

Case 3. Patient Drug Review Analysis

Machine Learning for Health Technology Applications

Spring 2020

Sakari Lukkarinen

Helsinki Metropolia University of Applied Sciences

Case 3. Patient Drug Review

Aim is

- to predict the patient's rating for the drug (=output)
- based on patient's review (text) (=input)

Data from Drugs.com

- 200,000 patient drug reviews
 - drugName - Name of drug (e.g. Levonorgestrel, Nexaplon, ...)
 - condition - Name of condition (e.g. birth control, depression, pain, ...)
 - Patient review (string) - patient's review about the drug for specific condition
 - Rating (1..10) – patient's rating for the drug
 - Date – date of review
 - UsefulCount – number of users who found review useful

Find Drugs & Conditions

Enter drug name or medical condition, pill imprint, etc.



Trending searches: [gabapentin](#), [amlodipine](#), [lisinopril](#), [tramadol](#), [prednisone](#)



Drugs & Medications



Pill Identifier



Interactions Checker



Side Effects



Browse Drugs

[Browse Conditions](#)

A B C D E F G H I J K L M

N O P Q R S T U V W X Y Z

0-9

[Advanced Search](#)

Browse A-Z: [Drug](#), [Treatment](#), [Condition](#) or [Class](#)

Browse by Site Section

[Drugs A-Z](#)

[Side Effects Checker](#)

[Dosage Guidelines](#)

[Manage your Meds](#)

[Mobile Apps](#)

[Health Professionals](#)

[Medical News](#)

[FDA Alerts](#)

[New Drugs](#)

[More](#)



FDA
Medwatch Alert



New Drug
Approval



[Support](#)

Drugs.com Content and Services Guide

Our mission is to provide the most accurate, up-to-date drug information on the Internet. See our [editorial policy](#) for more information on our content aimed at consumers and healthcare professionals. Supplementary services include interactive tools such as the [Pill Finder](#), [Drug Interactions Checker](#) and [Mednotes personal medication records](#).

Consumer Drug Information

- [Complete A to Z Drug List](#)
 - [MedFacts Consumer Information](#)
 - [Multum Consumer Information](#)
 - [Multum Info en Español](#)
 - [IBM Watson Micromedex Advanced Consumer](#)
 - [MedFacts Natural Products](#)
 - [OTC Drug Database](#)
 - [PDR \(no longer available\)](#)
- [Drugs by Condition](#)
- [Drugs by Classification](#)
- [FDA Medwatch Drug Alerts](#)

Interactive Tools & Resources

- [Medical Transcriptionist Search](#)
- [MedNotes - Medication Records](#)
- [Pill Identifier / Finder](#)
 - [Pill Logo Identification](#)
 - [Drug Image Search](#)
 - [Drug Imprint Codes](#)
- [Drug Interaction Checker](#)
- [Symptom Checker](#)
- [Drug Comparison Tool](#)
- [Measurement Conversion Calculator](#)
- [Drug Half Life Calculator](#)
- [Phonetic & Wildcard Search](#)

Recently Approved

[Nexletol](#)

Nexletol (bempedoic acid) is a first-in-class, adenosine triphosphate-citrate...

[Vyepi](#)

Vyepi (eptinezumab-jjmr) is a calcitonin gene-related peptide antagonist...

[Anjeso](#)

Anjeso (meloxicam) is an NSAID injection indicated for use in adults for the...

[Twirla](#)


Twirla (ethinyl estradiol and levonorgestrel transdermal system) is a low-dose...

[More...](#)

Subscribe to our newsletters

[FDA alerts](#)[Daily news summary](#)[Weekly news roundup](#)[Monthly newsletter](#)[I accept the Terms and Privacy Policy](#)[Subscribe](#)

Drugs A to Z

 [Print](#)  [Share](#)

Alvesco

Generic Name: [ciclesonide](#) (inhalation) (syeh KLES oh nide)**Brand Names:** *Alvesco HFA*Medically reviewed by [Sophia Entringer, PharmD](#) Last updated on Nov 26, 2019.[Overview](#)[Side Effects](#)[Dosage](#)[Professional](#)[Interactions](#)[More](#)

What is Alvesco?

Alvesco (ciclesonide) is a man-made corticosteroid. It prevents the release of substances in the body that cause inflammation.

Alvesco is used to prevent asthma attacks in adults and children who are at least 12 years. When used regularly, as prescribed by your health care provider, it will help to prevent and control symptoms of asthma.

Alvesco may also be used for purposes not listed in this medication guide.

Important information

Alvesco inhalation will not work fast enough to treat an asthma attack. Use only a fast acting inhalation medicine for an asthma attack. Tell your doctor if it seems like your asthma medications don't work as well.

Steroid medication can weaken your immune system, making it easier for you to get an infection. Steroids can also worsen an infection you already have, or reactivate an infection you recently had. Before taking Alvesco, tell your doctor about any illness or infection you have had within the past several weeks.

DRUG STATUS

**Availability**
Prescription only**Pregnancy & Lactation**
Risk data available**CSA Schedule***
Not a controlled drug**Approval History**
Drug history at FDA

NEWS

[Price Hikes Have Patients Turning to Craigslist for Insulin, Asthma Inhalers](#)

Manufacturer

[Sunovion Pharmaceuticals Inc.](#)

Drug Class

[Inhaled corticosteroids](#)

Related Drugs

[prednisone](#), [Symbicort](#), [Ventolin](#), [Breo Ellipta](#), [Ventolin HFA](#), [Dulera](#), [Atrovent](#), [Xopenex](#), [Nucala](#)

User Reviews & Ratings

[Alvesco reviews](#)

7.7 / 10

20 Reviews

<https://www.drugs.com/alvesco.html>

Drug reviews - Alvesco





User Reviews for Alvesco

The following information is NOT intended to endorse any particular medication. While these reviews might be helpful, they are not a substitute for the expertise, skill, knowledge and judgement of healthcare practitioners.

[Overview](#) [Side Effects](#) [Dosage](#) [Professional](#) [Interactions](#) [More](#) 

Filter by condition:

--- all conditions --- 


Condition 	Avg. Ratings 	Reviews	Compare
Asthma, Maintenance	8.2 	13 reviews	123 medications
Asthma	6.9 	7 reviews	145 medications
Summary of Alvesco reviews	7.7	20 reviews	

Reviews may be moderated or edited before publication to correct grammar and spelling or to remove inappropriate language and content. Reviews that appear to be created by parties with a vested interest in the medication will not be published. As reviews and ratings are subjective and self-reported, this information should not be used as the basis for any statistical analysis or scientific studies.

[Share your Experience](#)

[Ask a Question](#)

Reviews for Alvesco

Most Recent 


Shellyscorner · Taken for 5 to 10 years

January 25, 2020

For Asthma, Maintenance "I've been on Alvesco longer than any of the many other steroid inhalers that I've used. In all the YEARS that I've been using it, I've never developed a thrush infection. ALL the others that I've ever used I would develop thrush fairly regularly. Even at the dose of 2 puffs twice daily, I've only developed thrush once with Alvesco. And as a maintenance med, it's worked great for me!"

9.0 

What this helpful? [Yes](#) [No](#)

 1 · [Report](#)


MVE · Taken for 1 to 2 years

January 12, 2020

For Asthma, Maintenance "I've been using a different steroid inhaler for several years. My doctor changed my prescription to Alvesco. Even after the first use I noticed a difference in my asthma. I've had zero side effects and I've been using it for approximately a year."

