Data Mining Project Report

Analysis of Medical Drugs Data: Learning “useful drugs” based on the Prediction of Sentiment Score

By Aashish Kakar, Aayush Kumar, Mohammed Yunus, Uday Karnam

Abstract:

Online review sites and opinion forums contain a wealth of information regarding user preferences and experiences over multiple product domains. This information can be leveraged to obtain valuable insights using data mining approaches such as sentiment analysis. In this work, we examine online user reviews within the pharmaceutical field. Online user reviews in this domain contain information related to multiple aspects such as the effectiveness of drugs and side effects, which make automatic analysis very interesting but also challenging. However, analyzing sentiments concerning the various aspects of drug reviews can provide valuable insights, help with decision making and improve monitoring public health by revealing collective experience. DrugLib.com is a comprehensive drug database organized by relevance to specific drugs.

Dataset:

The dataset contains patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting overall patient satisfaction.

- The test dataset has 40k records and the training dataset has 120k records
- drugName (categorical): name of the drug
- condition (categorical): name of the condition
- review (text): patient review
- rating (numerical): 10-star patient rating. This is also the Target variable.
- date (date): date of review entry
- usefulCount (numerical): number of users who found the review to be useful
- Tokenized_Score(numerical): This will show the sentiment.

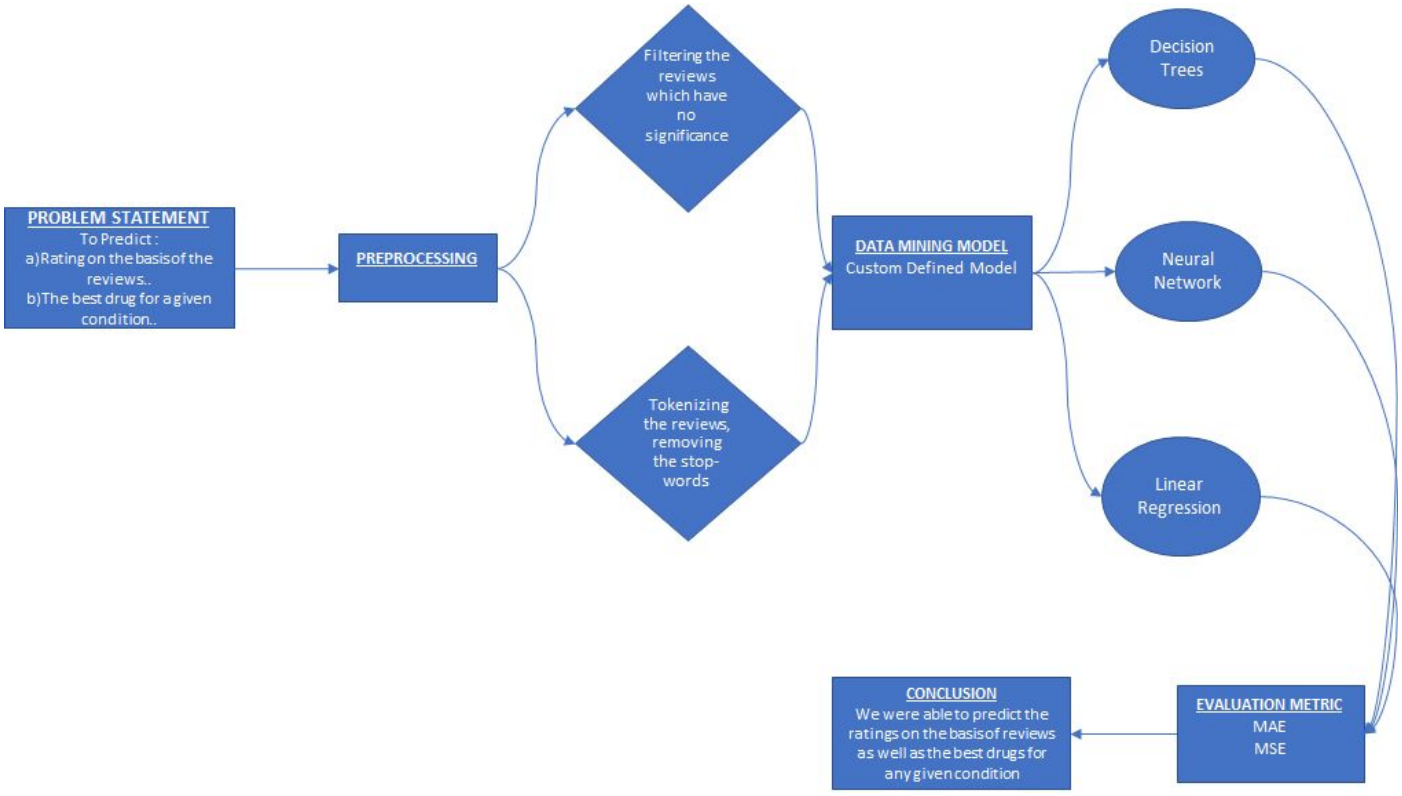
Motivation:

