Positive	7.350122	40
1	schizophrenia	Abilify	"This medication made me feel like my brain wa...	Negative	7.350122	10
2	schizophrenia	Abilify	"I've used 5mg daily for three weeks. My ...	Positive	7.350122	30
3	schizophrenia	Abilify	"Excellent medication! When I was first on Abi...	Negative	7.350122	11
4	schizophrenia	Abilify	"I used to hear bad voices in my head,and thou...	Negative	7.350122	107

Drug Recommendation

```
#Sorting dataset and grouping by disease
merged_wavg = merged_wavg.sort_values(['Disease','Rating_Wavg'],ascending=False, ignore_index=True).groupby('Disease').head(10060)
merged_wavg
```

	Disease	Drug	Review	Review_Sentiment	Rating_Wavg	UsefulCount
0	transient ischemic attack	Aspirin	"No side effects, easy to take, no more sympt...	Positive	10.000000	10
1	transient ischemic attack	Bayer Children's Aspirin	"No side effects, easy to take, no more sympt...	Positive	10.000000	10
2	transient ischemic attack	Clopidogrel	"I've been taking this medicine for a lit...	Negative	6.766667	8
3	transient ischemic attack	Clopidogrel	"I took ibuprofen (2 caps at night for severe ...	Negative	6.766667	13
4	transient ischemic attack	Clopidogrel	"After my VAD Stroke I am on plavix. I have a...	Positive	6.766667	9
...	...	...	...	...	...	...
10055	alzheimer's disease	Namenda	"I have taken it for about 4-5 yes. I feel ab...	Positive	4.100946	38
10056	alzheimer's disease	Namenda	"My mother has been taking Namenda for Alzheim...	Negative	4.100946	120
10057	alzheimer's disease	Namenda	"My mother has been treated with the Exelon (p...	Negative	4.100946	109
10058	alzheimer's disease	Namenda	"My wife (74) has been taking Namenda since th...	Positive	4.100946	55
10059	alzheimer's disease	Namenda	"My wife started with two a day in conjunction...	Negative	4.100946	55

10060 rows × 6 columns

```
# Taking predicted disease as input and recommending drug based on highest weighted average and useful count of ratings
groupedByCount = merged_wavg.groupby(['Disease', 'Drug', 'Rating_Wavg'])['UsefulCount'].sum().reset_index()
```

```
groupedByCount
```

	Disease	Drug	Rating_Wavg	UsefulCount
0	alzheimer's disease	Aricept	5.807916	1454
1	alzheimer's disease	Donepezil	5.503089	1559
2	alzheimer's disease	Exelon	7.526027	80
3	alzheimer's disease	Galantamine	8.510638	71
4	alzheimer's disease	Memantine	4.286241	218
...	...	...	...	...
326	schizophrenia	Zyprexa Zydis	6.304348	19
327	transient ischemic attack	Aspirin	10.000000	10
328	transient ischemic attack	Bayer Children's Aspirin	10.000000	10

```
groupedByDisease = groupedByCount.groupby('Disease')
```

Recommending drugs based on the predicted disease,

```
recommended_drug = pd.DataFrame(groupedByDisease.get_group((predicted_disease[0]).lower()).nlargest(3, ['Rating_Wavg', 'UsefulCount']))
recommended_drug
```

	Disease	Drug	Rating_Wavg	UsefulCount
3	alzheimer's disease	Galantamine	8.510638	71
2	alzheimer's disease	Exelon	7.526027	80
7	alzheimer's disease	Rivastigmine	7.481108	95

```
print("Recommended drugs for this disease are:\n ", recommended_drug["Drug"].unique())
```

```
Recommended drugs for this disease are:
  ['Galantamine' 'Exelon' 'Rivastigmine']
```

