Contents

TOPIC MODELING – IMDB REVIEW

I. Manually extract topics and associated terms from imdb_labelled.txt

Open file imdb_labelled.txt and select 100 first review into Excel. Manually extract the possible topics and the words associated with each topic (the terms are in bold on the left column). Result is **roughly 5-7 topics** as follows:

- **Plot / Story line:** plot, lines, message, predictable, screenplay, content, character, conception, idea, moment, story
- **Acting:** character, acting, act, actor, actress, casting, cast, talented, co-star, leading, performance, convincing
- **Cinematography / Directing:** artiness, camera angles, scenes, cinematography, directing
- **Music:** music, song
- **Effect / Post-production:** editing, structure, cinema, graphics, effects
- **Movie genres:** game, series, horror, comedy, suspense
- **Production / Budget:** budget, cost, production
- **Quality:** waste, masterpiece, unfunny, generic, funny, regret, resounding, disappointed

	plot	acting	Directing / post production			production budget	theme / genre	quality
			cinematography	music effects	special effects			
A very, very, very slow-moving, aimless movie about a distressed, drifting young man.	1							
Not sure who was more lost - the flat **characters** or the audience, nearly half of whom walked out.		1						
Attempting **artiness** with **black & white** and clever **camera angles**, the movie disappointed - became even more ridiculous - as the **acting** was poor and the **plot** and **lines** almost non-existent.	1	1	1					
Very little **music** or anything to speak of.				1				
The best **scene** in the movie was when Gerardo is trying to find a **song** that keeps running through his head.			1	1				
The rest of the movie lacks art, charm, meaning... If it's about emptiness, it works I guess because it's empty.								1
Wasted two hours.								1
Saw the movie today and thought it was a good effort, good **messages** for kids.	1							
A bit **predictable**.	1							
Loved the **casting** of Jimmy Buffet as the science teacher.		1						
And those baby owls were adorable.		1						
The movie showed a lot of Florida at it's best, made it look very appealing.			1					
The **Songs** Were The Best And The Muppets Were So Hilarious.				1				
It Was So Cool.								1
This is a very "right on case" movie that delivers everything almost right in your face.								1
It had some average **acting** from the main person, and it was a low **budget** as you clearly can see.		1				1		
This review is long overdue, since I consider A Tale of Two Sisters to be the single greatest film ever made.								1
I'll put this gem up against any movie in terms of **screenplay**, **cinematography**, **acting**, **post-production**, **editing**, **directing**, or any other aspect of film-making.	1	1	1					
It's practically perfect in all of them a true **masterpiece** in a sea of faux "masterpieces.								1
The **structure** of this film is easily the most tightly constructed in the history of cinema.			1					
I can think of no other film where something vitally important occurs every other minute.			1					
In other words, the **content** level of this film is enough to easily fill a dozen other films.	1							

II. LDA with words (noun, verb, adj, adv)

1. Load data into jupyter notebooks (the code for part II is in file "**9. Part IV - 2 - Python Code**")

	review	label
0	A very, very, very slow-moving, aimless movie ...	0
1	Not sure who was more lost - the flat characte...	0
2	Attempting artiness with black & white and cle...	0
3	Very little music or anything to speak of.	0
4	The best scene in the movie was when Gerardo i...	1
...	...	...
995	I just got bored watching Jessice Lange take h...	0
996	Unfortunately, any virtue in this film's produ...	0
997	In a word, it is embarrassing.	0
998	Exceptionally bad!	0
999	All in all its an insult to one's intelligence...	0

1000 rows × 2 columns

2. Text Preprocessing

   a. check the list of stop words -> add to that list the words 'movie, movies, film, films' because the dataset is about movie review and they will appear a lot in the dataset and not bring more meaning to form the topic later on

```
# create a spacy object, disable parser and ner for the script to run a bit faster
nlp = spacy.load('en_core_web_sm', disable=['parser','ner'])

#get the list of stop words
stopwords = stopwords.words('english')

#add the words movie, movies, film, films to the stopword list
stopwords.append('movie')
stopwords.append('movies')
stopwords.append('film')
stopwords.append('films')
print(stopwords)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'h
e', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'w
hat', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had',
'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with',
'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'o
ver', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most',
'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'shoul
d', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'had
n', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't",
'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't", 'movie', 'movies', 'film', 'films']
```