There are many drugs in the market which can be prescribed for a multiple number of diseases or conditions but which drug is better the other for a given condition- this is the motivation behind our project. We ran sentiment analysis on the reviews given by the users and then came up with a drug for a particular condition that had the most positives out of the lot.

Problem Statement:

1. How good are the reviews and ratings correlated?
2. To identify which drug performs the best for a given medical condition.

Methodology:


PROBLEM STATEMENT
To Predict :
a)Rating on the basis of the reviews..
b)The best drug for a given condition..
PREPROCESSING
Filtering the reviews which have no significance
Tokenizing the reviews, removing the stop-words
DATA MINING MODEL
Custom Defined Model
Decision Trees
Neural Network
Linear Regression
EVALUATION METRIC
MAE
MSE
CONCLUSION
We were able to predict the ratings on the basis of reviews as well as the best drugs for any given condition

Pre-Processing:

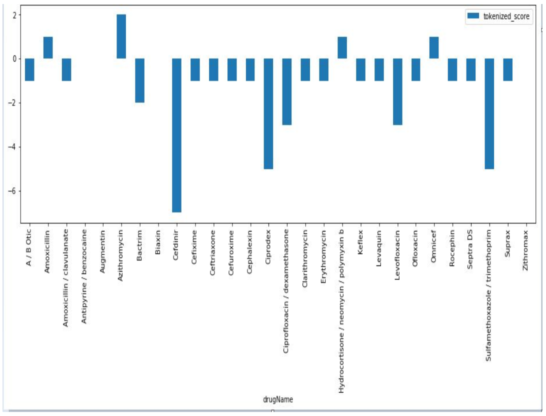
- The first step was to remove conditions which did not look like a condition from the conditions column (e.g expressions like “2 users found this comment useful”)
- We have used **tokenization** to split the review into tokens of size 1.

- In order to remove words which have no functional importance, we have made use of **stopword removal**.
- Defined **polarity index for** every word in the vocab file. The polarity index is added as a new column in the vocab file.

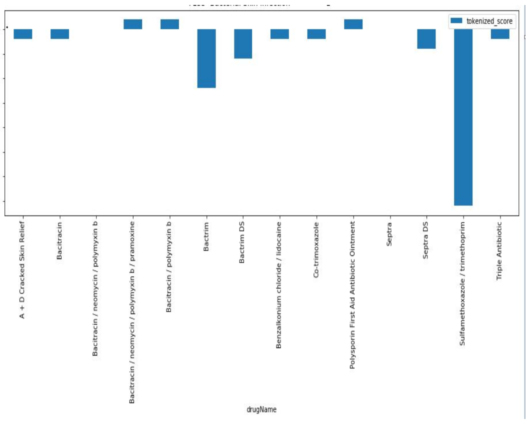
Data mining models:

Generating most "useful" drugs for a condition (Learning patterns from data):

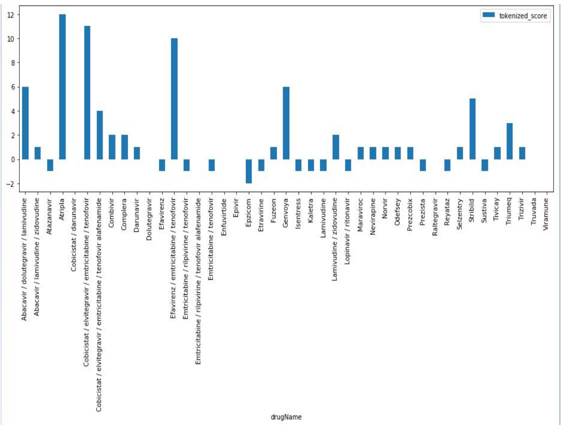
- We used a custom function to build a model to perform sentiment analysis.
- This function reads from the dictionary and assigns a score to every individual word present in the tokenized review column.
- The sentiment function declared above is applied to each and every word encountered in the review column and then the sentiments are summed up.
- We have selected four conditions and plotted visualizations to find the best drug for these conditions based on the tokenized score. The four conditions are:
    1. Otitis Media
    2. Bacterial Skin Infection
    3. HIV Infection
    4. Juvenile Idiopathic Arthritis