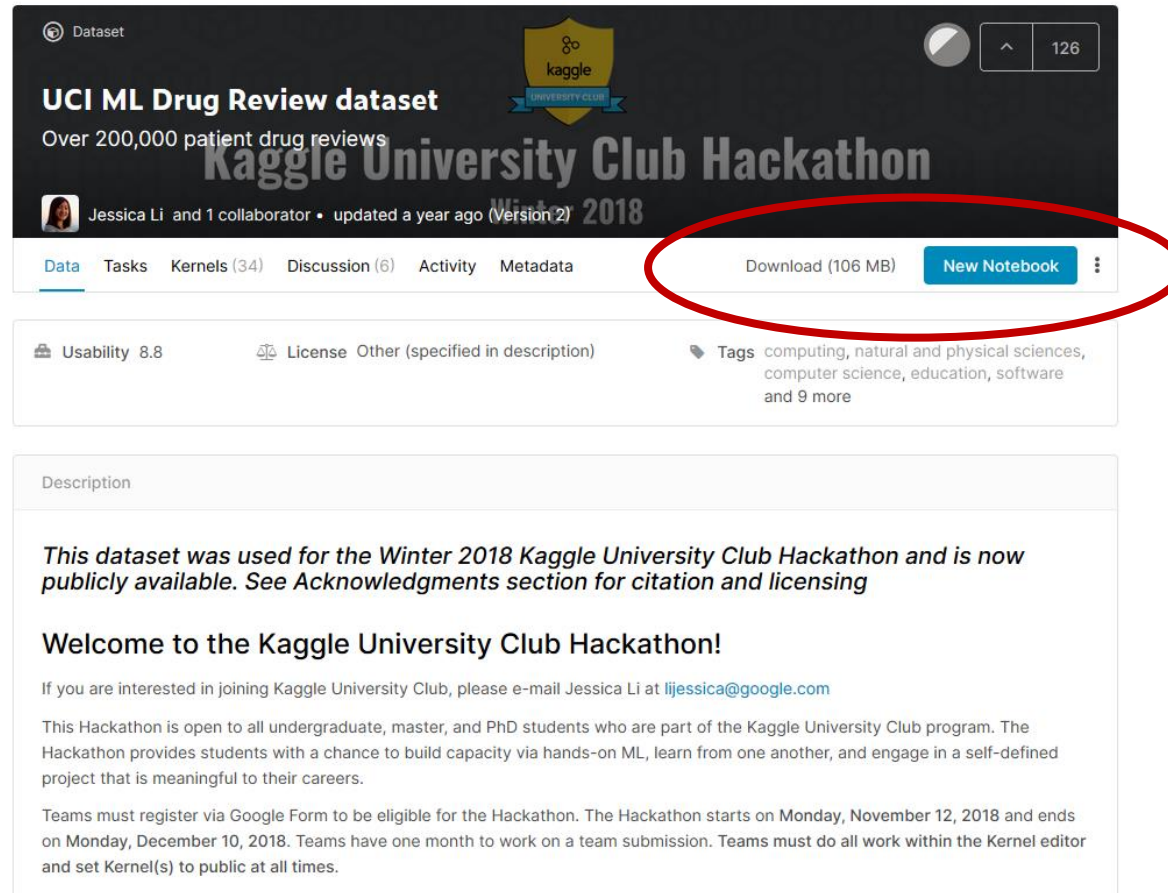
10 

What this helpful? [Yes](#) [No](#)

 1 · [Report](#)

<https://www.drugs.com/comments/ciclesonide/alvesco.html>

Dataset – Kaggle Winter 2018 hackaton



The screenshot shows the Kaggle dataset page for 'UCI ML Drug Review dataset'. The page has a dark header with the Kaggle logo and a '126' badge. The main title is 'UCI ML Drug Review dataset' with a subtitle 'Over 200,000 patient drug reviews'. Below this, it says 'Kaggle University Club Hackathon Winter 2018' and 'Jessica Li and 1 collaborator • updated a year ago (Version 2)'. A red circle highlights the 'Download (106 MB)' and 'New Notebook' buttons. Below the header, there are tabs for 'Data', 'Tasks', 'Kernels (34)', 'Discussion (6)', 'Activity', and 'Metadata'. The 'Data' tab is selected. Below the tabs, there are sections for 'Usability 8.8', 'License Other (specified in description)', and 'Tags computing, natural and physical sciences, computer science, education, software and 9 more'. The 'Description' section contains the following text:

This dataset was used for the Winter 2018 Kaggle University Club Hackathon and is now publicly available. See Acknowledgments section for citation and licensing

Welcome to the Kaggle University Club Hackathon!

If you are interested in joining Kaggle University Club, please e-mail Jessica Li at lijessica@google.com

This Hackathon is open to all undergraduate, master, and PhD students who are part of the Kaggle University Club program. The Hackathon provides students with a chance to build capacity via hands-on ML, learn from one another, and engage in a self-defined project that is meaningful to their careers.

Teams must register via Google Form to be eligible for the Hackathon. The Hackathon starts on Monday, November 12, 2018 and ends on Monday, December 10, 2018. Teams have one month to work on a team submission. Teams must do all work within the Kernel editor and set Kernel(s) to public at all times.

<https://www.kaggle.com/jessicali9530/kuc-hackathon-winter-2018>

Explorative data analysis

What can we learn from the data?

Example - Team NDL: Algorithms and illnesses



<https://www.kaggle.com/neilash/team-ndl-algorithms-and-illnesses>

Drug Ratings Dataset: Preliminary Data Exploration

Our ideas for preliminary exploration:

- Most common conditions
- Overall best and worst reviewed drugs
- The curability of each disease
- Best drugs for each condition
- Most useful reviews
- Usefulness vs review score
- Bias in reviews
 - Users tend to review things they really liked or really disliked, fewer reviews in the middle

Importing libraries

In [1]:

```
# ALL imports
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import style; style.use('ggplot')
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import time
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
```

Reading the datasets

```
In [2]: # Create dataframes train and test
train = pd.read_csv('../input/drugsComTrain_raw.csv')
test = pd.read_csv('../input/drugsComTest_raw.csv')
```

```
In [3]: train.head()
```

Out[3]:

	uniqueID	drugName	condition	review	rating	date	usefulCount
0	206461	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9	20-May-12	27
1	95260	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8	27-Apr-10	192
2	92703	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5	14-Dec-09	17
3	138000	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8	3-Nov-15	10
4	35696	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9	27-Nov-16	37

Check the column names and dataset sizes

In [6]:

```
list(train)
```

Out[6]:

```
['uniqueID',  
 'drugName',  
 'condition',  
 'review',  
 'rating',  
 'date',  
 'usefulCount']
```

In [7]:

```
train.values.shape[0], test.values.shape[0], train.values.shape[0] / test.values.shape[0]
```

Out[7]:

```
(161297, 53766, 2.999981400885318)
```

Yep, the train set is almost exactly 3 times as big as the test set! This is a typical 75:25 train:test split.

What are the most common (medical) conditions?

