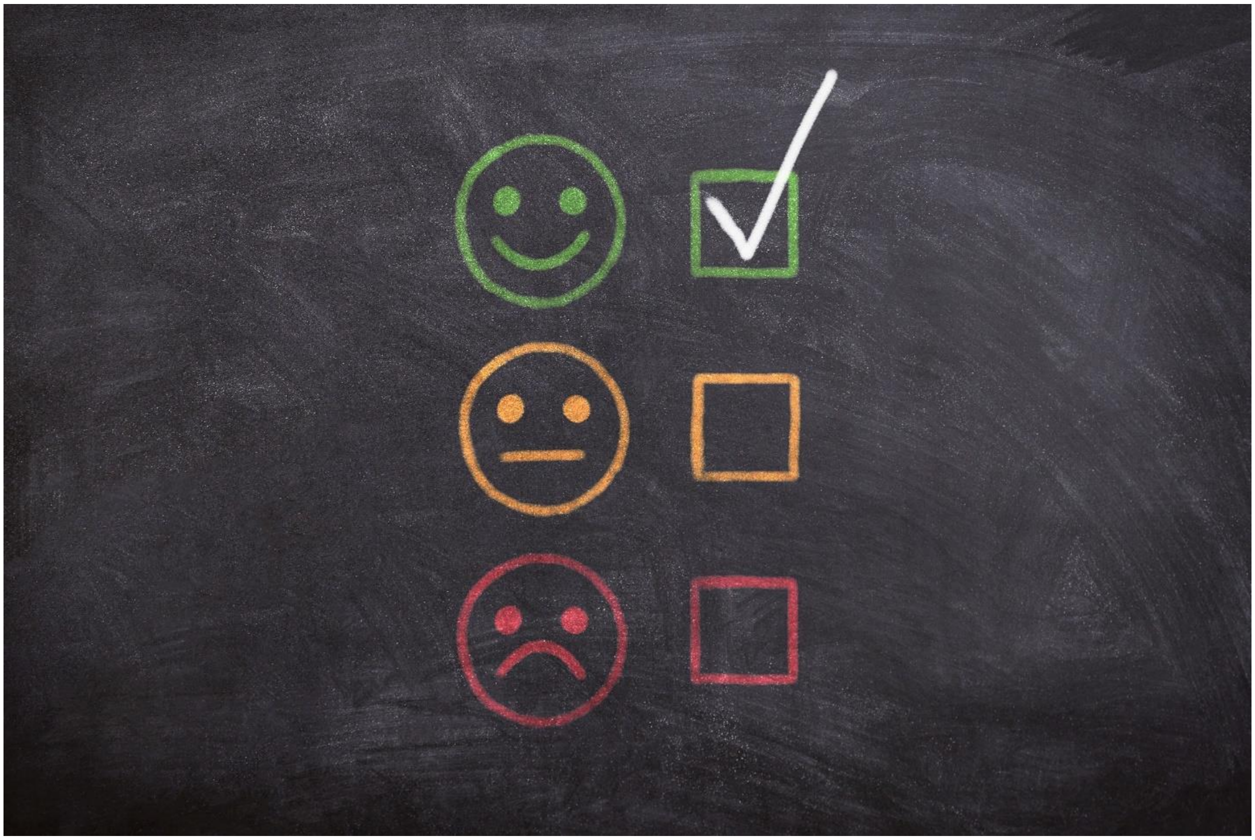


DRUG RATING AND REVIEW RELATIONS

Using a drug and giving a review is a movement which effects drugs' prescribing by doctors and effectiveness for patients while using them. In this project, a patient with a serious condition, is given a drug by a professional. If patient finds it beneficial after using, gives a positive review and high rating for it. Therefore, if another patient finds the review helpful, gives one point which will go to the usefulness of the reviews. The aim of this project is analyzing the reviews and making predictions of findings if they really fit with the real ratings.

DATA

This dataset is originally from the UCI Machine Learning repository.

Citation: Felix Gräßer, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. 2018. Aspect-Based Sentiment Analysis of Drug Reviews Applying

Cross-Domain and Cross-Data Learning. In Proceedings of the 2018 International Conference on Digital Health (DH '18). ACM, New York, NY, USA, 121-125.

GOAL

In consideration of the dataset, these are the questions we want to answer:

- How accurate we can predict ratings based on reviews?
- Do identifying the sentiment of a review help us to predict the rating?
- Which side do people have tendency to give rating: positive, negative, neutral?