   b. create a function preprocess that clean up the text before modelling
      - takes in a review
      - use gensim.untils.simple_preprocess to convert text to lowercase, tokenize text, etc
      - remove stop words
      - if the word is either **noun, verb, adj, or adv**, lemmatize the word. If not, ignore the word
      - return an array of clean words

```
# function to preprocess the text
def preprocess(texts):
    lemmatizer = WordNetLemmatizer()
    allowed_postags = ['NOUN','VERB','ADJ','ADV']
    filter_sentence = []
    temp = []
    words = gensim.utils.simple_preprocess(str(texts), deacc=True) # gensim - lowercase, tokenize

    words = [w for w in words if not w in stopwords] # stopwords removal

    temp = ' '.join(x for x in words) # combine the tokens into a sentence again
    doc = nlp(temp)
    for word in doc: # go through each words in that clean sentence
        if word.pos_ in allowed_postags: # remove words that are not verb or noun
            filter_sentence.append(word.lemma_) # lemmatization
    return filter_sentence
```

   c. clean up the text for all reviews

```
# preprocess text from all reviews
corpus = [preprocess(line) for line in review]
corpus[:5]
```

```
[['slow', 'move', 'aimless', 'distressed', 'drift', 'young', 'man'],
 ['sure', 'lose', 'flat', 'character', 'audience', 'nearly', 'half', 'walk'],
 ['attempt',
  'artiness',
  'black',
  'white',
  'clever',
  'camera',
  'angle',
  'disappoint',
  'become',
  'even',
  'ridiculous',
  'act',
  'poor',
  'plot',
  'line',
  'almost',
  'existent'],
 ['little', 'music', 'speak'],
 ['good', 'scene', 'gerardo', 'try', 'find', 'song', 'keep', 'run', 'head']]
```

d. build a dictionary with these clean words

```
# build the dicationary with gensim
dictionary = corpora.Dictionary(corpus)
len(dictionary)
```

```
2119
```

e. transform the dictionary into bag-of-word form

```
# convert corpus into bag-of-words format
bow = [dictionary.doc2bow(line) for line in corpus]
print(bow[0][0:20])
```

```
[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1)]
```

3. Topic Modelling

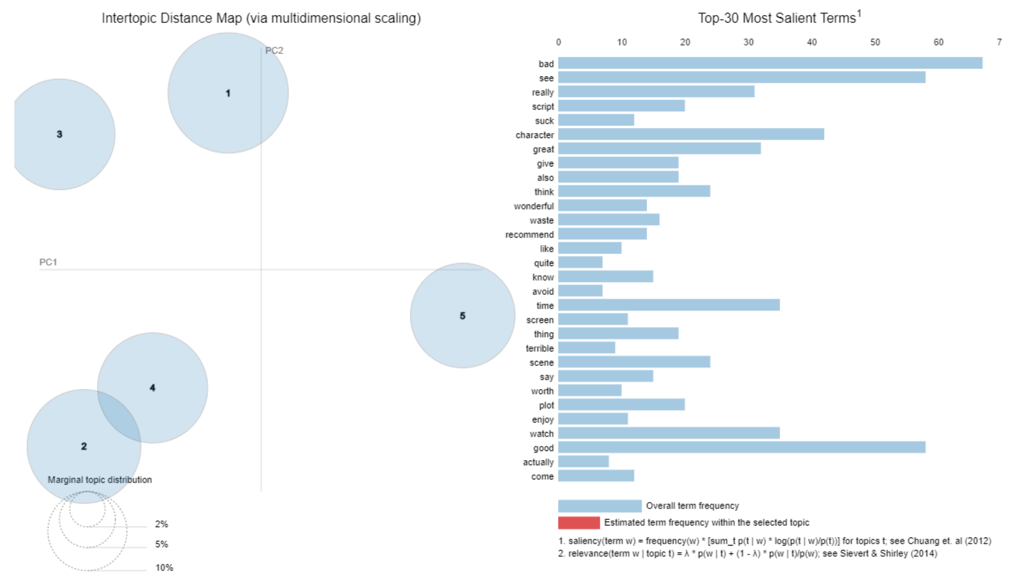
a. Run the unguided LDA with number of topics 5, 7 and 10 to see which number of topics gives the best clusters for our dataset.