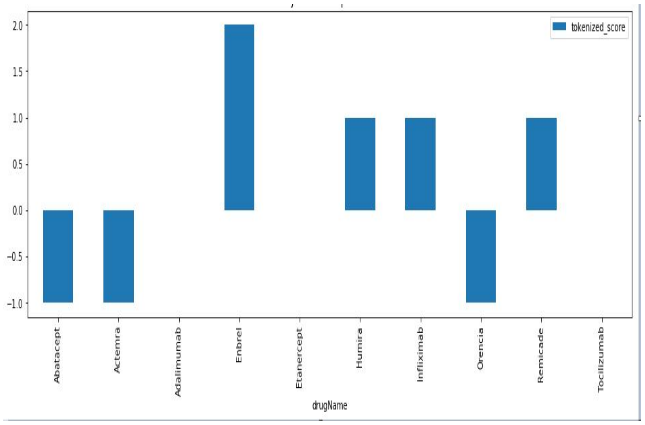


Otitis Media: Azithromycin



Bacterial Skin Infection: Bacitracin



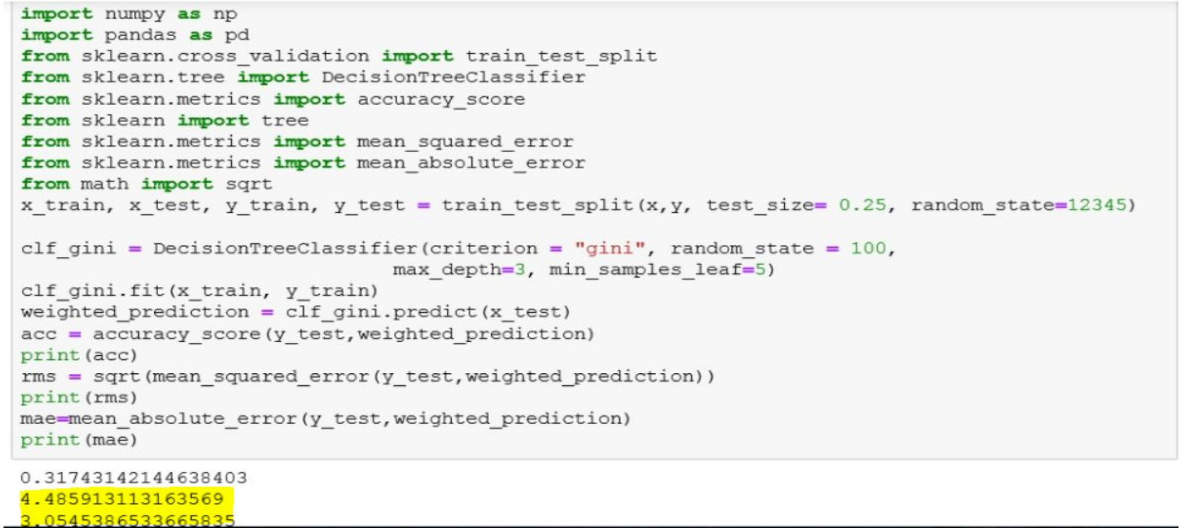


HIV: Atripla

Arthritis: Enbrel

Prediction of Ratings based on Sentiment Score of Reviews:

We also calculated the RMS and MAE values using Neural Network Classifier, Decision tree and Linear Regression models. RMS and MAE values are our evaluation metrics.