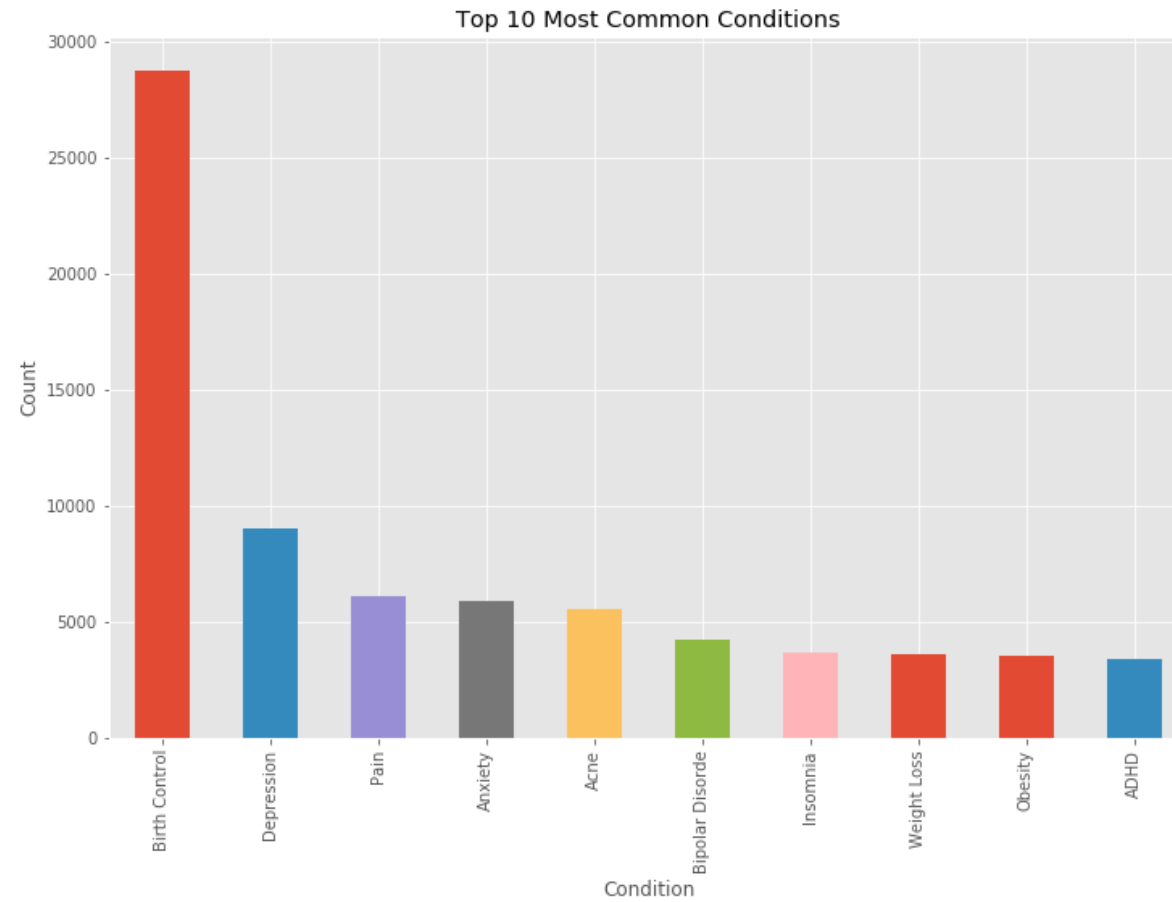
Common Conditions

```
In [10]: # I previously did this by creating and sorting a dictionary -- here's an easier way with pandas!  
(Inspiration from Sayan Goswami)  
conditions = train.condition.value_counts().sort_values(ascending=False)  
conditions[:10]
```

```
Out[10]:  
Birth Control      28788  
Depression         9069  
Pain               6145  
Anxiety           5904  
Acne              5588  
Bipolar Disorde   4224  
Insomnia          3673  
Weight Loss       3609  
Obesity           3568  
ADHD              3383  
Name: condition, dtype: int64
```

```
In [11]: plt.rcParams['figure.figsize'] = [12, 8]
```

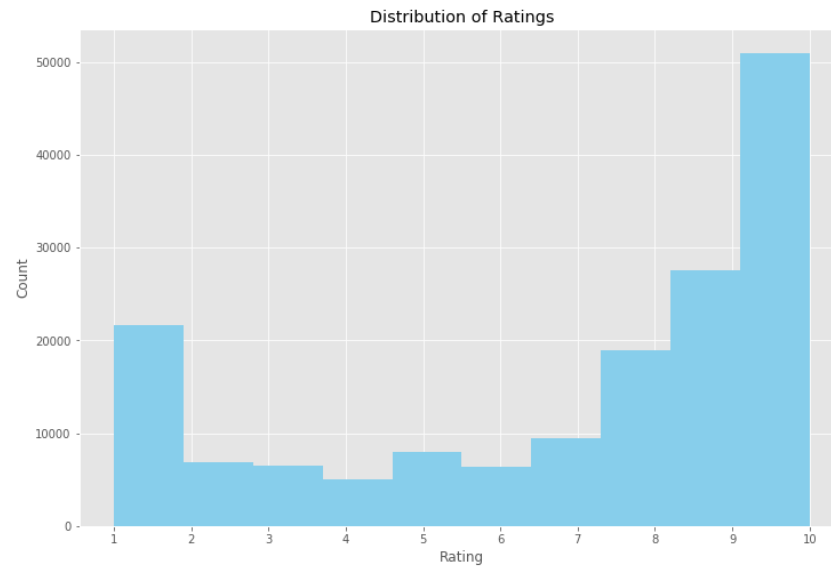
```
In [12]: conditions[:10].plot(kind='bar')  
plt.title('Top 10 Most Common Conditions')  
plt.xlabel('Condition')  
plt.ylabel('Count');
```



What is the rating distribution?

Rating Distribution

```
In [13]: # Look at bias in review (also shown on 'Data' page in competition: distribution of ratings)
train.rating.hist(color='skyblue')
plt.title('Distribution of Ratings')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks([i for i in range(1, 11)]);
```

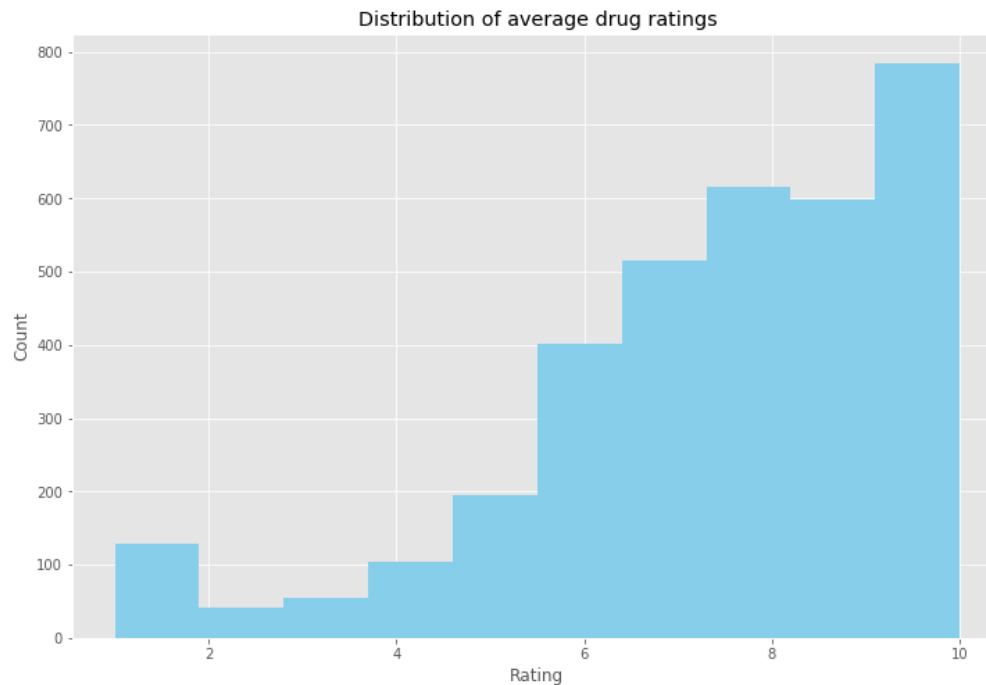


This distribution illustrates that people generally write reviews for drugs they really like (or those that they really dislike). There are fewer middle ratings as compared to extreme ratings.

What is the average drug rating?