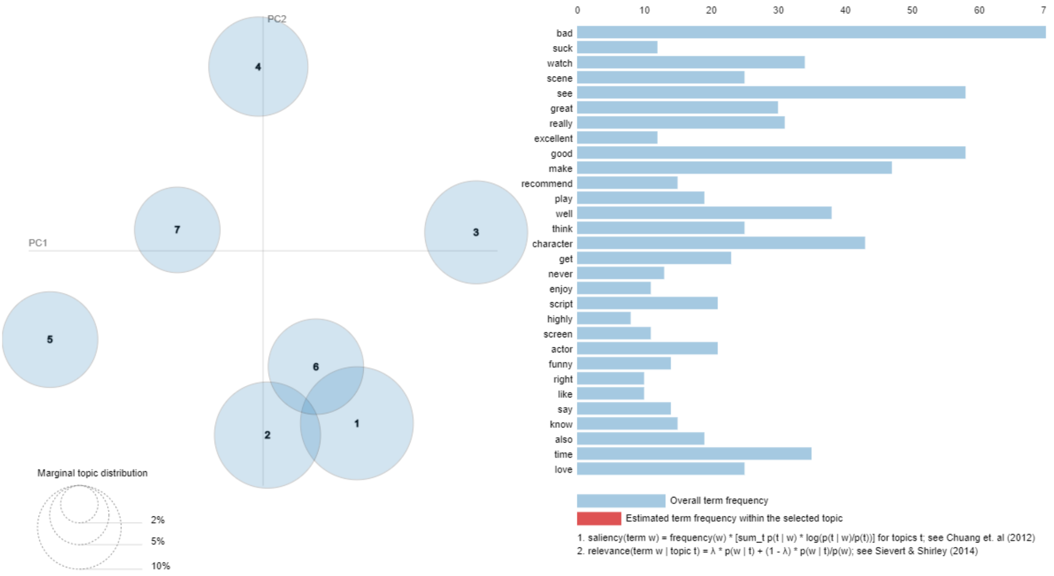
What we are looking for:

- Clusters are well spread across different directions and cover most of the area
- Clusters are not too close to each other or overlap (as we try to assign each topic to a single unique topic as possible).

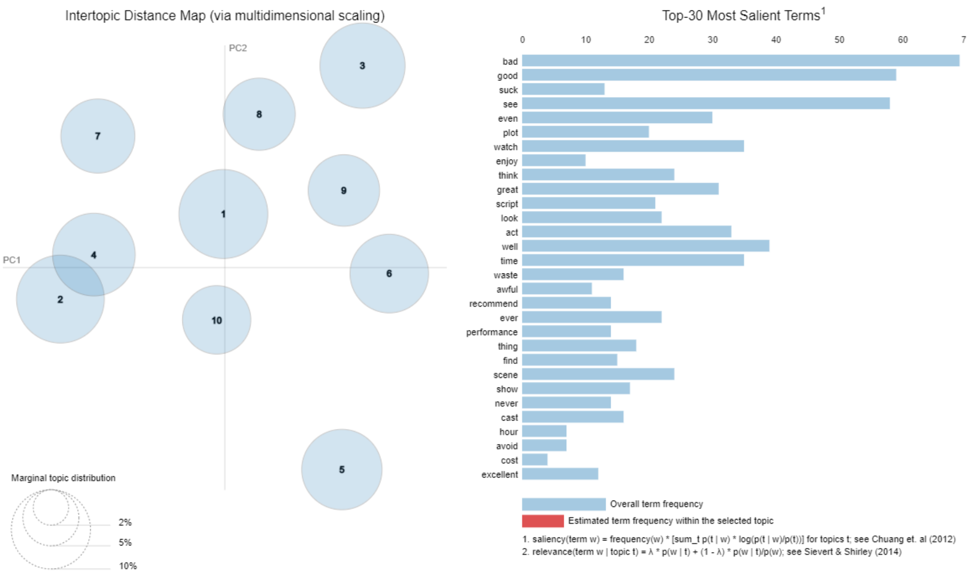
Results:

- 5 and 7 aren't quite good because the clusters are overlapped and don't spread across all directions much and tend to be in the same area





- 10 gives a good result as the topic clusters are quite equal in size with little overlap and well spread across all direction and cover the entire area.



Therefore, we procced next step with number of topics 7 and 10

b. Once the number of topics is chosen, run the unguided LDA on 7 and 10 topics.

