

```

In [1]: import requests
from PIL import Image
from io import BytesIO
import requests
import json
import random
from PIL import Image,ImageDraw
import os
import pprint
import time
import urllib2
import pandas as pd
from datetime import date,timedelta,datetime
import re
import random
from PIL import ImageFilter
from datetime import datetime
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import gensim
from gensim import parsing, corpora
#os.environ['HTTP_PROXY']='iproxy.ln.ril.com:8080'
today =datetime.today().strftime('%Y%m%d')
headers = {
    'X-CleverTap-Account-Id': '877-958-874Z',
    'X-CleverTap-Passcode': 'SFQ-TEW-CTAL',
    'Content-Type': 'application/json',
}

In [2]: r2 = requests.get("http://cdnsrv.jio.com/jiotv.data.cdn.jio.com...{maskedAPI}.../get?os=android&devicetype=phone")

In [3]: chan={}
for item in r2.json()['result']:
    chan[item['channel_id']] = item['channel_name']

In [4]: corpus=[]
list1=[]
cols = ['showId','showname','keywords','description','episode_desc','showGenre','starCast','director','duration']
showKeywords=pd.DataFrame(columns = cols)
for i in chan:
    try:
        r = requests.get('http://cdnsrv.jio.com/jiotv.data.cdn.jio.com...{maskedAPI}.../get?offset=-1&channel_id=' + str(i)).json()
    except ValueError:
        print("No value for " + str(i))
    for item in r['epg']:
        list1.append([item['showId'],item['showname'],item['keywords'],item['description'],item['episode_desc'],item['showGenre'],item['starCas
t'],item['director'],item['duration']])
        corpus.append(item['keywords'])
showKeywords = pd.DataFrame(list1, columns=cols)

In [5]: flat_list = [item for sublist in corpus for item in sublist]

In [6]: corpus_of_keywords=corpus
#dictionary = corpora.Dictionary(corpus)

In [5]: showKeywords.shape
Out[5]: (24109, 9)

In [6]: showKeywords1=showKeywords.drop_duplicates(subset=['showId'], keep='first').reset_index()
showKeywords1=showKeywords1.drop(showKeywords1.columns[0], axis=1)
showKeywords1.head()

Out[6]:


|   | showId                     | showname               | keywords                                          | description                                       | episode_desc | showGenre | starCast | director        | duration  |
|---|----------------------------|------------------------|---------------------------------------------------|---------------------------------------------------|--------------|-----------|----------|-----------------|-----------|
| 0 | CHN-016800000PRG-307890000 | BBC World News         | [Stories, Top stories, Latest, Updates, Develo... | The host presents some of the most recent and ... |              |           |          | Richard Murrell | 30.000000 |
| 1 | CHN-016800000PRG-549891043 | Outside Source         | [Breaking News, Set in a newsroom, Social Medi... | The host presents some of the latest news upda... |              |           |          |                 | 30.000000 |
| 2 | CHN-016800000PRG-219150000 | HARDtalk               | [Personalities, Interview, People, World, Disc... | The host poses tough questions to the prominen... |              | [News]    |          | Lorna Dickinson | 30.000000 |
| 3 | CHN-016800000PRG-983970000 | BBC World News America | [News bulletin, Breaking News, Latest, Social ... | The host presents the latest social, economic,... |              |           |          |                 | 30.000000 |
| 4 | CHN-016800000PRG-180760000 | Asia Business Report   | [Business, Asia, Stories, Top stories, Latest,... | The anchor presents the latest business and ma... |              |           |          |                 | 15.000000 |



5 rows × 9 columns


In [7]: showKeywords2=showKeywords1.drop_duplicates(subset=['showname','description','episode_desc'], keep='first').reset_index()
showKeywords2=showKeywords2.drop(showKeywords2.columns[0], axis=1)
showKeywords2.head()

Out[7]:


|   | showId                     | showname               | keywords                                          | description                                       | episode_desc | showGenre | starCast | director        | duration  |
|---|----------------------------|------------------------|---------------------------------------------------|---------------------------------------------------|--------------|-----------|----------|-----------------|-----------|
| 0 | CHN-016800000PRG-307890000 | BBC World News         | [Stories, Top stories, Latest, Updates, Develo... | The host presents some of the most recent and ... |              |           |          | Richard Murrell | 30.000000 |
| 1 | CHN-016800000PRG-549891043 | Outside Source         | [Breaking News, Set in a newsroom, Social Medi... | The host presents some of the latest news upda... |              |           |          |                 | 30.000000 |
| 2 | CHN-016800000PRG-219150000 | HARDtalk               | [Personalities, Interview, People, World, Disc... | The host poses tough questions to the prominen... |              | [News]    |          | Lorna Dickinson | 30.000000 |
| 3 | CHN-016800000PRG-983970000 | BBC World News America | [News bulletin, Breaking News, Latest, Social ... | The host presents the latest social, economic,... |              |           |          |                 | 30.000000 |
| 4 | CHN-016800000PRG-180760000 | Asia Business Report   | [Business, Asia, Stories, Top stories, Latest,... | The anchor presents the latest business and ma... |              |           |          |                 | 15.000000 |



5 rows × 9 columns


In [ ]:

```

## This is the MetaData Approach

```

In [8]: #nospaceKeywords={}
showKeywords2[ 'nospaceKeywords']=showKeywords2[ 'keywords'].apply(lambda x: [i.replace(' ','') for i in x])
#for item in showKeywords:
#    nospaceKeywords[item] = [i.replace(' ','') for i in showKeywords[item]]
showKeywords2.head()

Out[8]:


|   | showId                     | showname               | keywords                                          | description                                       | episode_desc | showGenre | starCast | director        | duration  | nospaceKeywords                                   |
|---|----------------------------|------------------------|---------------------------------------------------|---------------------------------------------------|--------------|-----------|----------|-----------------|-----------|---------------------------------------------------|
| 0 | CHN-016800000PRG-307890000 | BBC World News         | [Stories, Top stories, Latest, Updates, Develo... | The host presents some of the most recent and ... |              |           |          | Richard Murrell | 30.000000 | [Stories, Topstories, Latest, Updates, Develop... |
| 1 | CHN-016800000PRG-549891043 | Outside Source         | [Breaking News, Set in a newsroom, Social Medi... | The host presents some of the latest news upda... |              |           |          |                 | 30.000000 | [BreakingNews, Setinnewsroom, SocialMedia, St...  |
| 2 | CHN-016800000PRG-219150000 | HARDtalk               | [Personalities, Interview, People, World, Disc... | The host poses tough questions to the prominen... |              | [News]    |          | Lorna Dickinson | 30.000000 | [Personalities, Interview, People, World, Disc... |
| 3 | CHN-016800000PRG-983970000 | BBC World News America | [News bulletin, Breaking News, Latest, Social ... | The host presents the latest social, economic,... |              |           |          |                 | 30.000000 | [Newsbulletin, BreakingNews, Latest, Socialiss... |