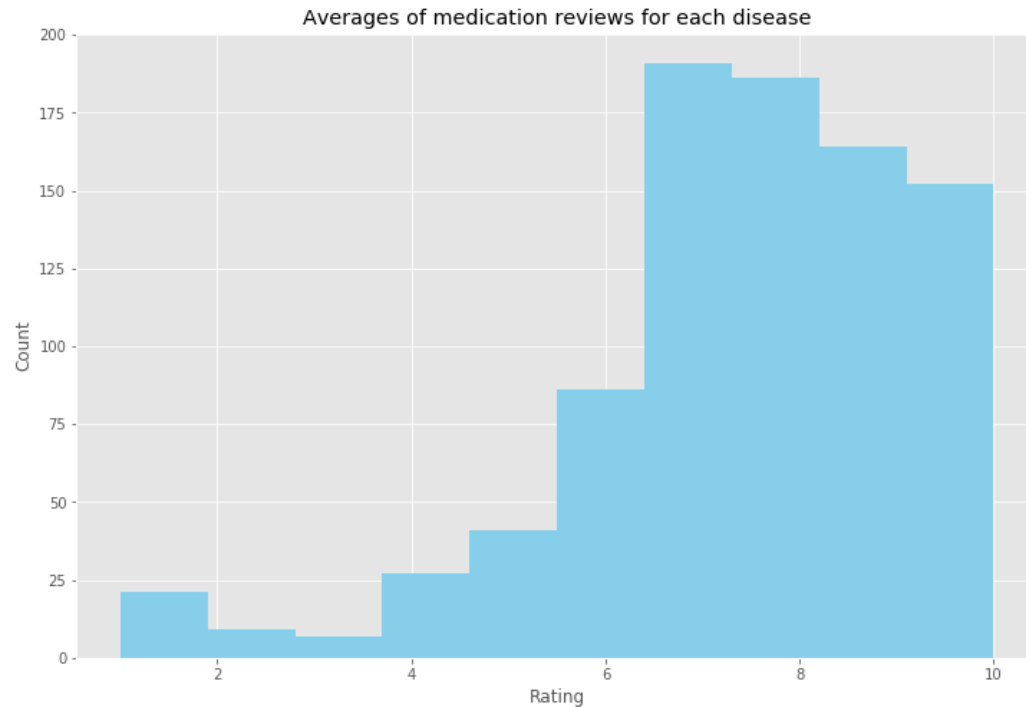
```
In [14]: rating_avgs = (train['rating'].groupby(train['drugName']).mean())  
rating_avgs.hist(color='skyblue')  
plt.title('Distribution of average drug ratings')  
plt.xlabel('Rating')  
plt.ylabel('Count')
```

```
Out[14]: Text(0,0.5, 'Count')
```



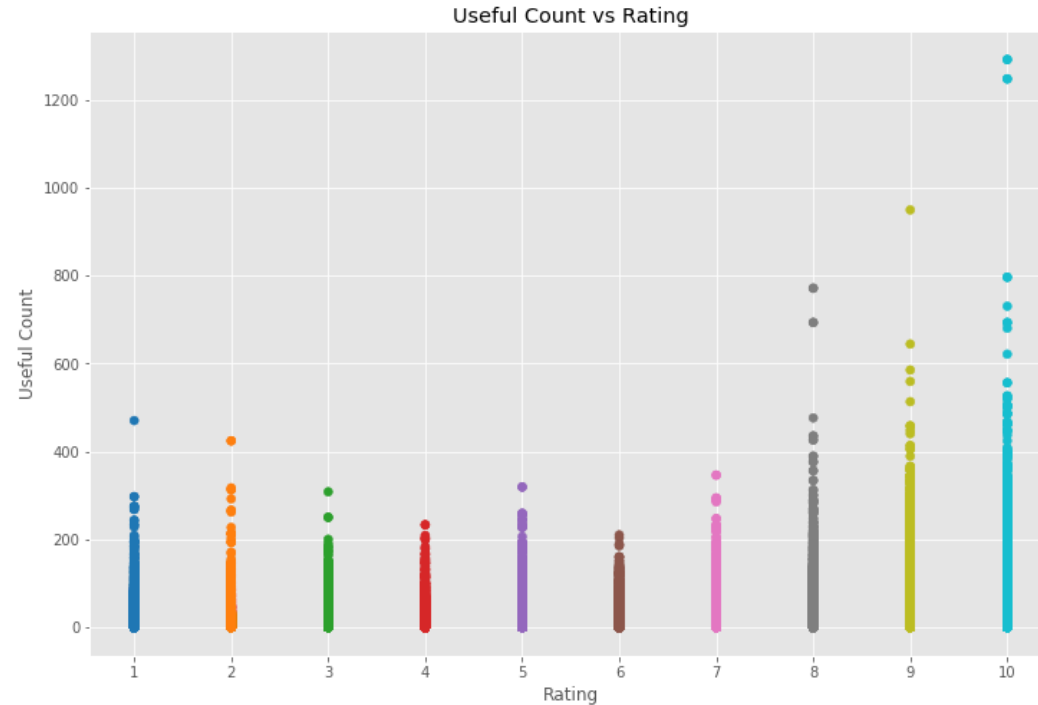
What is the average rating for condition?

```
In [15]: rating_avgs = (train['rating'].groupby(train['condition']).mean())
rating_avgs.hist(color='skyblue')
plt.title('Averages of medication reviews for each disease')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.show()
```