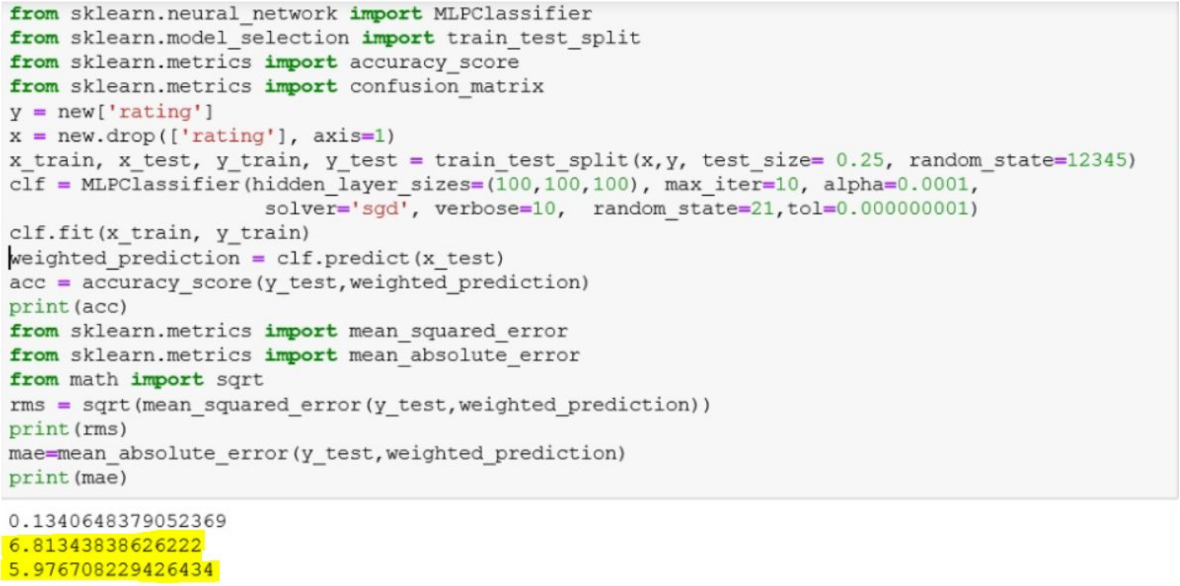


```
import numpy as np
import pandas as pd
from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from math import sqrt
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size= 0.25, random_state=12345)

clf_gini = DecisionTreeClassifier(criterion = "gini", random_state = 100,
                               max_depth=3, min_samples_leaf=5)
clf_gini.fit(x_train, y_train)
weighted_prediction = clf_gini.predict(x_test)
acc = accuracy_score(y_test,weighted_prediction)
print(acc)
rms = sqrt(mean_squared_error(y_test,weighted_prediction))
print(rms)
mae=mean_absolute_error(y_test,weighted_prediction)
print(mae)
```

```
0.31743142144638403
4.485913113163569
3.0545386533665835
```

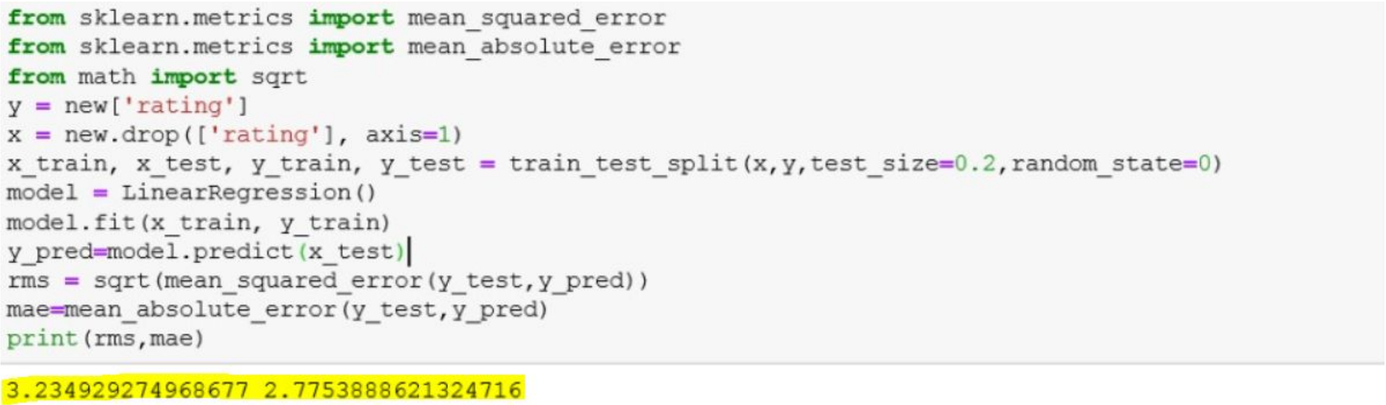
Decision Tree with the Root Mean Square Error & Mean Absolute Error respectively.



```
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
y = new['rating']
x = new.drop(['rating'], axis=1)
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size= 0.25, random_state=12345)
clf = MLPClassifier(hidden_layer_sizes=(100,100,100), max_iter=10, alpha=0.0001,
                     solver='sgd', verbose=10,  random_state=21,tol=0.000000001)
clf.fit(x_train, y_train)
weighted_prediction = clf.predict(x_test)
acc = accuracy_score(y_test,weighted_prediction)
print(acc)
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from math import sqrt
rms = sqrt(mean_squared_error(y_test,weighted_prediction))
print(rms)
mae=mean_absolute_error(y_test,weighted_prediction)
print(mae)
```

```
0.1340648379052369
6.81343838626222
5.976708229426434
```