DATA CLEANING AND WRANGLING

Data content:

uniqueID: Unique ID for every patient

drugName: Name of the drug

condition: Name of the condition

review: Patient review

rating: Patient rating out of 10

date: Date of given review

usefulCount: Number of users who found review useful

All features were examined to be ready for the next steps. Dataset shape, feature types were checked. It was seen, there were NULL values in ‘condition’ column which was lower than 1% hence they were dropped.

EXPLORATORY DATA ANALYSIS

For the model prediction, there are 2 important columns which matter to apply, rating and reviews, were focused in this section.

- A visualization technique, Word Cloud was used to show the most used words.

The most used words for reviews which has rating 10:

The most used words for reviews which has rating 1:

As it is seen from the images, there is not a clear pattern for words as bad and good reviews.

- VADER Sentiment Intensity Analyzer was applied for reviews: All named and numeric character references (e.g. &gt, >, >) in the string were converted to the corresponding Unicode characters.

The probability of the sentiment is positive : compound score>=0.05

The probability of the sentiment is neutral : compound score between -0.05 and 0.05

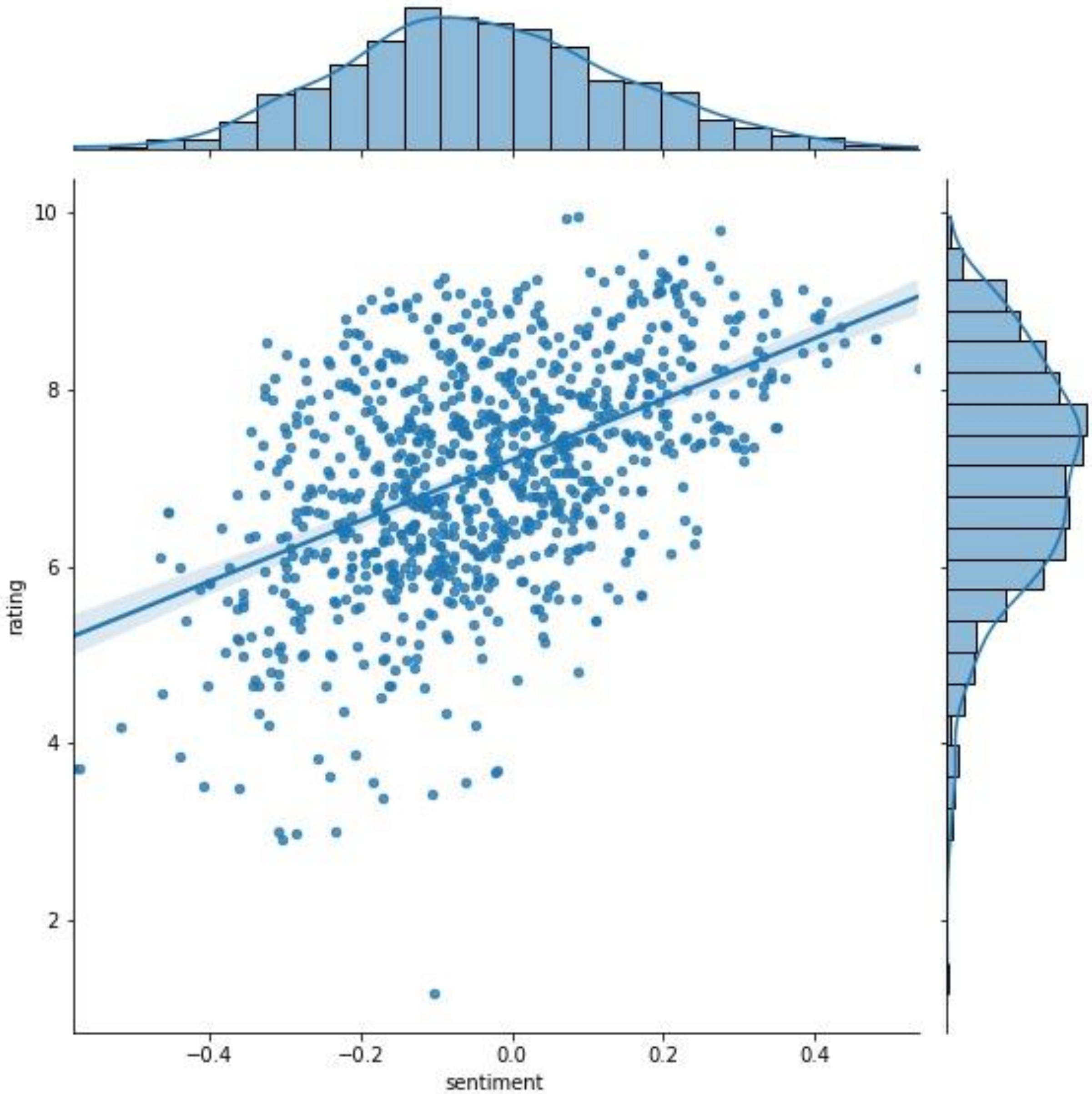
The probability of the sentiment is negative : compound score<=-0.05

compound: The normalized compound score which calculates the sum of all lexicon ratings and takes values from -1 to 1.