```

4	CHN-016800000PRG-180760000	Asia Business Report	[Business, Asia, Stories, Top stories, Latest,...]	The anchor presents the latest business and ma...		15.000000	[Business, Asia, Stories, Topstories, Latest,...]	
5 rows × 10 columns								
In [9]: showKeywords2['metadata'] = showKeywords2['nospacekeywords'].apply(' '.join) showKeywords2['metadata_spaced'] = showKeywords2['keywords'].apply(' '.join) #(pprint.pprint(l.encode('utf-8')) for l in df_nospaceshowKeywords['nospaceKeywords']) #df_nospaceshowKeywords['nospaceKeywords'].apply(' '.join) showKeywords2.head()								
Out[9]:								
0	016800000PRG-307890000	BBC World News	[Stories, Top stories, Latest, Updates, Develop...	The host presents some of the most recent and ...	Richard Murrell	30.000000	[Stories, Topstories, Latest, Updates, Develop...	
1	016800000PRG-549981043	Outside Source	[Breaking News, Set in a newsroom, Social Medi...	The host presents some of the latest news upda...		30.000000	[BreakingNews, Setinnewsroom, SocialMedia, St...	
2	016800000PRG-219150000	HARDtalk	[Personality, Interview, People, World, Disc...	The host poses tough questions to the prominen...	Lorna Dickinson	30.000000	[Personality, Interview, People, World, Disc...	
3	016800000PRG-983970000	BBC World News America	[News bulletin, Breaking News, Latest, Social ...	The host presents the latest social, economic...		30.000000	[Newsbulletin, BreakingNews, Latest, Socialissu...	
4	016800000PRG-180760000	Asia Business Report	[Business, Asia, Stories, Top stories, Latest,...]	The anchor presents the latest business and ma...		15.000000	[Business, Asia, Stories, Topstories, Latest, ...]	
5 rows × 12 columns								
In [ ]:								
In [ ]:								
In [10]: count_vec = CountVectorizer() count_vec_matrix = count_vec.fit_transform(showKeywords2['metadata']) cosine_sim_matrix = cosine_similarity(count_vec_matrix, count_vec_matrix) mapping = pd.Series(showKeywords2.index, index = showKeywords2['showId'])								
In [95]: showKidId="CHN-006930000PRG-549771701" movie_index = mapping@showKidId similarity_score = list(enumerate(cosine_sim_matrix[movie_index])) similarity_score = sorted(similarity_score, key=lambda x: x[1], reverse=True) similarity_score = similarity_score[0:15] movie_indices = [i[0] for i in similarity_score] showKeywords2.iloc[movie_indices]								
Out[95]:								
68	006930000PRG-549771701	The Avengers	[Adventure, Threat, Sci-fi, Superhero, Aliens ...]	Nick Fury is compelled to launch the Avengers ...	Robert Downey Jr, Chris Evans, Mark Ruffalo, C...	Joss Whedon	152.0	[Adventure, Threat, Sci-fi, Superhero, Aliens...]
677	001540000PRG-550531018	Ant-Man And The Wasp	[Superhero, Superhero movie, Based on a comic, ...]	Despite being under house arrest, Scott Lang, ...	[Action & Thriller] Paul Rudd, Evangeline Lilly, Michael Pena, Wal...	Peyton Reed	150.0	[Superhero, Superheromovie, Basedoncomic, Com...
5566	000930000PRG-550520013	DC Super Hero Girls: Legends of Atlantis	[Based on a comic, Comic book movies, Superher...	Mera and Siren steal the powerful Book of Leg...		Cecilia Aranovich	71.0	[Basedoncomic, Comicbookmovies, Superhero, Su...
9016	000911000PRG-550494933	The Death of Superman	[Based on a comic, Comic book movies, End of t...	Doomsday, a monstrous humanoid, emerges from a...	[Action & Thriller] Nathan Fillion, Rann Wilson	Sam Liu	202.0	[Basedoncomic, Comicbookmovies, EndoftheWorld...]
9021	000911000PRG-550560795	Aquaman	[Underwater, Epic fantasy, Set in a fantasy wo...	Half-human, half-Atlantean Arthur is born with...	[Action & Thriller] Jason Momoa, Amber Heard, Willem Dafoe, Patrick...	James Wan	161.0	[Underwater, Epicfantasy, Setinfantasyworld, ...]
9024	000911000PRG-549650843	The Dark Knight	[Superhero, Terrorism, Vigilante, Psychopath, ...]	After Gordon, Dent and Batman begin an assault...	[Action & Thriller] Christian Bale, Michael Caine, Heath Ledger, G...	Christopher Nolan	154.0	[Superhero, Terrorism, Vigilante, Psychopath, ...]
879	001310000PRG-550352132	Spider-Man: Homecoming	[Mutant, Good versus evil, Evil intentions, Tr...	Peter Parker tries to stop the Vulture from se...	[Action & Thriller] Tom Holland, Michael Keaton, Jon Favreau, Zend...	Jon Watts	180.0	[Mutant, Goodversusevil, Evilintentions, Trium...
561	004280000PRG-550098302	Shiva	[Superpower, Superhero, Suitable for children ...]	Shiva is a brave and intelligent child who sta...		Suhas Kadav	65.0	[Superpower, Superhero, Suitableforchildren...]
6820	006211000PRG-550059056	The Flash	[Crime, Speed, Evil, Superhero, Accident, Supe...	Barry Allen, a forensic scientist with the Cen...	[Action & Thriller] Grant Gustin, Candice Patton, Danielle Panabak...	Dermott Downs	60.0	[Crime, Speed, Evil, Superhero, Accident, Supe...
5557	000911000PRG-550616489	Shazam!	[Superhero, Superhero movie, Superpower, Aband...	After being abandoned at a fair, Billy constant...	[Action & Thriller] Asher Angel, Zachary Levi, Mark Strong, Jack D...	David F Sandberg	137.0	[Superhero, Superheromovie, Superpower, Abando...
7918	013300000PRG-549695588	Meet Dave	[Love, Aliens and extraterrestrials, Spaceship...]	A spaceship which is shaped like a human and n...	[Comedy] Eddie Murphy, Elizabeth Banks, Gabrielle Union...	Brian Robbins	95.0	[Love, Aliensandextraterrestrials, Spaceship, ...]
9025	000930000PRG-549642612	Batman	[Superhero, Superhero movie, Based on a comic, ...]	Batman, a masked vigilante from Gotham City, f...	[Action & Thriller] Michael Keaton, Jack Nicholson, Kim Basinger, ...	Tim Burton	36.0	[Superhero, Superheromovie, Basedoncomic, Com...
65	006930000PRG-549915675	The Amazing Spider-Man 2	[Superhero, Vengeance, Childhood, Crime, Sci-f...	Spider-Man embarks on a mission to protect his...	[Action & Thriller] Andrew Garfield, Emma Stone, Jamie Foxx, Dane ...	Marc Webb	140.0	[Superhero, Vengeance, Childhood, Crime, Sci-f...
7693	001540000PRG-	Thor: Ragnarok	[Superhero, movie, Mythological, Deprived of his mighty hammer]	Chris Hemsworth, Tom Hiddleston	[Action & Thriller] Taika Waititi	150.0	[Superheromovie, Mythological,	



```
(5811, 0.5771160371203719),
(5438, 0.5698028822981898),
(4263, 0.5345224838248487),
(4641, 0.5345224838248487),
(5256, 0.5345224838248487),
(5912, 0.5345224838248487)]
```

```
In [61]: showKeywords2.to_csv("C:\Users\Rahul17.Jain\Downloads\RahulReports\Code\for-keywords.csv",sep=',', encoding='utf-8')
```

```
In [ ]:
```

```
In [ ]:
```