- This time, get the words associated with each topic and calculate the probability for each topic in each review.
- Assign the topic for each review by selecting the topic with highest probability
- Generate a frequency table to check the distribution of all topics across all review

Results:

- For 7 topics, the result is really bad. The model assigns all review into only 1 topic (topic 1)

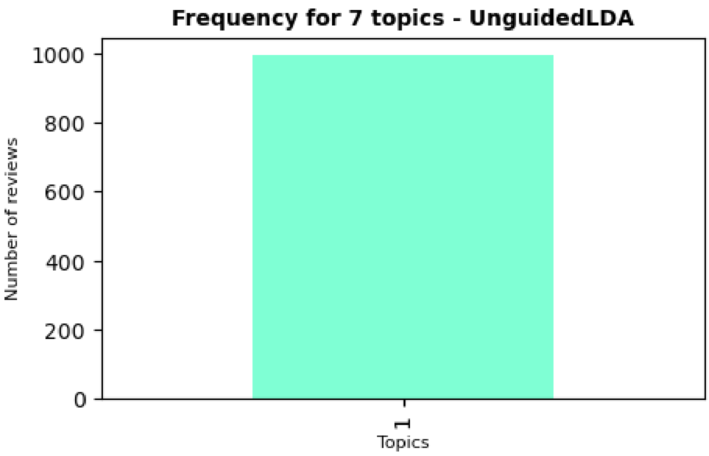
```
model, result = test_eta('auto', dictionary, ntopics=7)
```

```
Perplexity: -8.95
****************************************************************************************************
Topic 0: ['really', 'still', 'end', 'dialogue', 'short', 'way', 'year', 'seem', 'music', 'like']
Topic 1: ['bad', 'make', 'great', 'well', 'ever', 'look', 'real', 'write', 'thing', 'fast']
Topic 2: ['character', 'watch', 'script', 'art', 'waste', 'know', 'work', 'truly', 'screen', 'life']
Topic 3: ['act', 'even', 'plot', 'also', 'take', 'line', 'interesting', 'use', 'funny', 'give']
Topic 4: ['story', 'wonderful', 'scene', 'play', 'never', 'love', 'enjoy', 'drama', 'performance', 'excellent']
Topic 5: ['comedy', 'right', 'cast', 'come', 'top', 'role', 'many', 'portrayal', 'totally', 'awful']
Topic 6: ['see', 'good', 'time', 'think', 'recommend', 'get', 'suck', 'actor', 'go', 'become']
****************************************************************************************************

 A very, very, very slow-moving, aimless movie about a distressed, drifting young man.  ['(0, 15.4%)', '(1, 21.0%)', '(2, 11.6%)', '(3, 11.9%)', '(4, 14.2%)', '(5, 8.
0%)', '(6, 17.8%)']

 Not sure who was more lost - the flat characters or the audience, nearly half of whom walked out.  ['(0, 14.8%)', '(1, 20.9%)', '(2, 11.9%)', '(3, 11.9%)', '(4, 14.
7%)', '(5, 8.0%)', '(6, 17.8%)']
```

Review	Topic	Probability
A very, very, very slow-moving, aimless movie ...	1	0.210344
Not sure who was more lost - the flat characte...	1	0.209252
Attempting artiness with black & white and cle...	1	0.204117
Very little music or anything to speak of.	1	0.208477
The best scene in the movie was when Gerardo i...	1	0.205599
...	...	...
I just got bored watching Jessice Lange take h...	1	0.210269
Unfortunately, any virtue in this film's produ...	1	0.207247
In a word, it is embarrassing.	1	0.209558
Exceptionally bad!	1	0.212281
All in all its an insult to one's intelligence...	1	0.210039



- **For 10 topics, the result is much better** as the model assigns reviews into 10 different topics with the majority is assigned into topic 4.

