```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from collections import defaultdict
import json
import gzip
from urllib.request import urlopen
from collections import defaultdict
```

Read Data

```
!wget http://deepyeti.ucsd.edu/jianmo/amazon/categoryFilesSmall/Movies_and_TV_5.json.g
```

```
--2022-03-12 00:53:21--  http://deepyeti.ucsd.edu/jianmo/amazon/categoryFilesSma
Resolving deepyeti.ucsd.edu (deepyeti.ucsd.edu)... 169.228.63.50
Connecting to deepyeti.ucsd.edu (deepyeti.ucsd.edu)|169.228.63.50|:80... connecte
HTTP request sent, awaiting response... 200 OK
Length: 791322468 (755M) [application/octet-stream]
Saving to: 'Movies_and_TV_5.json.gz'

Movies_and_TV_5.jso 100%[===================>] 754.66M  73.4MB/s    in 11s

2022-03-12 00:53:33 (65.9 MB/s) - 'Movies_and_TV_5.json.gz' saved [791322468/791
```

```
### load the meta data

data = []
with gzip.open('Movies_and_TV_5.json.gz') as f:
    for l in f:
        j = json.loads(l.strip())
        d = {}
        d['overall'] = j['overall']
        if 'reviewText' in j:
            d['reviewText'] = j['reviewText']
        else:
            d['reviewText'] = None
        if 'summary' in j:
            d['summary'] = j['summary']
        else:
            d['summary'] = None
        data.append(d)

# total length of list, this number equals total number of products
print(len(data))
```

```
# first row of the list
print(data[0])
```

```
3410019
{'overall': 5.0, 'reviewText': "So sorry I didn't purchase this years ago when i
```

Clean Data

```
df = pd.DataFrame(data)


# Drop data with no review text
df = df[~df.reviewText.isnull()].copy()


# Create dataframes for each star rating
df_1 = df[df['overall'] == 1]
print('num 1-star reviews: ', len(df_1))

df_2 = df[df['overall'] == 2]
print('num 2-star reviews: ', len(df_2))

df_3 = df[df['overall'] == 3]
print('num 3-star reviews: ', len(df_3))

df_4 = df[df['overall'] == 4]
print('num 4-star reviews: ', len(df_4))

df_5 = df[df['overall'] == 5]
print('num 5-star reviews: ', len(df_5))
```

```
num 1-star reviews:  193110
num 2-star reviews:  172409
num 3-star reviews:  349641
num 4-star reviews:  665734
num 5-star reviews:  2027544
```

```
import scipy as sp
import seaborn as sns
#MLE
print("The resulting theta will be the one that maximizes the likelihood function -> t

#1star
n1 = len(df_1)
theta = len(df_1)/len(df)
X_arr = np.random.choice([0, 1], p=[1-theta, theta], size=len(df))
resulting_theta1 = sum(X_arr)/len(df)

print("The resulting theta will be the one that maximizes the likelihood function for
```