## This is TFIDF Description Clustering Approach

```
In [20]: from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.cluster import MiniBatchKMeans
import numpy as np
import pandas as pd
import csv
from sklearn.metrics import pairwise_distances_argmin_min
from sklearn.decomposition import PCA
```

```
In [21]: descriptions=showKeywords2['description'][ : ].tolist()
descriptions.head(6)
```

```
Out[21]: [u'An interesting and witty collection of personal stories of four filmmakers from around the world that showcases their day-to-day lives under lockdown.',
u'The host presents some of the most recent and significant political and economic news stories and updates from around the world.',
u'Experts give investment advice to the viewers about the growing trends of the stock market.',
u'Young achievers, who are poised to become future leaders, talk about their past experiences, the challenges they faced and their vision for the future.',
u'The host speaks to highly regarded figures in advertising and broadcasting about the dynamic landscape and issues of media in India.',
u'Talented contestants, belonging to different generations, showcase their dancing skills and use different styles to impress the judges and win the coveted title.'
...]
```

```
In [22]: #descriptions=showKeywords2['description'][ : ].tolist()
descriptions=showKeywords2[showKeywords2['description']!=''][ 'description'].tolist()
tfidfvec = TfidfVectorizer(stop_words='english')
X = tfidfvec.fit_transform(descriptions)

first_vector = X[0]

dataframe = pd.DataFrame(first_vector.T.todense(), index = tfidfvec.get_feature_names(), columns = ["tfidf"])
dataframe.sort_values(by = ["tfidf"], ascending=False).head()
```

```
Out[22]:
```

```
tfidf
day 0.427477
witty 0.400888
filmmakers 0.381775
lockdown 0.322723
showcases 0.316429
```

```
9417 rows x 1 columns
```

```
In [23]: num = 4
kmeans1 = KMeans(n_clusters = num, init = 'k-means++', max_iter = 5000, n_init = 1)
kmeans1.fit(X)
print(kmeans1.cluster_centers_) #This will print cluster centroids as tf-idf vectors
```

```
array([[5.56594988e-05, 4.92459817e-04, 1.1020660e-04, ...,
       6.96259844e-05, 6.10273653e-05, 6.117877706e-05],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
       0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
       0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
       0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
       0.00000000e+00, 0.00000000e+00, 0.00000000e+00]])
```

```
In [25]: Y = tfidfvec.transform(['In a divine discourse with Radha, Lord Krishna takes it upon himself to explain texts from the ancient and fabled Sr imad Bhagwat, sacred in Hinduism, which offers mankind profound spiritual knowledge.'])
predicted = kmeans1.predict(Y)
print(predicted)
```

```
[0]
```

```
In [26]: closest, _ = pairwise_distances_argmin_min(kmeans1.cluster_centers_, X)
closest
```

```
Out[26]: array([3122, 3122, 194, 350], dtype=int64)
```

```
In [28]: descriptions[0]
```

```
Out[28]: u'An interesting and witty collection of personal stories of four filmmakers from around the world that showcases their day-to-day lives under lockdown.'
```

```
In [29]: from sklearn.metrics import silhouette_score
```

```
sil = []
kmax = 20

# dissimilarity would not be defined for a single cluster, thus, minimum number of clusters should be 2
for k in range(2, kmax+1):
    kmeans12345 = KMeans(n_clusters = k).fit(X)
    labels = kmeans12345.labels_
    sil.append(silhouette_score(X, labels, metric = 'euclidean'))
```

```
In [30]: sil
```

```
Out[30]: [0.012795111421886567,
0.010398223690967115,
0.03203162020426666,
0.007916484643250008,
0.025925164223181106,
0.03303566908497246,
0.024411614579020077,
0.0313414189144309,
0.02668992715324961,
0.0322766835313412,
0.026882521040593875,
0.028510835155329803,
0.029955498873017728,
0.03378457530309911,
0.03248199152539941,
0.040430176300584242,
0.03316363799267441,
```

```
0.04058160573191141,  
0.04125776593872568]
```

```
In [ ]:
```