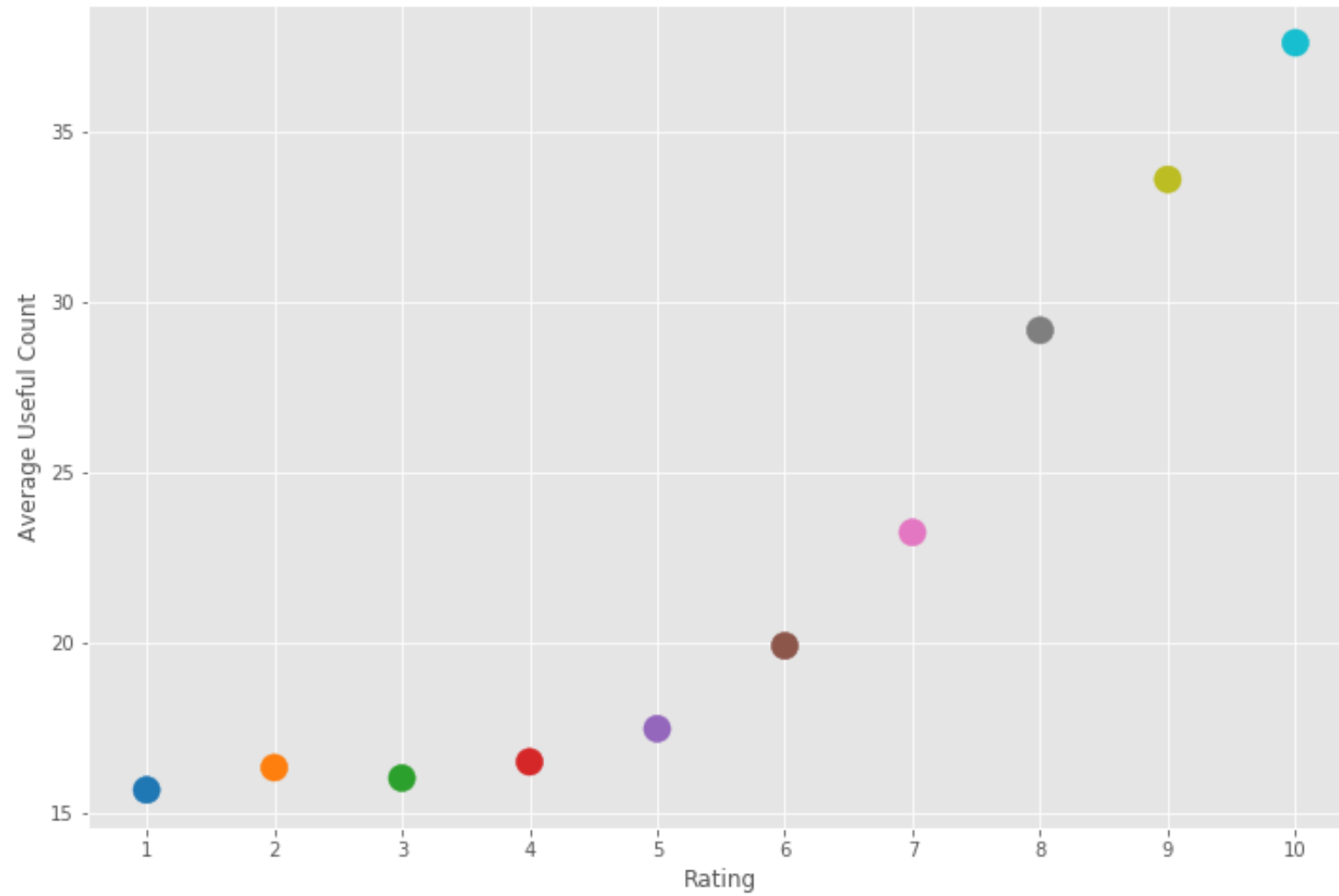


Is rating correlated with the usefulness of the review?

```
In [16]: # Is rating correlated with usefulness of the review?
plt.scatter(train.rating, train.usefulCount, c=train.rating.values, cmap='tab10')
plt.title('Useful Count vs Rating')
plt.xlabel('Rating')
plt.ylabel('Useful Count')
plt.xticks([i for i in range(1, 11)]);
```



Average Useful Count vs Rating



What makes a review useful? (most useful reviews)

```
In [19]: # Sort train dataframe from most to least useful
useful_train = train.sort_values(by='usefulCount', ascending=False)
useful_train.iloc[:10]
```

Out[19]:

	uniqueID	drugName	condition	review	rating	date	usefulCount
6716	96616	Sertraline	Depression	"I remember reading people's opinions, on...	10	31-Jul-08	1291
33552	119152	Zoloft	Depression	"I remember reading people's opinions, on...	10	31-Jul-08	1291
21708	131116	Levonorgestrel	Birth Control	"I have had my IUD for over a year now and I t...	10	1-Apr-09	1247
4249	182560	Mirena	Birth Control	"I have had my IUD for over a year now and I t...	10	1-Apr-09	1247
146145	119151	Zoloft	Depression	"I've been on Zoloft 50mg for over two ye...	9	5-Aug-08	949
58608	139141	Phentermine	Weight Loss	"I have used this pill off and on for the past...	10	19-Oct-08	796
16889	52305	Adipex-P	Weight Loss	"I have used this pill off and on for the past...	10	19-Oct-08	796
2039	62757	Citalopram	Depression	"I responded after one week. The side effects ...	8	25-Mar-08	771
152838	89825	Celexa	Depression	"I responded after one week. The side effects ...	8	25-Mar-08	771
5218	107655	Implanon	Birth Control	"I was very nervous about trying Implanon afte...	10	19-Jul-10	730

In [20]:

```
# Print top 10 most useful reviews
for i in useful_train.review.iloc[:3]:
    print(i, '\n')
```

"I remember reading people's opinions, online, of the drug before I took it and it scared me away from it. Then I finally decided to give it a try and it has been the best choice I have made. I have been on it for over 4 months and I feel great. I'm on 100mg and I don't have any side effects. When I first started I did notice that my hands would tremble but then it subsided. So honestly, don't listen to all the negativity because what doesn't work for some works amazing for others. So go based on yourself and not everyone else. It may be a blessing in disguise. The pill is not meant to make you be all happy go lucky and see "butterflies and roses", its meant to help put the chemicals in your mind in balance so you can just be who you are and not overly depressed. I still get sad sometimes, but that is normal, that is life, and it's up to people to take control to make a change. I did so by getting on this pill."

"I remember reading people's opinions, online, of the drug before I took it and it scared me away from it. Then I finally decided to give it a try and it has been the best choice I have made. I have been on it for over 4 months and I feel great. I'm on 100mg and I d

In [21]:

```
# Print 10 of the least useful reviews
for i in useful_train.review.iloc[-3:]:
    print(i, '\n')
```

```
"I started yesterday and today I see it darker. Should I stop? I have a wedding in 10 days... will my melasma be better by then or still this dark? Thank you"
```

The not-so-useful reviews seem much more negative. The final review listed is barely a review -- just a concerned patient asking questions about the product!

Our conclusions appear consistent with the above graph -- reviewers find higher ratings/better reviews to be more useful than lower ratings/worse reviews. Does this represent some sort of bias within the useful count?

We're also interested in quantifying the sentiment of these reviews.

Sentiment analysis (opinion or emotion analysis)

```
In [22]: sid = SentimentIntensityAnalyzer()
```

```
In [23]: # Create list (cast to array) of compound polarity sentiment scores for reviews
sentiments = []

for i in train.review:
    sentiments.append(sid.polarity_scores(i).get('compound'))


sentiments = np.asarray(sentiments)
```

```
In [24]: sentiments
```

```
Out[24]: array([-0.296 ,  0.8603,  0.7645, ..., -0.743 ,  0.6197,  0.6124])
```


Example – IBM Tone Analyzer

IBM Watson Developer Cloud



Tone Analyzer

This service uses linguistic analysis to detect joy, fear, sadness, anger, analytical, confident and tentative tones found in text.

*This system is for demonstration purposes only and is not intended to process Personal Data. No Personal Data is to be entered into this system as it may not have the necessary controls in place to meet the requirements of the General Data Protection Regulation (EU) 2016/679.

By using this application, you agree to the [Terms of Use](#)

Resources:

[Documentation](#)

[API Reference](#)

[Fork on Github](#)

Start for free in IBM Cloud

Sample use cases

Choose an example to learn how you can adjust the tone of your content to change people's perceptions, or improve its effectiveness.

[Learn more](#).

☒ Tweets ☐ Online Review ☐ Email message ☐ Product Review in French ☐ Your own text

Analyzing Customer Engagement Data? Try out the [Tone Analyzer Customer Engagement Endpoint](#).

I hate these new features On #ThisPhone after the update.
I hate #ThisPhoneCompany products, you'd have to torture me to get me to use #ThisPhone.
The emojis in #ThisPhone are stupid.
#ThisPhone is a useless, stupid waste of money.
#ThisPhone is the worst phone I've ever had - ever 😡.
#ThisPhone another ripoff, lost all respect SHAME.
I'm worried my #ThisPhone is going to overheat like my brother's did.
#ThisPhoneCompany really let me down - my new phone won't even turn on

Analyze

<https://tone-analyzer-demo.ng.bluemix.net/>

Sentence-level

Identify sentences with stronger tones in context or sorted by score. Highlighted sentences indicate the likelihood of a tone present. If more than one tone is present, the stronger one is shown. Click on a sentence to see a breakdown of all tones.

Tones

Analytical

Anger

Confident

Fear

Tentative

In context

Anger: Evoked due to injustice, conflict, humiliation, negligence or betrayal. If anger is active, the individual attacks the target, verbally or physically. If anger is passive, the person silently sulks and feels tension and hostility.

< .5

.5 - .75

> .75

NoneStrong

I hate these new features On #ThisPhone after the update.

I hate #ThisPhoneCompany products, you'd have to torture me to get me to use #ThisPhone.

The emojis in #ThisPhone are stupid.

#ThisPhone is a useless, stupid waste of money.

#ThisPhone is the worst phone I've ever had - ever 🤬.

#ThisPhone another ripoff, lost all respect SHAME.

I'm worried my #ThisPhone is going to overheat like my brother's did.