	review	rating	sentiment
4	"Suboxone has completely turned my life around...	9	0.9403
4	"I have been on this birth control for one cyc...	9	0.9559
75	"I've had mine for over a year and noticed the...	6	0.7777
75	"Pregabalin for me was miraculous. On the firs...	10	0.8809
157	"I got this inserted 3 years ago, my arm hurt ...	9	-0.4215
157	"I switched to Gianvi from Aviane in hopes of ...	4	-0.0545
254	"My Dr agreed to over see putting me on Qsymia...	9	0.7033
254	"On day 17 of 25 day Efudex fluorouracil 5% cr...	10	0.1842
578	"This medication completely changed my life fo...	10	-0.7224
578	"I have been on various medications for years,...	10	0.9711

It is seen from the figure that sentiment analyze did not work well. When we look at the rating 10, sentiment is seen both 0.9 and -0.7. But we need positive sentiment for rating 10.

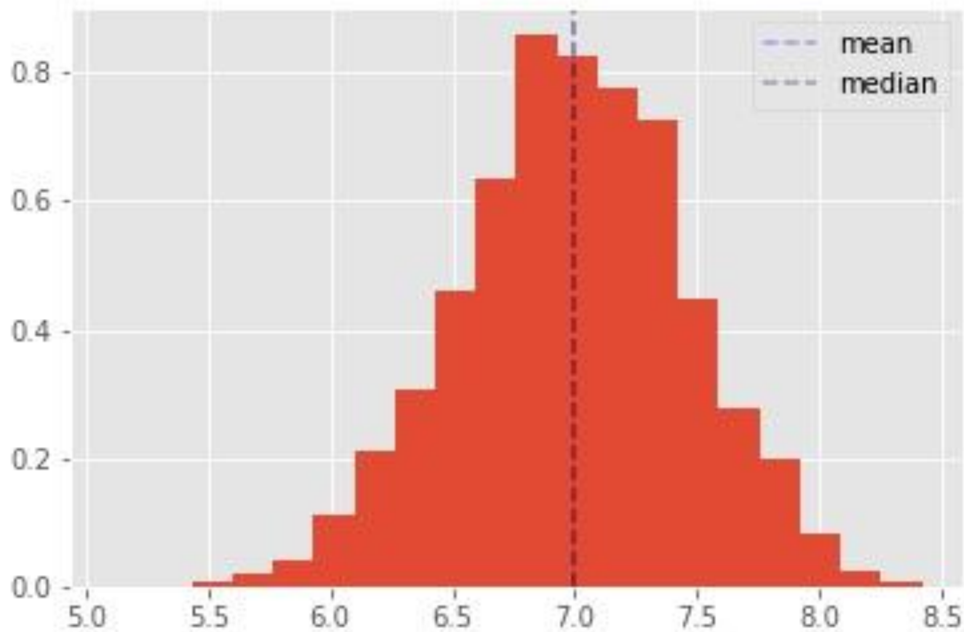
- Vader performance was checked on the average ratings in the most reviewed drugs. 75% quantile was used to see the most reviewed drugs.



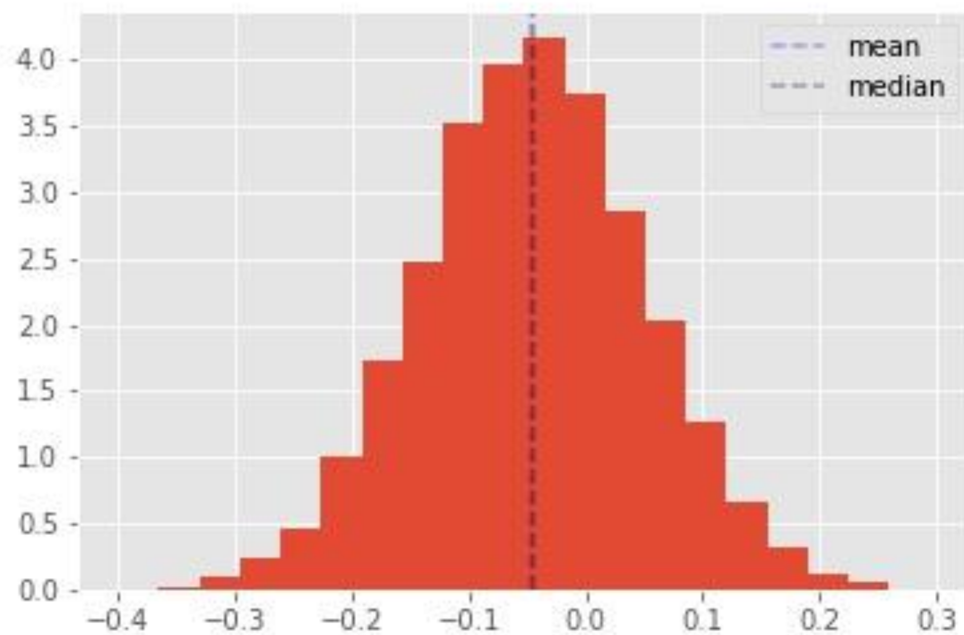
There is a moderate correlation between sentiment and rating for the most reviewed drugs.