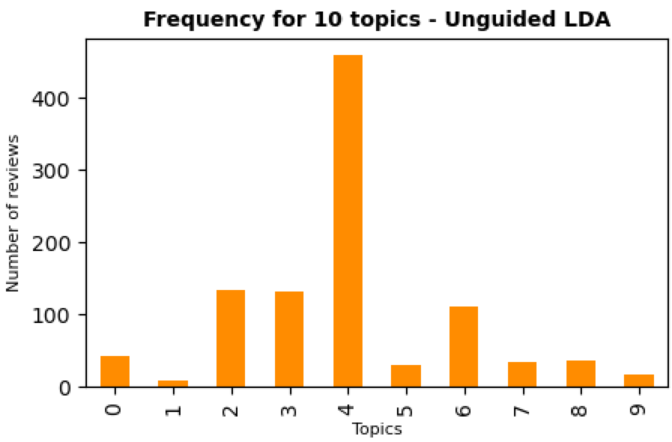
```
model1, result1 = test_eta('auto', dictionary, ntopics=10)
```

```
Perplexity: -9.32
****************************************************************************************************
Topic 0: ['make', 'fast', 'take', 'avoid', 'like', 'funny', 'believable', 'special', 'easy', 'experience']
Topic 1: ['classic', 'share', 'charming', 'heart', 'sentiment', 'masterpiece', 'release', 'original', 'joy', 'race']
Topic 2: ['bad', 'story', 'wonderful', 'real', 'art', 'go', 'work', 'truly', 'show', 'far']
Topic 3: ['act', 'look', 'even', 'plot', 'end', 'also', 'drama', 'year', 'line', 'performance']
Topic 4: ['good', 'time', 'watch', 'well', 'scene', 'still', 'script', 'suck', 'waste', 'short']
Topic 5: ['really', 'dialogue', 'seem', 'lot', 'mess', 'keep', 'effect', 'beautiful', 'enough', 'serious']
Topic 6: ['see', 'think', 'recommend', 'get', 'play', 'never', 'actor', 'love', 'enjoy', 'become']
Topic 7: ['great', 'ever', 'write', 'know', 'thing', 'screen', 'pace', 'fail', 'writer', 'long']
Topic 8: ['character', 'music', 'subtle', 'come', 'highly', 'little', 'hole', 'appreciate', 'wonderfully', 'age']
Topic 9: ['entire', 'boring', 'quite', 'simply', 'rate', 'camera', 'thoroughly', 'interest', 'level', 'new']
****************************************************************************************************

 A very, very, very slow-moving, aimless movie about a distressed, drifting young man.  ['(0, 4.3%)', '(2, 16.1%)', '(3, 17.2%)', '(4, 32.2%)', '(5, 4.1%)', '(6, 7.
7%)', '(7, 3.8%)', '(8, 12.7%)', '(9, 1.5%)']

 Not sure who was more lost - the flat characters or the audience, nearly half of whom walked out.  ['(0, 3.8%)', '(2, 6.9%)', '(3, 7.9%)', '(4, 26.9%)', '(5, 3.6%)',
'(6, 14.3%)', '(7, 3.4%)', '(8, 23.6%)', '(9, 9.0%)']
```

Review	Topic	Probability
A very, very, very slow-moving, aimless movie ...	4	0.322051
Not sure who was more lost - the flat characte...	4	0.268747
Attempting artiness with black & white and cle...	3	0.501455
Very little music or anything to speak of.	8	0.347308
The best scene in the movie was when Gerardo i...	4	0.590056
...	...	...
I just got bored watching Jessice Lange take h...	4	0.376393
Unfortunately, any virtue in this film's produ...	4	0.448228
In a word, it is embarrassing.	2	0.312106
Exceptionally bad!	2	0.312115
All in all its an insult to one's intelligence...	4	0.323558



c. Since the results for unguided LDA aren't so good, we can try to improve the result of 7 topics by running a guided LDA with predefined list of terms associated with each topic (using the list of keywords already manually extracted from part I) and recalculate all the steps above.

```
predefined_topic = {
   'plot':0, 'lines':0, 'line':0, 'message':0, 'predictable':0, 'screenplay':0, 'content':0, 'character':0, 'conception':0, 'idea':0, 'moment':0, 'story':0, 'write':0,
    'character':1, 'characters':1, 'acting':1, 'act':1, 'actor':1, 'actress':1, 'casting':1, 'cast':1, 'talented':1, 'star':1, 'leading':1, 'performance':1, 'convincin
    'artiness':2, 'camera':2, 'angles':2, 'scene':2, 'scenes':2, 'cinematography':2, 'direct':2, 'directing':2, 'art':2,
    'horror':3, 'comedy':3, 'cartoon':3,'game':3,'suspense':3,'series':3, 'drama':3,
    'budget':4, 'production':4,'cost':4,
    'waste':5, 'masterpiece':5, 'unfunny':5, 'generic':5, 'funny':5, 'regret':5, 'resounding':5, 'disappointed':5, 'recommend':5, 'boring':5,
    'music':6, 'song':6, 'songs':6,
    'edit':7, 'cinema':7, 'structure':7, 'editing': 7,
    'graphics':8, 'effect':8, 'effects':8, 'special':8
}
eta = create_eta(predefined_topic, dictionary, 10)
```