Sentiment Analysis with Python NLTK Text Classification

This is a demonstration of **sentiment analysis** using a **NLTK 2.0.4** powered **text classification** process. It can tell you whether it thinks the text you enter below expresses **positive sentiment**, **negative sentiment**, or if it's **neutral**. Using **hierarchical classification**, *neutrality* is determined first, and *sentiment polarity* is determined second, but only if the text is not neutral.

Analyze Sentiment

Language
english ▾

Enter text
great movie

Enter up to 50000 characters

Analyze

Sentiment Analysis Results

The text is **pos**.

The final sentiment is determined by looking at the classification probabilities below.

Subjectivity

- neutral: 0.1
- polar: 0.9

Polarity

- **pos: 0.7**
- neg: 0.3

<https://text-processing.com/demo/sentiment/>

Natural Language Toolkit ¶

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to [over 50 corpora and lexical resources](#) such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active [discussion forum](#).

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

[Natural Language Processing with Python](#) provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been updated for Python 3 and NLTK 3. (The original Python 2 version is still available at http://nltk.org/book_1ed.)

<https://www.nltk.org/>

Add sentiment analysis results to dataset

```
In [25]: useful_train['sentiment'] = pd.Series(data=sentiments)
```

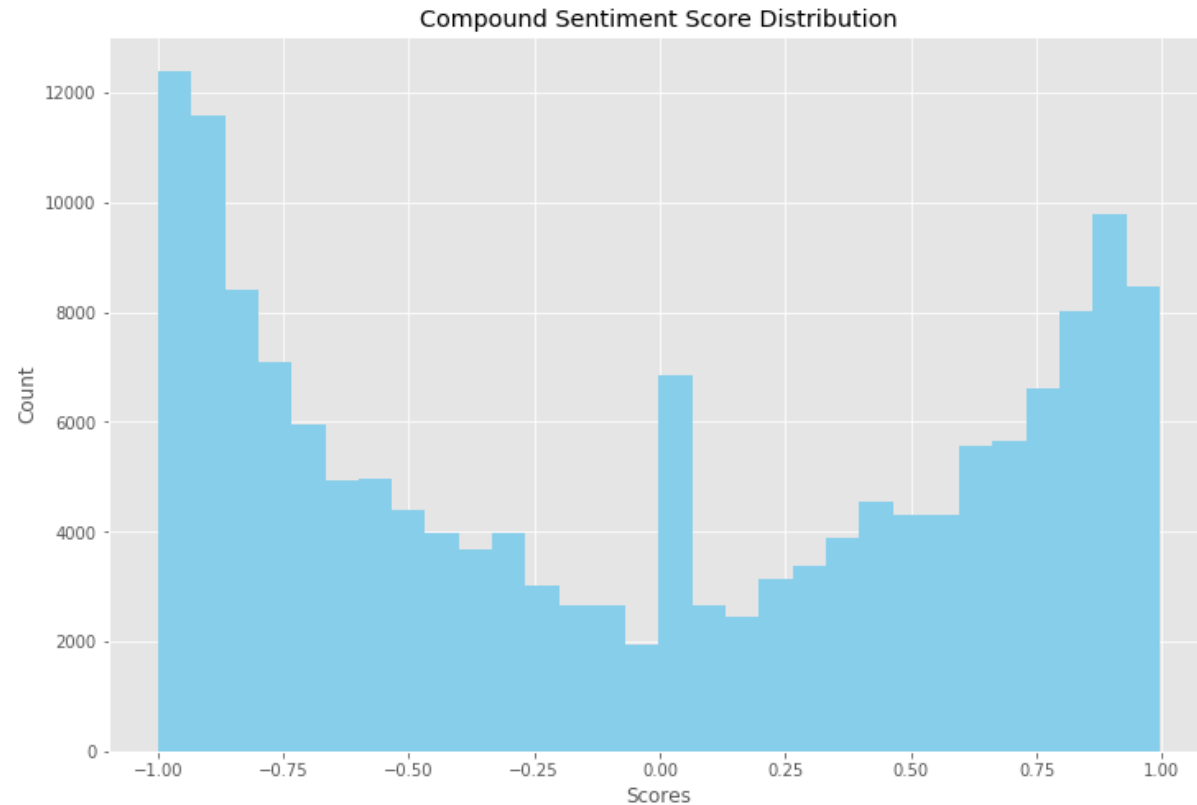
```
In [26]: useful_train = useful_train.reset_index(drop=True)
useful_train.head()
```

Out[26]:

	uniqueID	drugName	condition	review	rating	date	usefulCount	sentiment
0	96616	Sertraline	Depression	"I remember reading people's opinions, on...	10	31-Jul-08	1291	0.9772
1	119152	Zoloft	Depression	"I remember reading people's opinions, on...	10	31-Jul-08	1291	0.9772
2	131116	Levonorgestrel	Birth Control	"I have had my IUD for over a year now and I t...	10	1-Apr-09	1247	0.7739
3	182560	Mirena	Birth Control	"I have had my IUD for over a year now and I t...	10	1-Apr-09	1247	0.7739
4	119151	Zoloft	Depression	"I've been on Zoloft 50mg for over two ye...	9	5-Aug-08	949	-0.6815

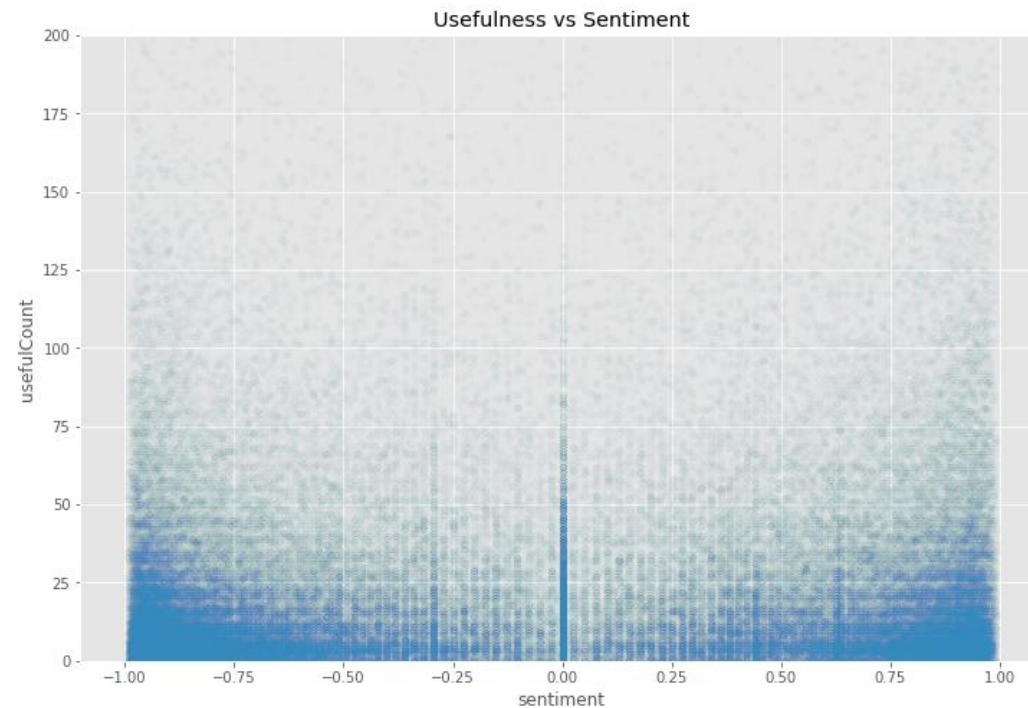
How the sentiment scores are distributed?

```
In [27]: useful_train.sentiment.hist(color='skyblue', bins=30)
plt.title('Compound Sentiment Score Distribution')
plt.xlabel('Scores')
plt.ylabel('Count');
```