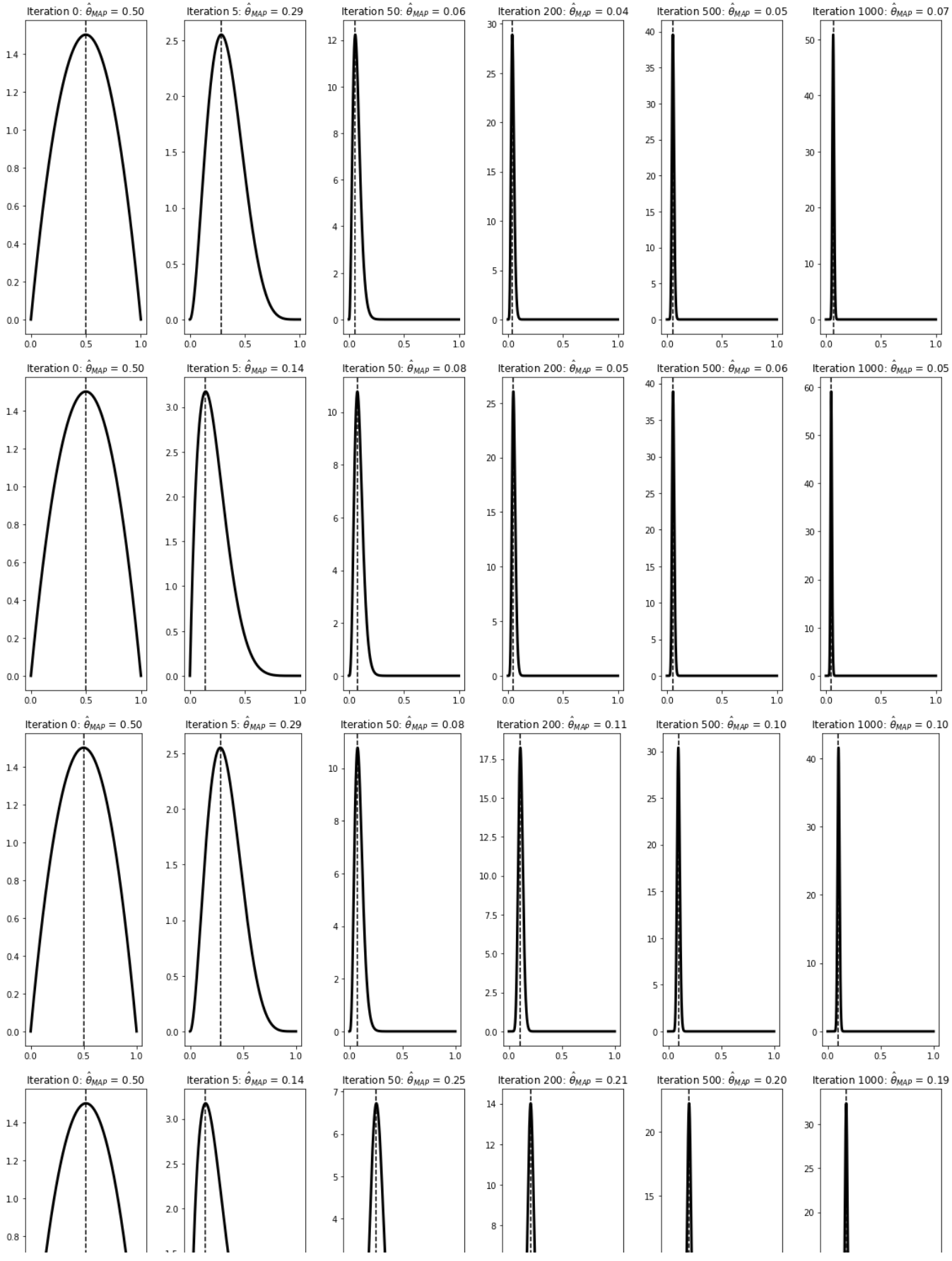
```
#2star
n2 = len(df_2)
theta = len(df_2)/len(df)
X_arr = np.random.choice([0, 1], p=[1-theta, theta], size=len(df))
resulting_theta2 = sum(X_arr)/len(df)

print("The resulting theta will be the one that maximizes the likelihood function for


#3star
n3 = len(df_3)
theta = len(df_3)/len(df)
X_arr = np.random.choice([0, 1], p=[1-theta, theta], size=len(df))
resulting_theta3 = sum(X_arr)/len(df)

print("The resulting theta will be the one that maximizes the likelihood function for

#4star
n4 = len(df_4)
theta = len(df_4)/len(df)
X_arr = np.random.choice([0, 1], p=[1-theta, theta], size=len(df))
resulting_theta4 = sum(X_arr)/len(df)

print("The resulting theta will be the one that maximizes the likelihood function for


#5star
n5 = len(df_5)
theta = len(df_5)/len(df)
X_arr = np.random.choice([0, 1], p=[1-theta, theta], size=len(df))
resulting_theta5 = sum(X_arr)/len(df)

print("The resulting theta will be the one that maximizes the likelihood function for


    The resulting theta will be the one that maximizes the likelihood function -> the
    The resulting theta will be the one that maximizes the likelihood function for 1
    The resulting theta will be the one that maximizes the likelihood function for 2
    The resulting theta will be the one that maximizes the likelihood function for 3
    The resulting theta will be the one that maximizes the likelihood function for 4
    The resulting theta will be the one that maximizes the likelihood function for 5


#MAP
n = 10000
X_arr = np.ones(n)
alpha = 2
beta = 2

#1 star
theta = len(df_1)/len(df)
```

```
X_arr = np.random.choice([0, 1], p=[1-theta, theta], size=n)
print(sum(X_arr) / n)

new_beta = np.asarray([[alpha + sum(X_arr[:i+1]), beta+(i+1-sum(X_arr[:i+1]))] for i i
new_beta = np.insert(new_beta, 0, [alpha, beta], 0)
beta_X = np.linspace(0, 1, 1000)
my_color = 'black'
fig, ax_arr = plt.subplots(ncols=6, figsize=(16,6), sharex=True)
for i, iter_ in enumerate([0, 5, 50, 200, 500, 1000]):
    ax = ax_arr[i]
    a, b = new_beta[iter_]
    beta_Y = sp.stats.beta.pdf(x=beta_X, a=a, b=b)
    ax.plot(beta_X, beta_Y, color=my_color, linewidth=3)
    if a > 1 and b > 1:
        mode = (a-1)/(a+b-2)
    else:
        mode = a/(a+b)
    ax.axvline(x=mode, linestyle='--', color='k')
    ax.set_title('Iteration %d: $\hat{\\theta}_{MAP}$ = %.2f'%(iter_, mode))
fig.tight_layout()

#2 star
theta = len(df_2)/len(df)
X_arr = np.random.choice([0, 1], p=[1-theta, theta], size=n)
print(sum(X_arr) / n)

new_beta = np.asarray([[alpha + sum(X_arr[:i+1]), beta+(i+1-sum(X_arr[:i+1]))] for i i
new_beta = np.insert(new_beta, 0, [alpha, beta], 0)
beta_X = np.linspace(0, 1, 1000)
my_color = 'black'
fig, ax_arr = plt.subplots(ncols=6, figsize=(16,6), sharex=True)
for i, iter_ in enumerate([0, 5, 50, 200, 500, 1000]):
    ax = ax_arr[i]
    a, b = new_beta[iter_]
    beta_Y = sp.stats.beta.pdf(x=beta_X, a=a, b=b)
    ax.plot(beta_X, beta_Y, color=my_color, linewidth=3)
    if a > 1 and b > 1:
        mode = (a-1)/(a+b-2)
    else:
        mode = a/(a+b)
    ax.axvline(x=mode, linestyle='--', color='k')
    ax.set_title('Iteration %d: $\hat{\\theta}_{MAP}$ = %.2f'%(iter_, mode))
fig.tight_layout()

#3 star
theta = len(df_3)/len(df)
X_arr = np.random.choice([0, 1], p=[1-theta, theta], size=n)
print(sum(X_arr) / n)

new_beta = np.asarray([[alpha + sum(X_arr[:i+1]), beta+(i+1-sum(X_arr[:i+1]))] for i i
new_beta = np.insert(new_beta, 0, [alpha, beta], 0)
```

```
beta_X = np.linspace(0, 1, 1000)
my_color = 'black'
fig, ax_arr = plt.subplots(ncols=6, figsize=(16,6), sharex=True)
for i, iter_ in enumerate([0, 5, 50, 200, 500, 1000]):
    ax = ax_arr[i]
    a, b = new_beta[iter_]
    beta_Y = sp.stats.beta.pdf(x=beta_X, a=a, b=b)
    ax.plot(beta_X, beta_Y, color=my_color, linewidth=3)
    if a > 1 and b > 1:
        mode = (a-1)/(a+b-2)
    else:
        mode = a/(a+b)
    ax.axvline(x=mode, linestyle='--', color='k')