## This is LDA Gensim Approach

```
In [30]: from gensim import corpora, models
```

```
In [62]: dictionary_LDA = corpora.Dictionary(showKeywords2['keywords'])  
dictionary_LDA.filter_extremes(no_below=3)  
corpus = [dictionary_LDA.doc2bow(list_of_tokens) for list_of_tokens in showKeywords2['keywords']]  
  
num_topics = 10  
#time lda_model = models.LdaModel(corpus, num_topics=num_topics, \  
#                                     id2word=dictionary_LDA, \  
#                                     passes=4, alpha=[0.01]*num_topics, \  
#                                     eta=[0.01]*len(dictionary_LDA.keys()))
```

```
Wall time: 11.8 s
```

```
In [201]: for i,topic in lda_model.show_topics(formatted=True, num_topics=num_topics, num_words=10):  
    print(str(i)+": "+ topic)  
    print()
```

```
0: 0.075*"Entertainment" + 0.052*"Actor" + 0.051*"Celebrity" + 0.051*"Movie" + 0.037*"Host" + 0.032*"Bollywood" + 0.030*"Videos" + 0.022*"Historical significance" + 0.022*"Inspiration" + 0.021*"Father"  
(  
1: 0.173*"Expert opinion" + 0.137*"Expert analysis" + 0.111*"Discussions" + 0.051*"Social Issues" + 0.049*"Panel Discussion" + 0.041*"Host" + 0.024*"Debate" + 0.022*"Talk Show" + 0.020*"Politics" + 0.020*"Interactive session"  
(  
2: 0.134*"Politician" + 0.100*"Healthcare" + 0.070*"Political party" + 0.066*"Health" + 0.059*"Politics" + 0.044*"Healthy diet" + 0.032*"Wellness" + 0.032*"Doctor" + 0.031*"Yoga" + 0.029*"Treatment"  
(  
3: 0.152*"Updates" + 0.118*"Latest" + 0.105*"Developments" + 0.081*"Top stories" + 0.068*"Breaking News" + 0.067*"News headlines" + 0.061*"Stories" + 0.057*"Current Affairs" + 0.038*"Events" + 0.031*"Host"  
(  
4: 0.089*"Movie" + 0.076*"Compilation" + 0.068*"Funny videos" + 0.049*"Sequences" + 0.047*"Collection" + 0.040*"Comedian" + 0.038*"Actor" + 0.033*"Real-Life" + 0.027*"Entertainment" + 0.025*"Highlights"  
(  
5: 0.060*"Family" + 0.059*"Love" + 0.056*"Couple in love" + 0.042*"Family Drama" + 0.037*"Friendship" + 0.036*"Lovers" + 0.035*"Relationship" + 0.026*"Love-struck" + 0.024*"Family Feud" + 0.022*"Adventure"  
(  
6: 0.153*"Songs" + 0.086*"Compilation" + 0.064*"Popular" + 0.063*"Musician" + 0.059*"Singer" + 0.044*"Musical" + 0.041*"Artist" + 0.038*"Music chart" + 0.037*"Singing" + 0.037*"Music"  
(  
7: 0.086*"Hymn" + 0.069*"Devotion" + 0.051*"Prayer" + 0.045*"Ritual" + 0.045*"Devotees" + 0.037*"Praising" + 0.037*"God" + 0.037*"Worship" + 0.029*"Goddess" + 0.027*"Songs"  
(  
8: 0.135*"Politics" + 0.111*"Social Issues" + 0.105*"Updates" + 0.076*"Economy" + 0.071*"Economic" + 0.064*"Developments" + 0.059*"Latest" + 0.052*"Host" + 0.031*"Sports" + 0.030*"Society"  
(  
9: 0.055*"Criminal" + 0.044*"Criminal offence" + 0.038*"Police" + 0.037*"Murder" + 0.033*"Revenge" + 0.022*"Criminal mastermind" + 0.022*"Revenge saga" + 0.019*"Deity" + 0.017*"Action sequences" + 0.017*"Team"  
(  
10: 0.104*"India" + 0.093*"National" + 0.064*"People" + 0.062*"International" + 0.061*"Culture" + 0.046*"Government" + 0.041*"Regional" + 0.040*"Lifestyle" + 0.038*"Local culture" + 0.036*"Locals"  
(  
11: 0.091*"Bible" + 0.056*"Temple" + 0.054*"Priest" + 0.049*"Traveller" + 0.042*"Travel destination" + 0.031*"Travel" + 0.026*"God" + 0.026*"Lord" + 0.025*"God dess" + 0.023*"Customs and Traditions"  
(  
12: 0.080*"Business" + 0.072*"Stock Market" + 0.058*"Crime" + 0.051*"Set in the film industry" + 0.050*"Investigation" + 0.046*"Share Market" + 0.046*"Market" + 0.038*"Finance" + 0.032*"Public opinion" + 0.031*"Market predictions"  
(  
13: 0.182*"News headlines" + 0.160*"Breaking News" + 0.146*"Set in a newsroom" + 0.107*"News" + 0.102*"News bulletin" + 0.094*"Updates" + 0.040*"Current Affairs" + 0.025*"Newspaper" + 0.019*"Information" + 0.017*"Developments"  
(  
14: 0.154*"Christian" + 0.122*"Social Awareness" + 0.082*"Reporter" + 0.069*"Awareness" + 0.053*"Village" + 0.050*"Information" + 0.042*"Church" + 0.040*"Education" + 0.037*"Educational" + 0.037*"Villagers"  
(  
15: 0.135*"Spirituality" + 0.123*"Spiritual" + 0.104*"Spiritual Leader" + 0.074*"Religious discourse" + 0.070*"Religious leader" + 0.063*"Religion" + 0.047*"Religious verses" + 0.040*"Religious" + 0.035*"Devotees" + 0.032*"Devotion"  
(  
16: 0.150*"Religious leader" + 0.142*"Religious verses" + 0.128*"Religious discourse" + 0.095*"Religion" + 0.062*"Religious" + 0.054*"Hinduism" + 0.051*"Devotion" + 0.050*"Devotees" + 0.045*"Hindu" + 0.025*"Hindu mythology"  
(  
17: 0.124*"Christianity" + 0.111*"Jesus Christ" + 0.048*"Preaching" + 0.047*"Islam" + 0.040*"Competition" + 0.038*"Teachings" + 0.030*"Contest" + 0.029*"Opinion" + 0.027*"Contestant" + 0.023*"Holy Quran"  
(  
18: 0.041*"Marriage" + 0.032*"Interaction" + 0.031*"Personalities" + 0.030*"Family" + 0.029*"Family Drama" + 0.028*"Gossip" + 0.025*"Married couple" + 0.024*"Life journey" + 0.023*"Professional" + 0.023*"Host"  
(  
19: 0.062*"Performance" + 0.053*"Sense of humour" + 0.049*"Performer" + 0.049*"Solutions" + 0.048*"Viewers" + 0.047*"Comedian" + 0.042*"Women-oriented" + 0.039*"Advice" + 0.036*"Problems" + 0.034*"Comedy-of-Errors"  
(  
)
```

```
In [63]: for i,topic in lda_model.show_topics(formatted=True, num_topics=num_topics, num_words=10):  
    print(str(i)+": "+ topic)  
    print()
```

```
0: 0.066*"Expert opinion" + 0.045*"Expert analysis" + 0.043*"Healthcare" + 0.031*"Personalities" + 0.029*"Health" + 0.024*"Food" + 0.023*"Host" + 0.020*"Doctor" + 0.020*"Tips" + 0.019*"Recipe"  
(  
1: 0.050*"Culture" + 0.030*"Local culture" + 0.028*"History" + 0.028*"India" + 0.027*"Historical significance" + 0.026*"Mental health" + 0.023*"Products" + 0.020*"Traveller" + 0.019*"Explorer" + 0.018*"Travel destination"  
(  
2: 0.107*"Christianity" + 0.092*"Christian" + 0.086*"Jesus Christ" + 0.041*"Religious leader" + 0.040*"Religious discourse" + 0.030*"Religious verses" + 0.022*"Islam" + 0.018*"Farming" + 0.017*"Farmer" + 0.016*"Life lesson"  
(  
3: 0.116*"Songs" + 0.075*"Compilation" + 0.057*"Musician" + 0.055*"Singer" + 0.045*"Popular" + 0.042*"Artist" + 0.039*"Musical" + 0.039*"Bible" + 0.037*"Music" + 0.037*"Movie"  
(  
4: 0.039*"Problems" + 0.020*"Predictions" + 0.017*"Friendship" + 0.017*"Children" + 0.017*"Future" + 0.017*"Animals" + 0.016*"Adventure" + 0.016*"Women-oriented" + 0.016*"Challenge" + 0.015*"Astrology"  
(  
5: 0.041*"Celebrity" + 0.039*"Entertainment" + 0.037*"Interaction" + 0.037*"Performance" + 0.031*"Actor" + 0.031*"Interactive session" + 0.030*"Sense of humour" + 0.028*"Comedian" + 0.028*"Performer" + 0.027*"Movie"  
(  
6: 0.056*"Love" + 0.040*"Couple in love" + 0.032*"Family" + 0.031*"Lovers" + 0.025*"Family Drama" + 0.025*"Marriage" + 0.024*"Drama" + 0.022*"Criminal" + 0.022*"Love-struck" + 0.021*"Police"  
(  
7: 0.091*"Religious leader" + 0.090*"Religious discourse" + 0.081*"Religious verses" + 0.074*"Religion" + 0.057*"Spirituality" + 0.055*"Religious" + 0.054*"Devotion" + 0.052*"Devotees" + 0.048*"Spiritual" + 0.044*"Spiritual Leader"  
(  
8: 0.069*"Politics" + 0.067*"Social Issues" + 0.058*"Host" + 0.053*"Updates" + 0.045*"Discussions" + 0.034*"Developments" + 0.033*"Politician" + 0.033*"Expert opinion" + 0.032*"Economic" + 0.030*"Economy"  
(  
9: 0.117*"Updates" + 0.095*"News headlines" + 0.086*"Breaking News" + 0.059*"Developments" + 0.059*"Latest" + 0.050*"Set in a newsroom" + 0.043*"Current Affairs" + 0.043*"News" + 0.039*"News bulletin" + 0.038*"Top stories"  
(  
)
```

```
In [33]: lda_model[corpus[166]]
Out[33]: [(3, 0.08777636), (8, 0.90561104)]
```

```
In [65]: showKeywords2.index[showKeywords2['showId']=="CHN-007460000PRG-406870000"].tolist()[0]
# pprint.pprint(showKeywords2['keywords'][1044])
Out[65]: Int64Index([36], dtype='int64')
```

```
In [35]: for item in lda_model[corpus[9]]:
    print(item[1])
    if item[1]>0.25:
        print("1234")
        topic_list.append(item)
topic_list
```

```
0_29029936
1234
```

```
NameError                                 Traceback (most recent call last)
<ipython-input-35-347347751e48> in <module>()
      3     if item[1]>=0.25:
      4         print("1234")
----> 5     topic_list.append(item)
      6 topic_list

NameError: name 'topic_list' is not defined
```

```
In [66]: showKeywords3=showKeywords2
showKeywords3['topic']=""
showKeywords3['topic'].astype(object)
for index in range(0,len(showKeywords3)):
    topic_list=[]
    for topic,probability in lda_model[corpus[index]]:
        if probability>=0.25:
            topic_list.append((topic,probability))
    showKeywords3.at[index,'topic']=topic_list
showKeywords3.head()
```

```
Out[66]:
   showid showname      keywords      description episode_desc showGenre starCast   director duration nospacekeywords      metadata      metadata_spaced      topic
0  CHN-016800000PRG-550720403  The Lockdown Diaries [Videos, Filmmaker, The realities of life, Wit... An interesting and witty collection of persona... [] 30.000000 [Videos, Filmmaker, The realities of life, Wit, ... Videos Filmmaker The realities of life Wit World [1, 0.620022], (3, 0.36429065)]
1  CHN-016800000PRG-307890000  BBC World News [Stories, Top stories, Latest Updates, Develop... The host presents some of the most recent and ... [] Richard Murrell 30.000000 [Stories, Topstories, Latest, Updates, Develop... Stories Topstories Latest Updates Developments.. Stories Top stories Latest Updates Developments.. [[9, 0.9852458]]
2  CHN-016800000PRG-549672872       Click [Technology, Gadgets, Consumer, Developments, ... The host presents the diverse key developments... [] 30.000000 [Technology, Gadgets, Consumer, Developments, ... Technology Gadgets Consumer Developments Business Technology Gadgets Consumer Developments Business [[1, 0.27112338], (4, 0.34358868), (9, 0.37156...]
3  CHN-016800000PRG-219260000  Dateline London [Talk Show, News, Breaking News, News bulletin,... Foreign correspondents posted to London look a... [News] 30.000000 [TalkShow, News, BreakingNews, Newsbulletin, N... TalkShow News BreakingNews Newsbulletin Newshe... Talk Show News Breaking News News bulletin New... [[9, 0.8133304]]
4  CHN-016800000PRG-550151456  Global Questions [Social Awareness, Social commentary, Social ... Audiences from around the world pose important... The audience put their questions to a panel of... [News] 30.000000 [SocialAwareness, Socialcommentary, Socialissu... SocialAwareness Socialcommentary Socialissues ... Social Awareness Social commentary Social Issu... [[8, 0.70266795]]
```