Side effects for the recommended drugs,

```
sd = pd.read_csv(path + f'/SideEffects_data.csv')
sd = sd.drop(['condition', 'commentsReview'], axis=1)
sd = sd.dropna()
```

```
sd.sideEffects = sd.sideEffects.replace('Moderate Side Effects', 2)
sd.sideEffects = sd.sideEffects.replace('Mild Side Effects', 2)
sd.sideEffects = sd.sideEffects.replace('Severe Side Effects', 1)
sd.sideEffects = sd.sideEffects.replace('No Side Effects', 3)
sd.sideEffects = sd.sideEffects.replace('Extremely Severe Side Effects', 0)
```

```
sd.effectiveness = sd.effectiveness.replace('Considerably Effective', 2)
sd.effectiveness = sd.effectiveness.replace('Highly Effective', 3)
sd.effectiveness = sd.effectiveness.replace('Marginally Effective', 1)
sd.effectiveness = sd.effectiveness.replace('Moderately Effective', 1)
sd.effectiveness = sd.effectiveness.replace('Ineffective', 0)
```

```
sd = sd[['urlDrugName', 'rating', 'effectiveness', 'sideEffects', 'sideEffectsReview']]
```

```
sd = sd.rename(columns={"urlDrugName": "Drug", "rating": "Rating", "effectiveness": "Effectivness Rating",
                        "sideEffects": "Side Effect Rating", "sideEffectsReview": "Side Effects"})
sd.Drug = sd.Drug.str.upper()
```

```
print('Drug with possible side effects')
sd
```

Drug with possible side effects

	Drug	Rating	Effectivness Rating	Side Effect Rating	Side Effects
0	ENALAPRIL	4	3	2	cough, hypotension , proteinuria, impotence , ...
1	ORTHO-TRI-CYCLEN	1	3	1	Heavy Cycle, Cramps, Hot Flashes, Fatigue, Lon...
2	PONSTEL	10	3	3	Heavier bleeding and clotting than normal.
3	PRILOSEC	3	1	2	Constipation, dry mouth and some mild dizzines...
4	LYRICA	2	1	1	I felt extremely drugged and dopey. Could not...
...	...	...	...	...	...
4146	FORTEO	7	1	2	abdominal pain, confusion, constipation, depre...
4147	ZOLEDRONIC ACID	6	1	2	Agitation, blurred vision, cough, depression, ...
4148	GALANTAMINE	10	3	3	Chest pain or discomfort, lightheadedness, diz...
4149	RIVASTIGMINE	7	3	2	Diarrhea, indigestion, loss of appetite, loss ...
4150	EXELON	5	1	1	an ulcer or stomach bleeding, a seizure, heart...

4149 rows × 5 columns

```
def _probScore(df, w1, w2, w3):
    return(1-(df[w1] * df[w2] * df[w3]).sum()/df[w1].sum()/10)
```

```
def _getSE(df, sedf, l):
    sdf = sedf
    sdf = sdf.loc[sdf['Drug'] == l]
    #print(sdf)
    w = sdf.groupby(["Drug"]).apply(_probScore, "Rating", "Effectivness Rating", "Side Effect Rating")
    sdf1 = sdf.loc[sdf['Effectivness Rating'] == 0]
    sdf1 = sdf1.loc[sdf['Side Effect Rating'] == 0]
    sdf1 = sdf1.loc[sdf['Rating'] <= 1 ]
    sdf1 = sdf1.reset_index()
    sdf1 = pd.DataFrame(sdf['Side Effects'])
    df1 = pd.DataFrame(w, columns=['Prob. of Side Effect'])
    df1 = df1.reset_index()
    #print(sdf[:1]['Side Effects'])
    df2 = pd.DataFrame(sdf['Side Effects'])
    df2 = df2.reset_index()
    df2 = df2.drop(['index'], axis=1)
    dd = pd.concat([df1, df2], axis=1)

    return dd
```

```
def _getSideEffects(df, sedf):
    df.Drug = df.Drug.str.upper()
    l = list(df.Drug)
    #print(l)
    for i in range(0, 1):
        df1 = _getSE(df, sedf, l[i])
    dd = df1
    for i in range(1, len(l)):
        df2 = _getSE(df, sedf, l[i])
        dd = pd.concat([dd, df2])
    return dd
```

```
se = _getSideEffects(recommended_drug, sd)
```

```
recommender = recommended_drug.set_index('Drug').join(se.set_index('Drug'))
recommender = recommender.reset_index()
recommender['Prob. of Side Effect'] = recommender['Prob. of Side Effect'].fillna(0)
recommender['Prob. of Side Effect'] = recommender['Prob. of Side Effect'].fillna('Not Available')
print('Mapped recommended drugs with possible side effects and probabilistic score')
recommender
```