    ax.set_title('Iteration %d: $\hat{\\theta}_{MAP}$ = %.2f'%(iter_, mode))
fig.tight_layout()

#4 star
theta = len(df_4)/len(df)
X_arr = np.random.choice([0, 1], p=[1-theta, theta], size=n)
print(sum(X_arr) / n)

new_beta = np.asarray([[alpha + sum(X_arr[:i+1]), beta+(i+1-sum(X_arr[:i+1]))] for i i
new_beta = np.insert(new_beta, 0, [alpha, beta], 0)
beta_X = np.linspace(0, 1, 1000)
my_color = 'black'
fig, ax_arr = plt.subplots(ncols=6, figsize=(16,6), sharex=True)
for i, iter_ in enumerate([0, 5, 50, 200, 500, 1000]):
    ax = ax_arr[i]
    a, b = new_beta[iter_]
    beta_Y = sp.stats.beta.pdf(x=beta_X, a=a, b=b)
    ax.plot(beta_X, beta_Y, color=my_color, linewidth=3)
    if a > 1 and b > 1:
        mode = (a-1)/(a+b-2)
    else:
        mode = a/(a+b)
    ax.axvline(x=mode, linestyle='--', color='k')
    ax.set_title('Iteration %d: $\hat{\\theta}_{MAP}$ = %.2f'%(iter_, mode))
fig.tight_layout()

#5 star
theta = len(df_5)/len(df)
X_arr = np.random.choice([0, 1], p=[1-theta, theta], size=n)
print(sum(X_arr) / n)

new_beta = np.asarray([[alpha + sum(X_arr[:i+1]), beta+(i+1-sum(X_arr[:i+1]))] for i i
new_beta = np.insert(new_beta, 0, [alpha, beta], 0)
beta_X = np.linspace(0, 1, 1000)
my_color = 'black'
fig, ax_arr = plt.subplots(ncols=6, figsize=(16,6), sharex=True)
for i, iter_ in enumerate([0, 5, 50, 200, 500, 1000]):
    ax = ax_arr[i]
```

```
        a, b = new_beta[iter_]
        beta_Y = sp.stats.beta.pdf(x=beta_X, a=a, b=b)
        ax.plot(beta_X, beta_Y, color=my_color, linewidth=3)
        if a > 1 and b > 1:
            mode = (a-1)/(a+b-2)
        else:
            mode = a/(a+b)
        ax.axvline(x=mode, linestyle='--', color='k')
        ax.set_title('Iteration %d: $\hat{\\theta}_{MAP}$ = %.2f'%(iter_, mode))
    fig.tight_layout()
```

```
0.0524
0.0481
0.1015
0.1897
0.5813
```




```
import random
random.seed(42069)

# Randomly sample n_train (100k) datapoints from each rating to form training data
n_train = 100000
data_1 = df_1.to_dict('records')
data_2 = df_2.to_dict('records')
data_3 = df_3.to_dict('records')
data_4 = df_4.to_dict('records')
data_5 = df_5.to_dict('records')

train = random.sample(data_1, n_train) + random.sample(data_2, n_train) + random.sampl
random.shuffle(train)
len(train)
```

```
500000
```

```
import string

# Using NLTK Library
import nltk
# import the stopwords collection from the nltkcorpus module
from nltk.corpus import stopwords
nltk.download('stopwords')
sWords = set(stopwords.words())


# New bag-of-words after removing punctuation capitalization and stopwords
wordCount = defaultdict(int)
punctuation = set(string.punctuation)
ascii_lower = set(string.ascii_lowercase)
```

```
whitespace = set(string.whitespace)

for d in train:
    r = ''.join([c for c in d['reviewText'].lower() if not c in punctuation and (c in
    for w in r.split():
        if not w in sWords:
            wordCount[w] += 1

print('num words case-insensitive and no punc: ', len(wordCount))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
num words case-insensitive and no punc:  407782
```

```
# Sort by popularity
counts = [(wordCount[w], w) for w in wordCount]
counts.sort()
counts.reverse()
words = [x[1] for x in counts[2:750]]

print('top 20 popular words: ')
for i in range(20):
    print(words[i], wordCount[words[i]])

print('top 20 least popular words: ')
for i in reversed(range(1, 20)):
    print(words[-i], wordCount[words[-i]])

words = set(words)
```

```
top 20 popular words:
like 177915
good 176169
story 115759
would 114727
great 112150
time 105217
really 104703
much 95481
even 92317
get 88493
see 84373
first 83358
well 83019
movies 77819
dont 77125
watch 74726
bad 70924
love 67341
dvd 65663
could 65145
top 20 least popular words:
dumb 5743
```

```
bunch 5734
shots 5730
killing 5727
personal 5715
edition 5690
potential 5677
doctor 5639
thriller 5638
showing 5625
brilliant 5622
solid 5613
leaves 5589
issues 5567
score 5563
acted 5560
musical 5559
mention 5557
worked 5546
```