5 rows x 13 columns

```
In [97]: showKeywords3.iloc[166]
```

```
Out[97]:
showId          CHN-007740000PRG-549635823
showname        Hum Aapke Hain Koun...
keywords        [Marriage, Sacrifice, Relationship, Siblings, ...
description    During his brother Rajesh's wedding, Prem meet...
episode_desc   []
starCast        Madhuri Dixit, Salman Khan, Mohnish Bahl, Renu...
director       Sooraj Barjatya
duration        240
nospacekeywords [Marriage, Sacrifice, Relationship, Siblings, ...
metadata       Marriage Sacrifice Relationship Siblings Famil...
metadata_spaced Marriage Sacrifice Relationship Siblings Famil...
topic           [(6, 0.8121164)]
```

```
Name: 166, dtype: object
```

```
In [38]: #corpus = [dictionary_LDA.doc2bow(list_of_tokens) for list_of_tokens in [u'Lovers',u'True Love',u'Romance',u'Love at first sight',u'Couple in love',u'Drama',u'Family Drama',u'Love',u'Love-struck']]
corpus
```

```
Out[38]: [((), 1), ((1, 1), (2, 1), (3, 1), (4, 1)), ((5, 1), (6, 1), (7, 1), (8, 1), (9, 1), (10, 1)), ((6, 1), (11, 1), (12, 1), (13, 1), (14, 1)), ((15, 1), (16, 1), (17, 1), (18, 1), (19, 1), (20, 1)), ((21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1)), ((20, 1), (28, 1), (29, 1), (30, 1), (31, 1), (32, 1), (33, 1), (34, 1), (35, 1), (36, 1), (37, 1), (38, 1)), ((10, 1), (17, 1), (22, 1), (39, 1), (40, 1), (41, 1), (42, 1), (43, 1)), ((11, 1), (44, 1), (45, 1), (46, 1), (47, 1)), ((48, 1), (49, 1), (50, 1), (51, 1), (52, 1), (53, 1), (54, 1), (55, 1)), ((51, 1), (56, 1), (57, 1), (58, 1), (59, 1)), ((5, 1), (7, 1), (10, 1), (22, 1), (35, 1)), ((5, 1), (10, 1), (17, 1), (35, 1), (60, 1)), ((4, 1), (5, 1), (17, 1), (61, 1), (62, 1), (63, 1)), ((11, 1), (35, 1), (62, 1), (64, 1), (65, 1), (66, 1)), ((33, 1), (62, 1), (67, 1), (68, 1), (69, 1), (70, 1)), ((11, 1), (21, 1), (32, 1), (33, 1), (71, 1), (72, 1)), ((51, 1), (54, 1), (55, 1), (59, 1), (73, 1), (74, 1), (75, 1), (76, 1)), ((21, 1), (45, 1), (62, 1), (66, 1), (77, 1), (78, 1)), ((5, 1), (7, 1), (10, 1), (17, 1), (21, 1), (79, 1)), ((62, 1), (80, 1), (81, 1), (82, 1), (83, 1)), [], ((70, 1), (82, 1)), ((32, 1), (33, 1), (71, 1), (83, 1), (84, 1), (85, 1), (86, 1), (87, 1)), ((5, 1), (88, 1), (89, 1), (90, 1), (91, 1)), ((21, 1), (33, 1), (62, 1), (92, 1), (93, 1))]
```

```

[(20, 1), (32, 1), (33, 1), (71, 1), (84, 1), (94, 1), (95, 1)],
[(21, 1),
(46, 1),
(77, 1),
(96, 1),
(97, 1),
(98, 1),
(99, 1),
(100, 1),
(101, 1)],
[(21, 1), (32, 1), (33, 1), (70, 1), (102, 1), (103, 1)],
[(69, 1), (70, 1), (104, 1), (105, 1), (106, 1)],
[(21, 1),
(30, 1),
(32, 1),
(33, 1),
(57, 1),
(91, 1),
(93, 1),
(103, 1),
(107, 1),
(108, 1),
(109, 1)],
[(110, 1), (111, 1), (112, 1), (113, 1), (114, 1)],
[(110, 1), (115, 1), (116, 1), (117, 1)],
[(118, 1), (119, 1), (120, 1), (121, 1), (122, 1)],
[(123, 1), (124, 1), (125, 1), (126, 1), (127, 1), (128, 1)],
[(120, 1), (129, 1), (130, 1), (131, 1), (132, 1), (133, 1), (134, 1)],
[(119, 1), (120, 1), (121, 1), (132, 1), (135, 1), (136, 1), (137, 1)],
[(138, 1), (139, 1), (140, 1), (141, 1), (142, 1), (143, 1)],
[(4, 1), (57, 1), (144, 1), (145, 1), (146, 1), (147, 1)]
...]