Neural Network with the Root Mean Square Error & Mean Absolute Error respectively.



```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from math import sqrt
y = new['rating']
x = new.drop(['rating'], axis=1)
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,random_state=0)
model = LinearRegression()
model.fit(x_train, y_train)
y_pred=model.predict(x_test)
rms = sqrt(mean_squared_error(y_test,y_pred))
mae=mean_absolute_error(y_test,y_pred)
print(rms,mae)
```

```
3.234929274968677 2.7753888621324716
```

Linear Regression with the Root Mean Square Error & Mean Absolute Error respectively which performs the best of the three.

Challenges:

- Some reviews had statements like “This drug is not that bad” or “This drug is not that good”. For these statements, the sentiments were wrongly predicted since ‘not’ was considered to be a stop word.
- As NLTK(Natural Language Toolkit) has its own set of stopwords, it becomes difficult for the classifiers to make the correct meaning out of it according to the dataset.

- Since the target variable is a numerical value, the accuracy of their prediction was too low. Hence, we went with Root Mean Square and Mean absolute Error as the evaluation metric.

Inference:

- We are able to learn interesting patterns like which drug is most “useful” for a given condition based on the sentiment score.
- For cases where multiple drugs are having equivalent sentiment score, multiple drugs are designated as “useful drugs”.
- We are able to predict ratings based on the sentiment of the reviews with a mean absolute error of 3.05, 5.97, 2.77 for models built using Decision-trees, Neural Networks and Linear Regression respectively.
- Linear regression performs the best among the three models when considering RMS and MAE as the evaluation metrics.

Improvements:

- Could have made own set of stopwords according to the dataset and then by adding bigrams or trigrams, it could have given better RME/MAE results for our models.
- Parts of Speech Tagging method could have been used. This is a method which extracts noun phrases and builds trees representing relationships between noun phrases and the other parts of the sentence.

Solution:

Therefore, as we are able to predict the ratings with a mean absolute error of 2.77(Linear Regression), we can incorporate this data model component in the sales department in order to sell drugs which are more useful for the users and in this process the firm can make some good fortunes.

References:

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

http://www.druglib.com/

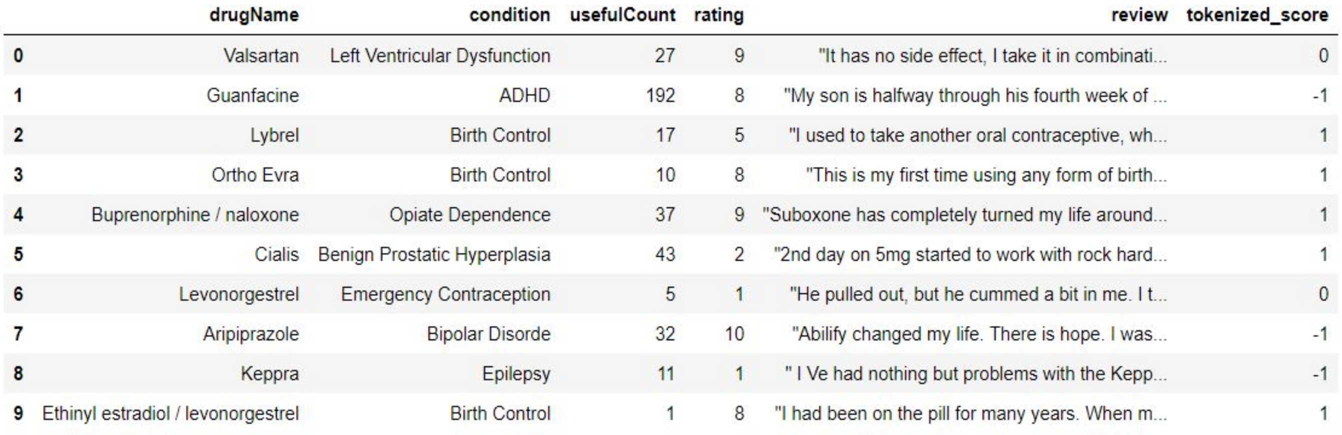
http://scikit-learn.org/stable/supervised_learning.html#supervised-learning