Feature Extraction

```
wordId = dict(zip(words, range(len(words))))

def feature(datum):
    feat = [0]*len(words)
    r = ''.join([c for c in datum['reviewText'].lower() if not c in punctuation and (c
    for w in r.split():
        if w in words:
            feat[wordId[w]] += 1
    return feat


# Get smaller training sample
small_train = random.sample(train, int(len(train)/2))


X = [feature(d) for d in small_train]
train_set = np.asarray(X)
train_set.shape
```

```
(250000, 748)
```

K-Means Algorithm

```
def calcSqDistances(X, Kmus):
    N,D = X.shape
    K = Kmus.shape[0]
```

```
    sqDist = np.zeros((N, K), dtype=np.float32)

    for i in range(N):
        for j in range(K):
            sqDist[i,j] = np.linalg.norm(X[i] - Kmus[j])
    return sqDist

def determineRnk(sqDmat):
    m = np.argmin(sqDmat, axis=1)
    return np.eye(sqDmat.shape[1])[m]


def recalcMus(X, Rnk):
#     N, D = X.shape
#     Kmus = np.zeros((len(Rnk[0]), D))
#     for k in range(len(Rnk[0])):
#         Rnk_new = np.stack([Rnk[:, k], Rnk[:, k]], axis=1)
#         Kmus[k,:] = np.sum(np.multiply(X, Rnk_new), axis=0) / np.sum(Rnk[:,k], axis=
#     return Kmus
    return (np.divide(X.T.dot(Rnk), np.sum(Rnk, axis=0))).T


def runKMeans(K):
    X = train_set
    N, D = X.shape
    Kmus = np.zeros((K, D))
    rand_inds = np.random.permutation(N)
    Kmus = X[rand_inds[0:K],:]
    maxiters = 1000

    for iter in range(maxiters):
        sqDmat = calcSqDistances(X, Kmus)
        Rnk = determineRnk(sqDmat)
        KmusOld = Kmus
        Kmus = recalcMus(X, Rnk)
        if np.sum(np.abs(KmusOld.reshape((-1, 1)) - Kmus.reshape((-1, 1)))) < 1e-6:
            break
    return Kmus, Rnk


kmus5, Rnk = runKMeans(5)


kmus5
```

```
array([[0.08152832, 0.03834253, 0.22514463, ..., 0.08791874, 0.18458227,
        0.05953182],
       [0.00478015, 0.00590329, 0.03084578, ..., 0.00408564, 0.01022224,
        0.00978275],
       [0.15018022, 0.04365238, 0.4989988 , ..., 0.18662395, 0.3360032 ,
        0.15698839],
       [0.03231975, 0.02803816, 0.12084052, ..., 0.03204955, 0.07029285,
        0.03161308],
```

```
            [0.00454545, 0.04545455, 0.39545455, ..., 0.31818182, 0.33181818,
             0.01818182]])
```

```
Rnk
```

```
     array([[0., 0., 0., 1., 0.],
            [0., 0., 0., 1., 0.],
            [0., 1., 0., 0., 0.],
            ...,
            [0., 1., 0., 0., 0.],
            [0., 0., 0., 1., 0.],
            [0., 1., 0., 0., 0.]])
```

```
def predict(feat, Kmus):
    sqDmat = calcSqDistances(feat, Kmus)
    Rnk = determineRnk(sqDmat)
    return Rnk
```