```

```

In [98]: from gensim import corpora, models, similarities
from itertools import chain

lda_corpus = lda_model[corpus]
scores = list(chain(*[[score for topic_id, score in topic] \
                     for topic in [doc for doc in lda_corpus]]))
threshold = sum(scores)/len(scores)
print threshold
print

cluster1 = [j for i, j in zip(lda_corpus, showKeywords2['showname']) if i[0][1] > threshold]
#cluster2 = [j for i, j in zip(lda_corpus, showKeywords2['showname']) if i[1][1] > threshold]
#cluster3 = [j for i, j in zip(lda_corpus, showKeywords2['showname']) if i[2][1] > threshold]

print cluster1
#print cluster2
#print cluster3

```

0.2820207960937817

```

[u'The Lockdown Diaries', u'BBC World News', u'The Travel Show', u'Russia - The Empire Strikes Back', u'Panorama', u'Newsnight', u'Our World', u'Storyboard', u'Markets In Turmoil', u'Conversation', u'Debate', u'American Greed', u'Wizards Of Dalal Street', u'The Global Eye', u'Young Turks', u'Know Your Fund', u'Commodity Champions', u'Khatra Khatra Khatra', u'Shrimad Bhagwat Mahapuran', u'Dard-E-Bil', u'Mahabharat', u'How Sex Changed The World', u'Storage Wars', u'The Curse of Oak Island', u'Pawn Stars', u'King Of Restoration', u'Dynamo: Magician Impossible', u'Crazy Wheels', u'Counting Cars', u'Meenakshi Amman & The Marvel Of Madurai', u'OMG! Yeh Meri India', u'Modern Marvels-Mega Countdown', u'Food Tech', u'Meet The Spartans', u'Land Of The Bears', u'The Pink Panther Strikes Again', u'Jumanji: Welcome To The Jungle', u'The Amazing Spider-Man 2', u'The Star', u'Aladdin', u'The Avengers', u'Maachicha Chulli', u'Bandishala', u'The Silence', u'Patanjali Yog', u'Mai Mandir', u'Gajra Kirtanacha, Sohala Anandacha', u'Man Mandira Vishesh', u'Andya Cha Funda', u'Poshter Boyz', u'Aladdin - Naam Toh Suna Hoga', u'Jijaji Chhat Per Hain', u'Waah Waah Kya Baat Hai', u'The Blue Revolution', u'Turf Wars', u'Great Centuries', u'Men In Blue Victorious', u'Arena - E Sports 2020', u'Nail-Biting Finishes', u'Savdhaan India : India Fights Back', u'Ramayan', u'Pyar Ka Dard Hai Meetha Meetha Pyara Pya', u'Tu Mera Hero', u'IPL 2012 HLS', u'ICC WC 2007 T20I', u'VIVO IPL Heroes 2019', u'IND vs AUS 2020 ODI HLS', u'Cricketer Connected 2020', u'Ind vs Aus 2019 ODI HLS', u'VIVO IPL 2019 HLS', u'IPL 2014 HLS', u'Titan Cup 1996 ODI', u'Ind vs Aus 2013 ODI HLS', u'IND vs AUS 2017 ODI HLS', u'Vivo PKL 2019 HLS', u'Best of UCL', u'UCL 2019/20', u'NBA 2019/20 HLS', u'NBA 2K Players Tournament', u'NBA2K Sundays With Thibaut Courtois', u'WWE SmackDown', u'Undertaker Special', u'Backpack', u'World Spas', u'Food Fact Fun', u'Off The Grid', u'Xplore Belgium', u'Unwind', u'Quest', u'City Breaks', u'Landmarks', u'Kissed By The Sea', u'Batti Gul Meter Chalu', u'RamaIya VasavaIya', u'Jai Simha', u'Kesari', u'Dream Girl', u'Amavas', u'Zee Rishtey Awards 2019', u'Paramavat Shri Krishna', u'News', u'Advertisement', u'Min 25 Khaabar', u'Nonstop 100', u'Subah Subah', u'Aapke Taare', u'10 Minute 50 Khabar', u'Shatak Ajatkar', u'Virat Test', u'Saas, Bahu Aur Betiyan', u'Dharm', u'Dangal', u'Halla Bol', u'Desh Tak', u'Khabardar', u'Kahani', u'Acharya Avdheshanand Giriji Maharaj', u'Aastha Special', u'Dhyan Yog', u'Yoga', u'Khel Khel Me Yog', u'Sadhguru Jaggi Vasudeviji', u'Shradhey Gaurishi Swami Datsarnanand Ji Maharaj', u'Seva Ganga', u'Yog Shivir', u'PrakrutiChikitsa', u'Gurudev Suresh Shrimali', u'Pt Jayaprakash Sharma', u'Dr. Ramavtaar - Jaha Jaha Charan Padhe Raghuvir Ke, u'Khabar Din Bhar', u'Fatafat', u'Badi Khabar', u'News Center', u'Aadha Din, Poori Khabar', u'Khabar Ab Tak', u'Saas Bahu Aur Saazish', u'Corona Control Top 100', u'Special Report', u'News 7', u'9 PM Report', u'Sansani', u'Manithali', u'Addheham Enna Iddeham', u'Play Back', u'Autograph - The Motor Show', u'16th Asianet Film Awards 2014', u'Veendum Chila Veettu Visheshangal', u'Chirikkum Thali ka', u'Farz', u'Beti No. 1', u'Baashshah', u'Woh Lamhe', u'Loop', u'Solid Hits', u'Bollywood Bang Bang', u'Bajao', u'Aawaaz De Kahan Hai', u'Boom Box', u'Morning Masala', u'Sooper Hits', u'Roots', u'Dip In Kitchen', u'Costar', u'Kadak', u'Non Stop Hits', u'Full Volume', u'Hitz', u'A-X-L', u'Phir Hera Pheri', u'Kedarnath', u'Snake In The Eagle's Shadow', u'Nagaichuvali Galatta', u'Thirai Thendral', u'Sugamana Ragangal', u'Bhakthi Padalgal', u'Aalaya Vazhipadu', u'Malla Kaalam Pirakkuthu', u'Diveega Dharisanam', u'Star Special', u'Alavudeen', u'Champions', u'Overdrive'
...]

```

```

In [42]: for i in range(0, len(dictionary_LDA)):
    print(str(i)+ " : "+ dictionary_LDA[i])

```

```

0 : Filmmaker
1 : The realities of life
2 : Videos
3 : Wit
4 : World
5 : Current Affairs
6 : Developments
7 : Latest
8 : Stories
9 : Top stories
10 : Updates
11 : Business
12 : Consumer
13 : Gadgets
14 : Technology
15 : Breaking News
16 : Events
17 : News
18 : News bulletin
19 : News headlines
20 : Talk Show
...