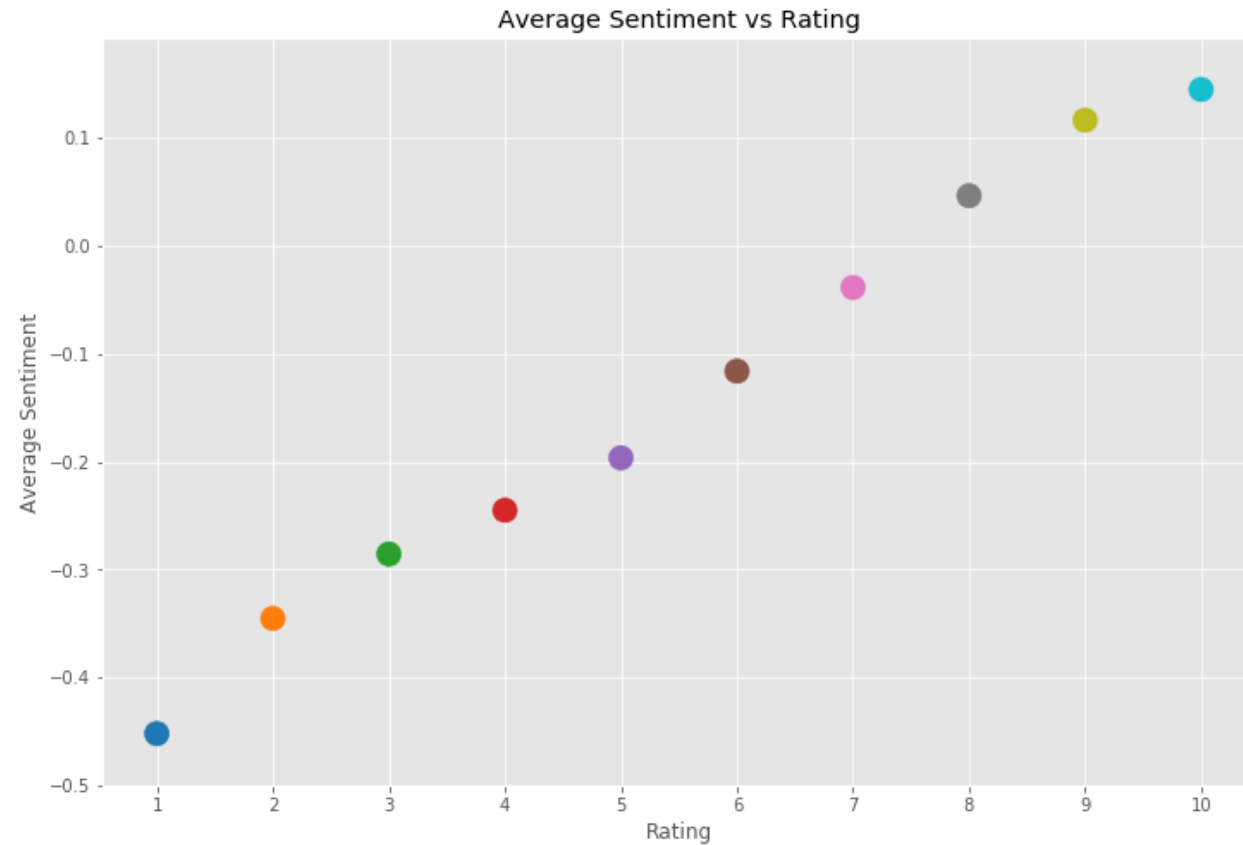


How the sentiment score and usefulness are correlated?

```
In [28]: useful_train.plot(x='sentiment', y='usefulCount', kind='scatter', alpha=0.01)
plt.title('Usefulness vs Sentiment')
plt.ylim(0, 200);
```



How does the average sentiment score correlate with rating?



Highest and lowest rated drugs

	0	1
0	Zutripro	10.000000
1	Chlorpheniramine / hydrocodone / pseudoephedrine	10.000000
2	Silver sulfadiazine	9.972222
3	Drixoral Cold and Allergy	9.948718
4	Dexbrompheniramine / pseudoephedrine	9.947368
5	Emend	9.900000
6	Aprepitant	9.900000
7	Tegaserod	9.812500
8	Zelnorm	9.687500
9	Cyanocobalamin	9.666667

	0	1
1371	Pprevnar 13	3.363636
1372	Fosamax	3.166667
1373	Blisovi 24 Fe	3.088889
1374	Opdivo	3.083333
1375	Miconazole	3.033000
1376	Monistat 7	3.032258
1377	Alendronate	2.954545
1378	Yuvaferm	2.318182
1379	Monistat 1-Day or Night Combination Pack	1.416667
1380	ProAir RespiClick	1.193548

Tensorflow classifier example

How can we predict the output based on text input and using neural networks?

Tensorflow classifier

- Handling text with tensorflow
- Categorizing the labels
- Split into training and validation sets
- One-hot-coding the labels
- Standard dense NN model
- Training
- Results
- Next steps

Handling text with tensorflow

```
In [5]: 1 from tensorflow.keras.preprocessing.text import Tokenizer
        2
        3 # Note!| the data is cut to 15,000 samples for demonstration purposes
        4 samples = train['review'].iloc[:15000]
        5 tokenizer = Tokenizer(num_words = 5000)
        6 tokenizer.fit_on_texts(samples)
        7
        8 word_index = tokenizer.word_index
        9 print('Found %s unique tokens.' % len(word_index))
```

Found 20153 unique tokens.

```
In [6]: 1 # Make one hot samples
        2 data = tokenizer.texts_to_matrix(samples, mode='binary')
        3 data.shape
```

Out[6]: (15000, 5000)

Categorize labels

```
In [21]: 1 # Create 3 categories
          2 # label = 2, when rating >= 8
          3 # label = 1, when rating >= 5
          4 # label = 0, otherwise
          5 labels = train['rating'].iloc[:15000].values
          6 labels = 1.0*(labels >= 8) + 1.0*(labels >= 5)
          7 labels[:10]
```

```
Out[21]: array([2., 2., 1., 2., 2., 0., 0., 2., 0., 2.])
```

Split into training and validation sets

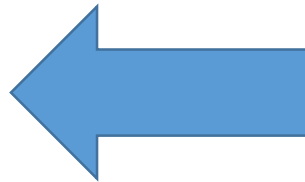
```
In [24]: 1 # Split into training and validation sets
          2 from sklearn.model_selection import train_test_split
          3 x_train, x_val, y_train, y_val = train_test_split(data, labels, test_size = 0.333, random_state = 2019)
```

Or you could use `validation_split = 0.333` when training the model. Your choice.

One-hot-code the output values

```
In [26]: 1 from tensorflow.keras.utils import to_categorical
          2
          3 y_train_cat = to_categorical(y_train)
          4 y_val_cat = to_categorical(y_val)
          5
          6 y_train_cat[:20]
```

```
Out[26]: array([[0., 0., 1.],
                [1., 0., 0.],
                [1., 0., 0.],
                [0., 0., 1.],
                [0., 0., 1.],
                [0., 0., 1.],
                [0., 0., 1.],
                [0., 0., 1.],
                [0., 0., 1.],
                [0., 0., 1.],
                [0., 0., 1.],
                [0., 0., 1.],
                [0., 0., 1.],
                [0., 0., 1.],
                [1., 0., 0.],
                [1., 0., 0.],
                [1., 0., 0.],
                [0., 1., 0.],
                [0., 0., 1.],
                [1., 0., 0.],
                [0., 1., 0.],
                [1., 0., 0.]], dtype=float32)
```



```
7 labels[:10]
```

```
Out[21]: array([2., 2., 1., 2., 2., 0., 0., 2., 0., 2.])
```

To_categorical converts the numerical labels into one-hot-coded vectors:

0 → [0., 0., 1.]

1 → [0., 1., 0.]