Sanity Check and Results

5 Examples of Cluster 1 Reviews

```
cluster_1_idx = []
for i in range(len(Rnk)):
    if np.where(Rnk[i] == 1)[0] == 0:
        cluster_1_idx.append(i)

rev1 = [small_train[i] for i in cluster_1_idx]
rev1[:5]
```

```
     [{'overall': 3.0,
       'reviewText': "I saw this movie today, at 1:30 in the morning. And I was very
     excited.'Cause I love Jason and his Friday the 13th movies.\nI always liked
     them,and watched them since I was very little, I got scared like a baby with
     Jason(Of course that was many years ago.\nThe movie starts with a great slasher
     beginning. Rowan(Lexa Doig) and Jason(Kane Hodder) are in the same room and only
     one of them will survive...No. Rowan and Jason get both frozen in time when she
     suddenly tryes to scape from him(The year is 2010).\nAfter, like 400 years, this
     two frozen bodys are found by a goup of students. The take them to their ship
     and bring Rowan back to like with the new technology.\nBut she's not the only
     one that wakes up, Jason begins his large(Very large, I dare you to try to count
     them) body count, just after he gets up from bed.\nJason X is a real well done
     movie. It's visually amazing, the athmosphere it's creepy, and it's a real big
     step for the Friday the 13th movies.\nActing was pretty good for a Friday movie,
     I specially enjoyed Mellysa Ade's(Jannesa) and Lexa's(Rowan)
     performances.\nJannesa was like the funny character, always making sarcastic
     jokes.\nAnd Rowan is the main character, she's very pretty.\nBut they were
     nothing compared to the master of Friday the 13th, that's right, I'm talking
```

about Kane Hodder. He's perfect for Jason, he was born for becoming Jason. He's a real horror king.\nWhat I really loved here was Crystall lake 3-D version, I mean, it was like and old Friday the 13th movie mixed up with a new one. It was just great. Uber-Jason looked awesome, but I just wanted a longer part with him.\nWhat I really hated here was KM-14, it was so stupid!!!.\nI rate this with 3 stars out of five, 'cause the movie was a little too stupid and sarcastic for a Friday movie. But I have to say that is one of the best sequels in the series.\nThe go like this(From great to bad)\n1-Part 6(Awesome)\n2-Part 4(Tommy rules!)\n3-Part X(Cool)\n4-Part 1(The original, but not the best)\n5-Part 7(Really cool kills)\n6-Part 3(Just liked it 'cause it was 3-D and Jason gets his mask in there)\n7-Part 8(Loved the Manhattan secuence, the rest of the movie sucked)\n8-Part 2(Not bad sequel, not great either)\n9-Part 5(There's no Jason, so ther's no more to say)\n10-Part 9(What a BAD movie)\nJASON X it's a real horror movie for fans, but it's true. If your not a Jason fan, you'll never like this movie.\nBut I did, it's pure fun, I mean it doesn''t make any sense, but It was an hour and a half enterteining flick.\nJASON X\nDirected by J. Isaacs\nStarring: Kane Hodder,\nLexa Doig, Lisa Ryder\nRunning time: 93 minutes\nRated: R\nOrigina: U.S.A.\nYear: 2002\nFINAL GRADE: C+(7)",
  'summary': 'JASON IN SPACE? Only good for fans'},
 {'overall': 1.0,
  'reviewText': "Before I go any further, you may want to know where this is coming from. I am 15, Catholic, home-schooled, and look upon the use of contractions in formal writing as the first sign that a book is bad. Rowling uses a lot of contractions.\n\nI cannot compare the movie to the book, because I have never had the time necessary to waste on it - though it only took me 2 hrs to finish the first two.\n\nThe movie was just stupid - any lover of plot or logic will do well to keep away.\n\nItem #1: under-age wizards are not alowed to use magic away from the school.\n\nThis is a sensible enough rule, but it seems as though a) it is never enforced; or b) an exception can be made for 'the one who lived' (or whatever he is called)\n\nItem #2: Harry is disturbed that his god-father is after him, and holds him in contempt for being a murder.\n\nCome on!! I mean, his life is always in danger because of Valdemort anyway, so lighten up!! As for not liking his god-father's being a murder, what about Harry's blowing-up his aunt?!? That's attempted murder!\n\nItem #3: Harry's god-father was one of his parents' best friends.\n\nWitches and wizards obviously cannot believe in God. How can these atheistic/agnostic witches have god-parents? Why would they even baptize their kids? This brings up a similar point: why do they celebrate Christmas? Wouldn't they more then likely celebrate Winter Solstice or the Feast of Thor or something?\n\nItem #4: a wizard has to learn spells.\n\nWouldn't simple mathematics help with this

5 Examples of Cluster 2 Reviews

```
cluster_2_idx = []
for i in range(len(Rnk)):
    if np.where(Rnk[i] == 1)[0] == 1:
        cluster_2_idx.append(i)

rev2 = [small_train[i] for i in cluster_2_idx]
rev2[:5]
```