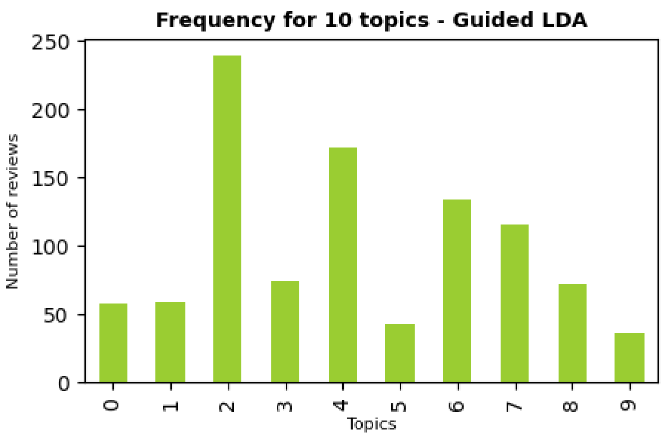
```
model2, result2 = test_eta(eta, dictionary, ntopics=10)
```

```
Perplexity: 12.57
****************************************************************************************************
Topic 0: ['story', 'wonderful', 'plot', 'also', 'write', 'line', 'pace', 'involve', 'predictable', 'many']
Topic 1: ['actor', 'suck', 'performance', 'way', 'excellent', 'people', 'use', 'become', 'actress', 'cast']
Topic 2: ['bad', 'scene', 'script', 'real', 'art', 'go', 'short', 'work', 'truly', 'screen']
Topic 3: ['act', 'even', 'drama', 'interesting', 'avoid', 'comedy', 'say', 'right', 'pretty', 'writer']
Topic 4: ['good', 'time', 'watch', 'waste', 'follow', 'find', 'lot', 'actually', 'definitely', 'whole']
Topic 5: ['recommend', 'funny', 'life', 'give', 'boring', 'fail', 'however', 'first', 'tell', 'cheap']
Topic 6: ['see', 'think', 'get', 'play', 'never', 'love', 'enjoy', 'year', 'music', 'seem']
Topic 7: ['make', 'great', 'well', 'ever', 'look', 'end', 'thing', 'like', 'hand', 'hole']
Topic 8: ['character', 'still', 'know', 'take', 'come', 'effect', 'little', 'fan', 'special', 'easy']
Topic 9: ['really', 'fast', 'dialogue', 'mess', 'keep', 'beautiful', 'fun', 'deliver', 'guy', 'piece']
****************************************************************************************************

 A very, very, very slow-moving, aimless movie about a distressed, drifting young man.  ['(0, 4.6%)', '(1, 4.5%)', '(2, 26.2%)', '(3, 5.7%)', '(4, 33.6%)', '(5, 2.
7%)', '(6, 7.7%)', '(7, 7.0%)', '(8, 4.9%)', '(9, 3.1%)']

 Not sure who was more lost - the flat characters or the audience, nearly half of whom walked out.  ['(0, 4.0%)', '(1, 4.0%)', '(2, 15.5%)', '(3, 5.0%)', '(4, 14.3%)',
'(5, 2.4%)', '(6, 14.3%)', '(7, 6.2%)', '(8, 31.5%)', '(9, 2.8%)']
```

Review	Topic	Probability
A very, very, very slow-moving, aimless movie ...	4	0.335568
Not sure who was more lost - the flat characte...	8	0.314510
Attempting artiness with black & white and cle...	3	0.292560
Very little music or anything to speak of.	8	0.263201
The best scene in the movie was when Gerardo i...	4	0.401369
...	...	...
I just got bored watching Jessice Lange take h...	8	0.370668
Unfortunately, any virtue in this film's produ...	2	0.405304
In a word, it is embarrassing.	2	0.345535
Exceptionally bad!	2	0.345540
All in all its an insult to one's intelligence...	4	0.332675



As you can see here**, the distribution of 10 topics is much better across all reviews with guided LDA.** And this is our final model.