Mapped recommended drugs with possible side effects and probabilistic score

	Drug	Disease	Rating_Wavg	UsefulCount	Prob. of Side Effect	Side Effects
0	GALANTAMINE	alzheimer's disease	8.510638	71	0.1	Chest pain or discomfort, lightheadedness, diz...
1	EXELON	alzheimer's disease	7.526027	80	0.9	an ulcer or stomach bleeding, a seizure, heart...
2	RIVASTIGMINE	alzheimer's disease	7.481108	95	0.4	Diarrhea, indigestion, loss of appetite, loss ...

```
#sorting based on probabilistic score
recommender = recommender.sort_values('Prob. of Side Effect')
print("The recommended Drugs for the given Disease along with possible side effects is:")
recommender
```

The recommended Drugs for the given Disease is:

	Drug	Disease	Rating_Wavg	UsefulCount	Prob. of Side Effect	Side Effects
0	GALANTAMINE	alzheimer's disease	8.510638	71	0.1	Chest pain or discomfort, lightheadedness, diz...
2	RIVASTIGMINE	alzheimer's disease	7.481108	95	0.4	Diarrhea, indigestion, loss of appetite, loss ...
1	EXELON	alzheimer's disease	7.526027	80	0.9	an ulcer or stomach bleeding, a seizure, heart...

METRICS FOR EVALUATION

Accuracy

Accuracy is used in classification problems to tell the percentage of correct predictions made by a model. Accuracy score in machine learning is an evaluation metric that measures the number of correct predictions made by a model in relation to the total number of predictions made. We calculate it by dividing the number of correct predictions by the total number of predictions.

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

This formula provides an easy-to-understand definition that assumes a **binary classification** problem. In this model, we have obtained around 88% accurate results using decision tree classifier to predict diseases for a given array of inputs.

CONCLUSION

Drug Recommendation systems are a prevailing technology in today's online services and with the rise of requirements for these services there is more and more need to automate the processes subsequently we have designed a drug recommendation system. Below are the key conclusions from our project.
Successfully built a drug recommendation system that predicts diseases and recommends drugs along with possible side effects based on user symptoms input. We designed three models for this project implementation. A disease prediction model, a Sentiment Analysis model, and a recommendation model.
Experimented with different approaches for each of the three models. Each of the three models gave good accuracies contributing to the overall reliability of the drug recommendation model and we finally chose the decision tree classifier to predict the disease has it provided a higher accuracy rate when compared to the other models.

REFERENCES

[1] Kamaraj, K.Gomathi & Priyaa, D.Shanmuga. (2016). Multi Disease Prediction using Data Mining Techniques. International Journal of System and Software Engineering.

[2] M.A.Nishara Banu, B Gomathy. (2013). Disease Predicting System Using Data Mining Techniques. International Journal of Technical Research and Applications.

[3] H Wang Q Gu J Wei Z Cao Q Liu (2015). Mining drug–disease relationships as a complement to medical genetics-based drug repositioning: Where a recommendation system meets genome-wide association studies.

[4] Yin Zhang, Daqiang Zhang, Mohammad Mehedi Hassan, Atif Alamri & Limei Peng (2014). CADRE: Cloud-Assisted Drug Recommendation Service for Online Pharmacies.

[5] Druglib.com - Drug Information, Research, Clinical Trials, News. http://www.druglib.com/

[6] Hutto, C.J. & Gilbert, Eric. (2015). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014.

[7] Y. Bao and X. Jiang, ”An intelligent medicine recommender system framework,” 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), Hefei, 2016, pp. 1383-1388, doi: 10.1109/ICIEA.2016.7603801.

[8] Zhang, Yin & Zhang, Dafang & Hassan, Mohammad & Alamri, Atif & Peng, Limei. (2014). CADRE: Cloud-Assisted Drug REcommendation Service for Online Pharmacies. Mobile Networks and Applications. 20. 348-355. 10.1007/s11036-014-0537-4.

[9] J. Li, H. Xu, X. He, J. Deng and X. Sun, ”Tweet modeling with LSTM recurrent neural networks for hashtag recommendation,” 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 1570-1577, doi: 10.1109/IJCNN.2016.7727385

[10] Zhang, Yin & Jin, Rong & Zhou, Zhi-Hua. (2010). Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics. 1. 43-52. 10.1007/s13042-010-0001- 0.