[{'overall': 3.0, 'reviewText': 'Great', 'summary': 'Three Stars'},
 {'overall': 1.0,
  'reviewText': "This movie would lead you to believe you can not Kiss before

```
      marriage.... We are a Christian Family I've been married to my wife for
      43years... our Children have both been married long a time. one for16years the
      other for 11years My wife and I waited until we got married.. however we did
      Kiss..... This Movie is Over the TOP....",
        'summary': 'Over the Top'},
       {'overall': 2.0,
        'reviewText': 'This movie has no idea whether its a buddy cop comedy, or a
      parody of one. At points its serious, and then it is the complete opposite the
      next minute, to the point of being rediculous. Will Ferell never "goes there"
      with his classic Anchorman rants, and Mark Wahlberg plays the most annoying
      character ever. It was just a mess of a movie, and I wanted to give it a low
      star rating, b/c I was sickened to see 4 stars, and so many positive reviews.
      Click the dislike button now, you freaking no-lifes. I\'m not gonna respond to
      hostile comments either, so just to get this out of the way: SCREW YOU.',
        'summary': 'A throwaway movie. Waste of time.'},
       {'overall': 4.0,
        'reviewText': 'Was a decent action movie with a very basic revenge story line.
      No issues with streaming or audio.',
        'summary': 'Four Stars'},
       {'overall': 3.0,
        'reviewText': 'This time Sylvester Stallone is against a bunch of ax wielding
      serial killers, with nylons over their faces, who terrorize the LA
      nights...\n\nWhen a fashion model happens to see the ugly face of a sadistic
      psychopath (Brian Thompson), Ingrid (Brigitte Nielsen) becomes the main target
      of the secret "New World" society stopping at nothing to slain
      her...\n\nLieutenant Marion Cobretti (Stallone), in his gun metal-gray classic
      Mercury, and armed with guns, knives, grenades, and firearms, is assigned to
      protect the statuesque blonde...\n\nThe movie is too violent and too bloody and
      contains one of the most interesting car chase sequences ever filmed...',
        'summary': '"I don\'t deal with psychos, I put them away!"'}]
```

5 Examples of Cluster 3 Reviews

```
cluster_3_idx = []
for i in range(len(Rnk)):
    if np.where(Rnk[i] == 1)[0] == 2:
        cluster_3_idx.append(i)

rev3 = [small_train[i] for i in cluster_3_idx]
rev3[:5]
```

```
       {'overall': 2.0,
        'reviewText': 'If one were to look at Inside on any level other than that of a
      splatter horror film in the vein of a poor man\'s Lucio Fulci, it would be a sad
      response to the current state of horror. Inside is a film that seems like it
      will only appeal to those who think gore-- no matter how realistic and well
      executed it may be-- means a film is effective. That would be a falsity. As
      Inside both lacks in nearly every department aside from conjuring some adamantly
      gruesome images, and is the antithesis of  both thought and suspense.\n\nGood
      horror movies for me are not unlike that of suspense or thrillers. In fact they
      are often much, much better because of their sense of both baroque and dire
      atmosphere and a feeling of all-bets-are-off bleakness. Whether you are Alfred
      Hitchcock, Brian De Palma, or Dario Argento you must understand all three of
      these genres. As Hitchcock put it: "There is no terror in the bang, only in the
```

anticipation of it."\n\nInside is all about the bang. Any sense of anticipation, (with the exception of one humorously ironic scene involving a toaster,) is shoddy and amateur: often being telegraphed long before it happens. Suspense and actual tensions should always be a necessity. Gore without this is simply for pussies. As Inside is a film dependent on startling images, much like the aforementioned films of Lucio Fulci. Yet, in the case of effects, although it may have improved on realistic bloodshed, it sadly lacks the best element of that auteur\'s Grindhouse output in which the dreamlike and otherworldly whimsy of films like The Beyond is rather traded for nothing of real artistic merit or value.\n\nThough this writer does not have a problem at all with bloodshed, it must be said that after you see a gore scene once, no matter how well executed, you become desensitized; it\'s just not that effective anymore. It is the bang without a menacing triggerman-- yet, if you have a scene, gore or not in tact, and have an ample amount of suspense built around character development, pacing, tension and attack and release, then you have a film that will withstand multiple viewings.\n\nOther than bloodshed, Inside tries to make a bleak atmosphere through its use of shadows (doesn\'t work, by the way) and its angry lead character. Yet, the film does not earn it, rather the character of the pregnant Sarah (Alysson Paradis) feels as static as a walking dead extra in a Romero zombie film. Though the audience knows she is mad at the world and misses her husband, but it falls into telling not showing character development. Inside "tells" us to feel empathetic for this character and all the victims due to their ordeal, but it does not establish any sort of relationship with any of them of any worth. Although this may seem intentional, especially involving Sarah\'s possible more than friendship with her boss, this sense of mystery certainly doesn\'t help, rather it just seems lazy.\n\nOf course if one where to really want to they can pick apart many dumb character moves, as well. However, one must know that often you simply won\'t have a movie without them. Horror, especially this genre, utilizes a narrative that relies on mistakes, often as a commentary--- whether intentional or not, it depicts humans for what they are, flawed animals that often make mistakes when drastic times call for drastic measures. Dumb character moves can be forgiven if they resemble the character. It\'s again too bad that Inside doesn\'t have much character depth or layers, which does not exactly help matters when the bad logic comes into play.\n\nThe half-twist comes as braindead and unclear as the rest of the movie-- even negates itself in logic somewhat when Sarah mentions that she was told there were, "No survivors." A scene involving a believed to be dead character that can be compared to something out of a zombie-movie seems more hokey than anything in context to the rest of this otherwise played-straight film. The final shot of Inside, also, rather feels like it has false feelings of grandeur.\n\nIn the end, Inside may have a taboo feel like some of those golden-age horror movies, and possibly can be seen for its no-holds-barred approach with some of its not-easily-shaken images. It\'s too bad the budget\'s load was blown on gore shots than a good script.\n\nScore: 5.0 / 10 (in 0.5 increments)',
  'summary': 'Not Truly Terrifying'}]