- Mean of the rating and sentiment was checked:

Rating mean is 7



Sentiment mean is neutral



PRE-PROCESSING

Using this knowledge, the rating mean is 7, ratings were separated 2 group to make classification work better.

rating>7 was assigned 1

rating<=7 was assigned 0

After this separation, last step is pre-processing for reviews which includes:

1. Remove HTML tags
2. Remove Stop Words
3. Remove symbols and special characters
4. Tokenize
5. Stemming

We will predict the new_sentiment by using the cleaned reviews.

Before modeling, we have to convert the review to numeric values by using TF IDF Vectorizer. For this step, by using the bag-of-words matrix, the tf-idf will be created. Then the bag-of-words representation will be normalized.

The TF counts how many times a word has repeated in a given corpus. Since a corpus is occured by many documents, each documents and its words will have their own TF count. As for IDF, it counts how rarely a word occurs in a document.

PREDICTIONS

1. Three model, LightGBM, Random Forest and Logistic Regression were used with different hyperparameter tunings. The best prediction was gained by LightGBM, by applying GridSearchCV with cross validation.

Accuracy score is 0.8608161936793812
Training set score: 0.9785
Test set score: 0.8608

Confusion matrix

```
 [[19928  5847]
 [ 3133 35611]]
```

True Positives(TP) = 19928

True Negatives(TN) = 35611

False Positives(FP) = 5847

False Negatives(FN) = 3133

	precision	recall	f1-score	support
0	0.86	0.77	0.82	25775
1	0.86	0.92	0.89	38744
accuracy			0.86	64519
macro avg	0.86	0.85	0.85	64519
weighted avg	0.86	0.86	0.86	64519

2. Word Embedding -Word2vec

After cleaning the text data - removing punctuations and stopwords, tokenizing the sentences and lemmatizing the words to their original form - word2vec model was trained. Word embeddings were created on cleaned text data by using word2vec.

Some functions of gensim word2vec:

- most_similar('effect')

```
[('effects', 0.8488506078720093),
 ('affects', 0.8269054293632507),
 ('affect', 0.6979695558547974),
 ('effectsthe', 0.6248028874397278),
 ('incurr', 0.5831418037414551),
 ('swtiching', 0.5422803163528442),
 ('effectsno', 0.5152512788772583),
 ('jehovah', 0.5115261077880859),
 ('tabletsthere', 0.4891676902770996),
 ('effests', 0.4779714345932007)]
```

- similar_by_word('doctor')

[('dr', 0.9641481637954712),
('doc', 0.9321452379226685),
('gp', 0.8473139405250549),
('pcp', 0.8405479192733765),
('md', 0.8237873315811157),
('obgyn', 0.797286868095398),
('physician', 0.7918844819068909),
('gyno', 0.7623308300971985),
('doctors', 0.7620154023170471),
('gynecologist', 0.7492671608924866)]

After loading the word embeddings, the data was padded to have similar length and vectorized the text samples into a 2D integer tensor. Data splitting and embedding matrix for words were done.

- Creating neural network:

  Use word embeddings from word2vec in first layer

  Build Conv1D, GRU, LSTM network

  Add Dense layers

Training the network by using Word2vec, embedding-Conv1D gave the highest accuracy.

Test score: 1.0388768911361694
Test accuracy: 0.8929833769798279

It seems like the neural network gives the best overall accuracy with 89.2%.

FUTURE IMPROVEMENTS

- Trying different feature explorations in different ways would be helpful in developing insights and meaningful conclusions.
- Trying different sets of parameters for classification models would result higher accuracy.
- Applying different Neural Network architecture would be helpful to improve the result.