https://pandas.pydata.org/

https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/

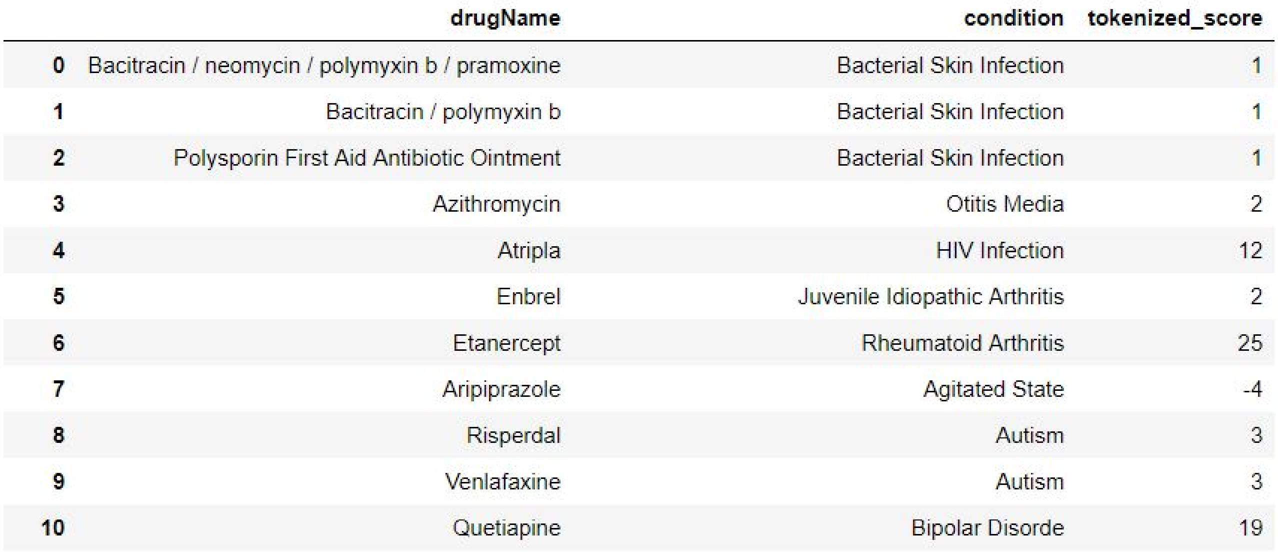
Appendix:

Data-Frame where the ratings are predicted according to the useful-count.