5 Examples of Cluster 4 Reviews

```
cluster_4_idx = []
for i in range(len(Rnk)):
    if np.where(Rnk[i] == 1)[0] == 3:
        cluster_4_idx.append(i)
```

```
rev4 = [small_train[i] for i in cluster_4_idx]
rev4[:5]
```

```
[{'overall': 3.0,
  'reviewText': 'I\'m sure a lot of people have asked what went wrong with "Twin Peaks: Fire Walks with Me." It adds nothing new to the brilliance of the series and it\'s purpose seems to be to show the audience what kind of person Laura Palmer was, which we already pick up from the series. Story and characters are sacrificed for the purpose of shocking people with Laura Palmer\'s actions, but it just doesn\'t work. Things are vulgar, annoying, and often hard to take. Plus, since we already know who killed Laura Palmer (from the TV series), there is nothing new to be said.\nThe film is too long for its own good, with the entire 20 odd opening minutes a pointless addition to the rest of the story; it just slows things down and becomes boring. While entertaining at times, especially when Laura Palmer comes into the picture (after the excruciating start), this was extremely disappointing. It seems the whole concept of Twin Peaks has went to David Lynch\'s head a little bit. This being the movie, David Lynch decided to turn up the sexual antics and the bizarre plot.\nAside from that, the setting is breathtaking and bizarre all at once, and Sherly Lee is just superb as Laura Palmer. However, this just doesn\'t do any justice whatsoever to the TV series.',
  'summary': 'Watch at your own risk!!!'},
 {'overall': 1.0,
  'reviewText': 'Thank you to the one star reviewer who gave me a heads up about this. As I result, I will contribute a one-star without watching the movie.\n\nSince Bridesmaids--one of the most disgusting movies I have ever had to stop watching--I have started to watch for this s*** (so to speak). I wish they would develop a new rating sytem that would warn us. I can handle sex and "foul" language, and I really don\'t need to be warned about them. But fecal scenes--oh, yeah, and vomit scenes--are right up there with violence, as far as I\'m concerned. Could we have a special star system for the gross-out factor?',
  'summary': 'More fecal scenes!'},
 {'overall': 3.0,
  'reviewText': "THE TROLLENBERG TERROR or THE CRAWLING EYE is a British production written by Jimmy Sangster and directed by Quentin Lawrence in 1958.\n\nGiant crawling eyes are hiding in a radioactive fog resting on the slopes of the Trollenberg mountain in Switzerland. When infortunate climbers enter the fog, they are simply decapitated by the eyes. By the tentacles of the eyes, I mean. Meanwhile, a young British girl with telepathic powers is attracted by the mountain. The eyes send two zombies to kill her but she's saved by the hero Forrest Tucker, a United Nations observer.\n\nWell, that's a screenplay, isn't it ? A little bit irrational maybe, but, remember, we're in the fifties and strange things happened then in the world like the birth of John Travolta, for instance, so let's be tolerant. The copy presented by Image is superb and the actors are surprisingly good for this kind a production. Recommended if you're a movie lover.\n\nA DVD zone they're here but they are so stupid.",
  'summary': "I'M  WATCHING  YOU"},
 {'overall': 2.0,
  'reviewText': 'Olympus Has Fallen is one of the worst movies ever.\nThe story is only possible if a multitude of idiotic screw ups ALL occur.\n1. Unidentified aircraft cruises into DC w/o ID and early prevention.\n2. Unidentified aircraft opens fire, likely without sophisticated aiming systems, and takes down 2 fighter planes within seconds.\n3. Unidentified aircraft opens fire, likely without sophisticated aiming systems, and pretty much takes out the whole security team on the roof.\n4. A huge crew of terrorists 1) get pass customs, 2)
```

```
none get picked up by CIA/FBI/NSA or what have you, 3) get access to heavy
artillery w/o being caught by aforementioned agencies, 4) modify vehicles
without notice near DC.\n5. After the aircraft guns down the units on top, the
large crew of terrorists with heavy artillery guns down the remaining armed
```