2 → [1., 0., 0.]

Standard dense neural network model

```
1  # Create a simple sequential model
2  from tensorflow.keras.models import Sequential
3  from tensorflow.keras.layers import Dense, Activation
4
5  model = Sequential()
6  model.add(Dense(256, input_dim = 5000))
7  model.add(Activation('relu'))
8  model.add(Dense(3))
9  model.add(Activation('softmax'))
10 model.compile(optimizer = 'adam',
11               loss = 'categorical_crossentropy',
12               metrics = [tfa.metrics.CohenKappa(num_classes=3)])
13 model.summary()
```

Notice! We have 3 dense layers at the bottom and 'softmax' activation as we have 3 categories. The metrics is changed to Cohen's Kappa. More about that later on You can first try accuracy for metrics.

Training the model

```
1 %%time
2 history = model.fit(x_train, y_train_cat,
3                     epochs = 10,
4                     batch_size = 32,
5                     verbose = 1,
6                     validation_data = (x_val, y_val_cat))
```

Train on 10005 samples, validate on 4995 samples

```
Epoch 10/10
10005/10005 [=====] - 4s 425us/sample - loss:
6 - val_cohen_kappa: 0.4272
Wall time: 48.5 s
```

%%time - counts how much time has elapsed during processing the cell.

- -Works on laptop and Google Colab, but how about in Kaggle?

This demo is using only a subset (10,005 samples) of the original data to test the code.

You should use all data in your final experiments.

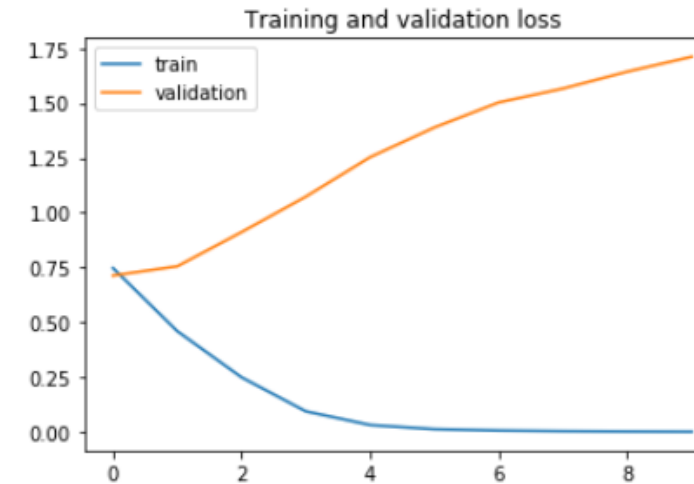
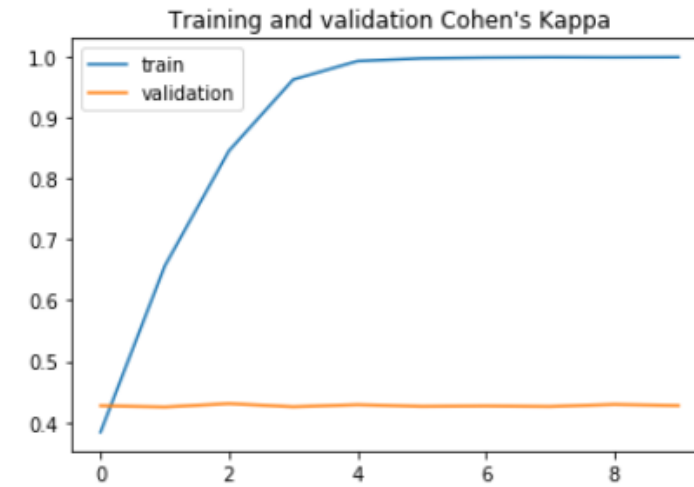
Results

```
: 1 # Plot the accuracy and loss
2 acc = history.history['cohen_kappa']
3 val_acc = history.history['val_cohen_kappa']
4 loss = history.history['loss']
5 val_loss = history.history['val_loss']
6
7 plt.plot(acc, label = 'train')
8 plt.plot(val_acc, label = 'validation')
9 plt.title("Training and validation Cohen's Kappa")
10 plt.legend()
11
12 plt.figure()
13
14 plt.plot(loss, label = 'train')
15 plt.plot(val_loss, label = 'validation')
16 plt.title('Training and validation loss')
17 plt.legend()
18
19 plt.show()
```

Cohen's kappa:

1.0 = perfect match,

-1.0 total disagreement



Next steps

- Use whole training dataset
- Use all rating categories
 - from 1 to 10
- Try
 - recurrent neural networks (RNN)
 - long-short-term-memory networks (LSTM)
- Train with one condition, validate with other condition
 - See [the Reference article, Table 1 \(page 3\)](#)

Next week ...

tfa.metrics.CohenKappa

Computes Kappa score between two raters.

Might need additional installation (Anaconda prompt):

- pip install tensorflow_addons

Remember to add import libraries

- import tensorflow_addons as tfa

```
8 model.add(Dense(3))
9 model.add(Activation('softmax'))
10 model.compile(optimizer = 'adam',
11               loss = 'categorical_crossentropy',
12               metrics = [tfa.metrics.CohenKappa(num_classes=3)])
13 model.summary()
```

Cohen's Kappa

Cohen's kappa coefficient (κ) is a statistic that is used to measure inter-rater reliability (and also Intra-rater reliability) for qualitative (categorical) items.

It is generally thought to be a more robust measure than simple percent agreement calculation, as κ takes into account the possibility of the agreement occurring by chance.

https://en.wikipedia.org/wiki/Cohen%27s_kappa

Interpretation

The score lies in the range $[-1, 1]$.

- a score of 1 represents complete agreement between the two raters.
 - over 0.75 as excellent,
 - 0.40 to 0.75 as fair to good, and
 - below 0.40 as poor
- a score of 0 means agreement by chance.
- a score of -1 represents complete disagreement between two raters.

https://www.tensorflow.org/addons/api_docs/python/tfa/metrics/CohenKappa

Example calculation

		B	
		Yes	No
A	Yes	20	5
	No	10	15

The observed proportionate agreement is:

$$p_o = \frac{a + d}{a + b + c + d} = \frac{20 + 15}{50} = 0.7$$

To calculate p_e (the probability of random agreement) we note that:

- Reader A said "Yes" to 25 applicants and "No" to 25 applicants. Thus reader A said "Yes" 50% of the time.
- Reader B said "Yes" to 30 applicants and "No" to 20 applicants. Thus reader B said "Yes" 60% of the time.

So the expected probability that both would say yes at random is:

$$p_{\text{Yes}} = \frac{a + b}{a + b + c + d} \cdot \frac{a + c}{a + b + c + d} = 0.5 \times 0.6 = 0.3$$

Similarly:

$$p_{\text{No}} = \frac{c + d}{a + b + c + d} \cdot \frac{b + d}{a + b + c + d} = 0.5 \times 0.4 = 0.2$$

Overall random agreement probability is the probability that they agreed on either Yes or No, i.e.:

$$p_e = p_{\text{Yes}} + p_{\text{No}} = 0.3 + 0.2 = 0.5$$

So now applying our formula for Cohen's Kappa we get:

$$\kappa = \frac{p_o - p_e}{1 - p_e} = \frac{0.7 - 0.5}{1 - 0.5} = 0.4$$

		B	
		Yes	No
A	Yes	a	b
	No	c	d

Suppose that you were analyzing data related to a group of 50 people applying for a grant. Each grant proposal was read by two readers and each reader either said "Yes" or "No" to the proposal. Suppose the disagreement count data were as follows, where A and B are readers, data on the main diagonal of the matrix (a and d) count the number of agreements and off-diagonal data (b and c) count the number of disagreements:

See Wikipedia, Cohen's Kappa, [Simple Example](#)