```

```

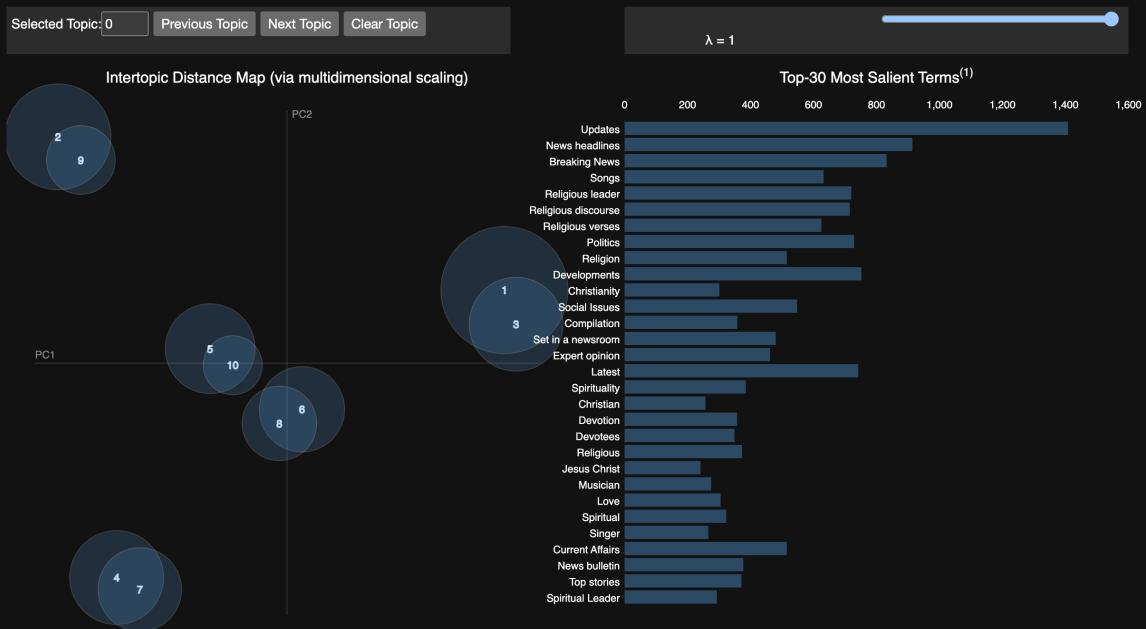
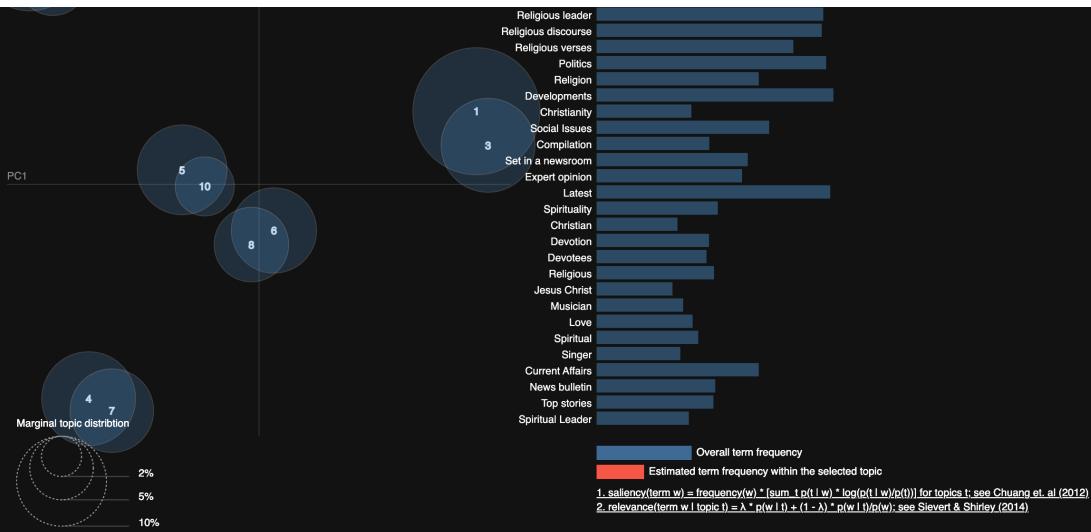
In [99]: %matplotlib inline
import pyLDAvis
import pyLDAvis.gensim
vis = pyLDAvis.gensim.prepare(topic_model=lda_model, corpus=corpus, dictionary=dictionary_LDA)
pyLDAvis.enable_notebook()
pyLDAvis.display(vis)