5 Examples of Cluster 5 Reviews

```
cluster_5_idx = []
for i in range(len(Rnk)):
    if np.where(Rnk[i] == 1)[0] == 4:
        cluster_5_idx.append(i)

rev5 = [small_train[i] for i in cluster_5_idx]
rev5[:5]
```

```
[{'overall': 4.0,
  'reviewText': "Wrestlemania is the biggest event of the year for wrestling
fans around the world. We have seen some great matches in Wrestlemania past such
as Hogan/Andre(WM3), Hogan/Warrior(WM6) and so many others. So did this PPV live
up to the hype or did it sink like the Titanic? Well I'll go down the list of
matches match by match. So lets get started: (I'll be rating everything out of 5
stars)\n\nMoney in the Bank was the first match of the night, and while it
wasn't the greatest Money in the bank it was still very enjoyable and as always
included tons of crazy spots. My only gripe was even though I am a fan of CM
Punk I would have liked to have seen Shelton or MVP win it also it could have
been a tad bit longer. 4 out of 5\n\n25 Diva Battle Royal was very bad. First of
all the divas didn't have a proper entrance, they all just came out together
while Kid Rock played. So, you didn't know who was all in the match except for a
few divas here and there. I found out the next day that former WWE diva Joy
Giovani was in it, but I would of never known that because well we were never
told or she was never seen. So, that was a huge disappointment, as the match
which could have been decent ended up being one giant mess. 1 out of
5\n\nJericho vs The Legends was ok, at first it was boring and slow and then
when it came down to just Jericho and Steamboat it took off. Steamboat for being
up there in years looks like he did years ago. He looks like he can still go and
put on a good 20 min match. I hope at Backlash Jericho faces Steamboat in a one
on one match. 3 out of 5\n\nJeff Hardy vs. Matt Hardy was better then I
expected, I remember when these two guys feuded years ago and their matches
weren't that good. Well this match was a lot better then I had thought but still
could have been better then it was. The double table spots was cool as was the
Jeff jumping over the ladder and missing Matt. The ending with the Twist of Fate
using the chair was sick and was a great ending to this match. I doubt this feud
is over so lets hope these two can continue to put on great matches. 3 out of
5\n\nJBL (c) vs. Rey Mysterio for the IC title was a joke, literally maybe thats
why Rey came out as the joker. I don't have much to say about this match as it
was only 21 seconds long. So all im saying is it sucked. 1 out of 5 stars\n\nThe
Undertaker vs. Shawn Michaels, OMG this match was great and I mean great. This
should have been the main event. I t was so good it made the 2 world title
matches look like crap compared to it and it made it hard to follow it up. The
match is worth buying the event just for this match alone. The spot when Taker
flew over the top rope and on to the camera man was sick as hell, I thought
Taker and the cameraman were dead. The ending with both guys hitting their
finishers and only getting a two count was awesome this match has to be Taker
greatest Mania match ever and also one of the best matches we have seen in WWE
```

```
in a while. 5 out 5\n\nAs for the last two matches, for the Edge (c) vs. John
Cena vs. Big Show it was boring and I really can't remember all that happend in
the match except for the spots where Cena picked up Big Show and Edge at the
same time and Big SHow stuckn the ropes, and Cenas entrance with 300 Cena
clones. Cena winning was predictable as Big Show has only won 1 Wrestlemania
match in his life. 2 out 5\n\nTriple H vs Randy Orton was a complete and total
let down, they had this really good feud going and had it very personal with
great segments leading up to the match. The match in no way lived up to its
expectations and as I said thats the reason why the Taker/HBK match should have
been last it was just too good to have any other match follow it up. The ending
was dumb and nothing special really happened in this match. I thought we were
going to have someone come down and possibly help Orton win and turn on HHH but
nothing. 2.5 out of 5\n\nOverall, it was pretty good, match of the night was
Taker/HBK and while there were some matches that sucked, it was Mania and thats
good enough for me. [...] I still enjoyed it and I'm still glad I ordered it.
The DVD will be out on May 19, according to Amazon as its up for Pre-order, so
I'll be picking it up and recommend that you do to.\n\nReview brought to you by
Dreadfulentertainment.com",
  'summary': 'Not the best but still a good event'}
```

Mean Rating of Each Cluster

```
cluster_1_ratings = [d['overall'] for d in rev1]
cluster_2_ratings = [d['overall'] for d in rev2]
cluster_3_ratings = [d['overall'] for d in rev3]
cluster_4_ratings = [d['overall'] for d in rev4]
cluster_5_ratings = [d['overall'] for d in rev5]

np.mean(cluster_1_ratings), np.mean(cluster_2_ratings), np.mean(cluster_3_ratings), np
```

```
(2.846024485402933,
 3.077421000086813,
 2.800961153384061,
 2.7687527279529442,
 3.309090909090909)
```

Mode Rating of Each Cluster

```
from scipy import stats
stats.mode(cluster_1_ratings)[0][0], stats.mode(cluster_2_ratings)[0][0], stats.mode(c
```

```
(2.0, 5.0, 2.0, 2.0, 4.0)
```