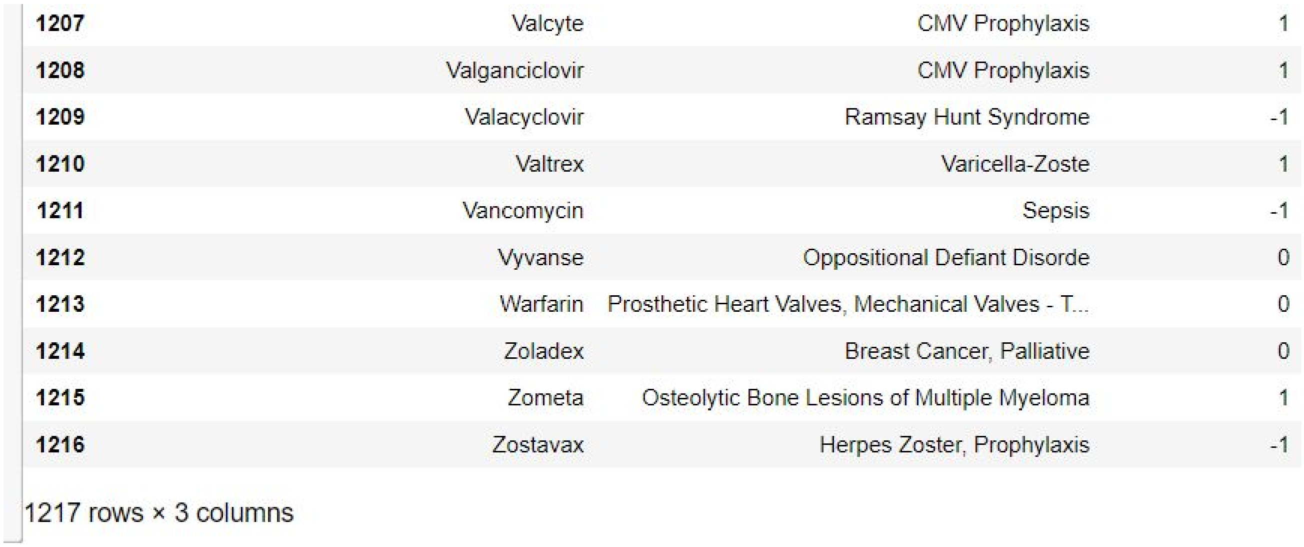
	drugName	condition	usefulCount	rating	review	tokenized_score
0	Valsartan	Left Ventricular Dysfunction	27	9	"It has no side effect, I take it in combinati...	0
1	Guanfacine	ADHD	192	8	"My son is halfway through his fourth week of ...	-1
2	Lybrel	Birth Control	17	5	"I used to take another oral contraceptive, wh...	1
3	Ortho Evra	Birth Control	10	8	"This is my first time using any form of birth...	1
4	Buprenorphine / naloxone	Opiate Dependence	37	9	"Suboxone has completely turned my life around...	1
5	Cialis	Benign Prostatic Hyperplasia	43	2	"2nd day on 5mg started to work with rock hard...	1
6	Levonorgestrel	Emergency Contraception	5	1	"He pulled out, but he cummed a bit in me. I t...	0
7	Aripiprazole	Bipolar Disorde	32	10	"Abilify changed my life. There is hope. I was...	-1
8	Keppra	Epilepsy	11	1	" I Ve had nothing but problems with the Kepp...	-1
9	Ethinyl estradiol / levonorgestrel	Birth Control	1	8	"I had been on the pill for many years. When m...	1

The final data-frame in which tokenized score is given for each condition:



	drugName	condition	tokenized_score
0	Bacitracin / neomycin / polymyxin b / pramoxine	Bacterial Skin Infection	1
1	Bacitracin / polymyxin b	Bacterial Skin Infection	1
2	Polysporin First Aid Antibiotic Ointment	Bacterial Skin Infection	1
3	Azithromycin	Otitis Media	2
4	Atripla	HIV Infection	12
5	Enbrel	Juvenile Idiopathic Arthritis	2
6	Etanercept	Rheumatoid Arthritis	25
7	Aripiprazole	Agitated State	-4
8	Risperdal	Autism	3
9	Venlafaxine	Autism	3
10	Quetiapine	Bipolar Disorde	19

.
.
.



1207	Valcyte	CMV Prophylaxis	1
1208	Valganciclovir	CMV Prophylaxis	1
1209	Valacyclovir	Ramsay Hunt Syndrome	-1
1210	Valtrex	Varicella-Zoste	1
1211	Vancomycin	Sepsis	-1
1212	Vyvanse	Oppositional Defiant Disorde	0
1213	Warfarin	Prosthetic Heart Valves, Mechanical Valves - T...	0
1214	Zoladex	Breast Cancer, Palliative	0
1215	Zometa	Osteolytic Bone Lesions of Multiple Myeloma	1
1216	Zostavax	Herpes Zoster, Prophylaxis	-1

1217 rows × 3 columns

Jargon:

Tokenization: It is the process of splitting the review into tokens of size one.

Stop-word removal: It is the process of removing stop-words(words without functional importance).

Stemming: It is the process of generating the root words and eliminate redundant words.