```

Out[99]: Selected Topic: 0 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2) 0.0 0.2 0.4 0.6 0.8 1  
Slide to adjust relevance metric:(2) 0.0 0.2 0.4 0.6 0.8 1





```
In [ ]: 
```

```
In [46]: from gensim.models.coherencemodel import CoherenceModel
cm=CoherenceModel(model=lda_model, texts=showKeywords2['keywords'], dictionary=dictionary_LDA, coherence='c_v')
cm
```

```
Out[46]: <gensim.models.coherencemodel.CoherenceModel at 0x13547fd0>
```

```
In [47]: print cm.get_coherence()
0.5451826341982785
```

```
In [48]: from gensim.models import CoherenceModel, LdaModel, LsiModel, HdpModel
hdpmodel = HdpModel(corpus=corpus, id2word=dictionary_LDA)
hdpmodel.show_topics()
```

```
Out[48]: [(0,
  u'0.006*Writer + 0.006*Magic + 0.005*Unhealthy obsession + 0.005*Small Town + 0.004*Violation of rights + 0.004*Cooking + 0.004*Engineering + 0.004*Grooming
  tips + 0.004*Christian + 0.004*Patriot + 0.004*Terrorist + 0.004*Wit + 0.004*Kidnapping + 0.004*Ramayana + 0.004>New Arrivals + 0.004*Biography + 0.004*Good ve
  rsus evil + 0.004*Defend a friend + 0.004*Extraordinary + 0.004*Action'),
 (1,
  u'0.006*Chaos + 0.006*Latest trends + 0.005*Action + 0.005*Health + 0.005*Holiday + 0.005*Government + 0.004*Community + 0.004*Devotion + 0.004*Hospital + 0.
  004*Trend + 0.004*Technology + 0.004*Intellectual + 0.004*Virus outbreak + 0.003*False Charge + 0.003*Warrior + 0.003*Fasting + 0.003*Map + 0.003*Ritual + 0.00
  3*Guest + 0.003*Machine'),
 (2,
  u'0.007*Lookalike + 0.006>Showbiz dramas + 0.005*Secret + 0.005*Positive quality + 0.004*Grandparents + 0.004*Spirit + 0.004*Music instruments + 0.004*Detect
  ive + 0.004*Adventure on the seas + 0.004*Sufi + 0.004*Doctor + 0.004*ODI + 0.004*Sermon + 0.004*Krishna + 0.004*Poet + 0.004*Sci-fi + 0.003*Motivational speec
  h + 0.003*Vision + 0.003*Poverty + 0.003*Preaching'),
 (3,
  u'0.005*Author + 0.005*Saint + 0.005*Conversation + 0.004*Religious verses + 0.004*Metropolitan + 0.004*Call-in + 0.004*Variety programme + 0.004*Heist + 0.
  004*Strategy + 0.004*Gulf + 0.004*Compilation + 0.004*Outlaw + 0.004*Crime + 0.003*Forest + 0.003*Set in Kolkata + 0.003*Europe + 0.003*Retired + 0.003*Failed S
  uicide Attempt + 0.003*Physical training + 0.003*Train'),
 (4,
  u'0.005*Genie + 0.005*Criminal offence + 0.004*Hunting + 0.004*Philosopher + 0.004*Corruption + 0.004*Animation + 0.004*Unlikely bond + 0.004*Paranormal + 0.
  004*Opposite + 0.004*Factory + 0.004*Heart condition + 0.003*War + 0.003*Virus outbreak + 0.003*Origin + 0.003*Daughter-in-law + 0.003*Unrequited Love + 0.003*
```

```

Ninja + 0.003*Collection + 0.003*Doctor + 0.003*Sports'),
(5,
    u'0.006*Adult education + 0.005*Forbidden love + 0.005*Religion + 0.004*Historical epic + 0.004*Composer + 0.004*Scandal + 0.004*Fantasy + 0.004*Australia +
0.004*Social Media + 0.004*Election + 0.004*Doctor + 0.004*Folk dance + 0.004*Premier league + 0.004*Finance + 0.004*Parenting + 0.004*Married Woman + 0.003*Ep-
ic adventure + 0.003*Reunion + 0.003*Sisters + 0.003*Participant'),
(6,
    u'0.007*Separation + 0.005*Character is a housewife + 0.005*King + 0.005*Family Feud + 0.004*Movie launch + 0.004*Slapstick + 0.004*Sikh + 0.004*Landlord +
0.004*Hanuman + 0.004*Defend a friend + 0.004*Action + 0.004*Fast food + 0.003*Women empowerment + 0.003*Political Turmoil + 0.003*Folklore + 0.003*Politics +
0.003*Reunion + 0.003*Revenge + 0.003*Depression + 0.003*Hinduism'),
(7,
    u'0.006*Author + 0.005*Saint + 0.005*Technology + 0.005*Review + 0.004*Dispute + 0.004*Champion + 0.004*USA + 0.004*Director + 0.004*Marketing + 0.004*Heist +
0.004*War + 0.004*Business + 0.004*Brothers + 0.004*Celestial objects + 0.004*Terminal illness + 0.004*Fast food + 0.004*Impact + 0.004*Triumph over adversit-
y + 0.003*Restoration + 0.003*IPL'),
(8,
    u'0.005*Throne + 0.005*Healthcare + 0.005*Luxury + 0.004*Alcoholism + 0.004*Jesus Christ + 0.004*Live performances + 0.004*Agent + 0.004*Spirituality + 0.004
*Learn a lesson + 0.004*Food + 0.004*Awards + 0.004*Reforms + 0.004*Robot + 0.004*Historical epic + 0.004*Rebel + 0.004*Robbery + 0.004*Shopping + 0.004*Goal +
0.004*Overview + 0.004*Ruler'),
(9,
    u'0.006*Opportunities + 0.005*Europe + 0.005*Depression + 0.005*Road Trip + 0.004*Flashback + 0.004*Kingdom + 0.004*Jesus Christ + 0.004*False Charge + 0.004
*Treasure hunt + 0.004*Variety programme + 0.004*Holiday + 0.004*Funny videos + 0.004*Obstacles + 0.004*Mysterious + 0.004*Teammates + 0.004*Engineer + 0.004*D-
estiny + 0.004*Kitchen + 0.004*Compilation + 0.004*Virus'),
(10,
    u'0.006*Talk Show + 0.005*Demonstration + 0.005*Set in Kashmir + 0.005*Community + 0.005*Devotees + 0.005*Lookalike + 0.004*Technology + 0.004*Prime Time +
0.004*School + 0.004*Set in the future + 0.004*Social revolution + 0.004*Studio + 0.004*Frank + 0.004*IPL + 0.004*Truth + 0.004*Yesteryear + 0.003*Royal + 0.003
*Literature + 0.003*Rival + 0.003*Regional'),
(11,
    u'0.007*Country + 0.007*Single Mother + 0.006*Wrestling + 0.005*Creative + 0.005*Biographical account of life + 0.004*Media + 0.004*Personalities + 0.004*La-
ndlord + 0.004*Poetry + 0.004*Family Feud + 0.004*Testosterone-fuelled drama + 0.004*History + 0.004*Religious discourse + 0.004*Sai Baba + 0.004*Businessman +
0.004*Stock exchange + 0.004*Participant + 0.004*Vacation + 0.004*Teasers + 0.004*Teacher'),
(12,
    u'0.005*Precaution + 0.005*Match + 0.005*Successful + 0.005*Survival + 0.004*Metropolitan + 0.004*Make a promise + 0.004*Healing + 0.004*Regional + 0.004*Pro-
fit + 0.004*NRI + 0.004*Biological parent + 0.004*Married couple + 0.004*Makeover + 0.004*Grandparents + 0.004*Influence + 0.004*Robber + 0.004*Mountains + 0.004
*Officer + 0.003*Superhero + 0.003*Based on a true story'),
(13,
    u'0.007*Change versus tradition + 0.006*Overview + 0.005*Self-discovery + 0.005*Ghost + 0.005*Campus + 0.004*Chaos + 0.004*Incidents + 0.004*Successful + 0.004
*Sports + 0.004*Entertainment + 0.004*Travel + 0.004*Glamour + 0.004*Mythical + 0.004*Treasure + 0.004*Goddess + 0.004*Movie + 0.004*Odd team of heroes + 0.004
*Remarriage + 0.004*Intellectual + 0.004*Mother-in-law'),
(14,
    u'0.006*Partners + 0.005*Conversation + 0.005*Jesus Christ + 0.005*Social Media + 0.004*Reenactment + 0.004*Property + 0.004*Fantastic creature + 0.004*Victi-
m + 0.004*Family Drama + 0.004*Ninja + 0.004*Architects + 0.004*Urban + 0.004*Spiritual + 0.004*Innocent + 0.004*Creator + 0.004*Statue + 0.004*Heir + 0.004*Ad-
venture on the seas + 0.003*Disorder + 0.003*Revenge saga'),
(15,
    u'0.006*Agriculture + 0.006*CID + 0.006*Comedian + 0.005*Conversation + 0.005*Yoga + 0.005*Farmer + 0.005*Request + 0.004*Ruler + 0.004*Internet + 0.004*Coll-
ege + 0.004*Factory + 0.004*Fortune Teller + 0.004*Teasers + 0.004*Superstar + 0.004*Battling demons + 0.004*Market + 0.004*Futuristic + 0.004*Live performance
s + 0.004*Struggle + 0.004*Grandparents'),
(16,
    u'0.007*Magician + 0.005*Adoption + 0.005*Ghost + 0.005*Hell + 0.005*Criminal mastermind + 0.005*Forest + 0.005*Traditional + 0.004*Puzzle + 0.004*Battle +
0.004*Sacrifice + 0.004*Prime Minister + 0.004*Chartbuster + 0.004*Top 100 + 0.004*Gossip + 0.004*Rescue mission + 0.004*Classic + 0.003*Demonstration + 0.003*
Traditions + 0.003*Photographer + 0.003*Lawyer'),
(17,
    u'0.005*Promotional event + 0.005*Updates + 0.005*Sense of humour + 0.005*CID + 0.005*Matters of the heart + 0.005*Virus + 0.005*Sexuality + 0.004*Painting +
0.004*Regional + 0.004*Weekend + 0.004*Principles + 0.004*Devotees + 0.004*Recitation + 0.004*Farmer + 0.004*Legal + 0.004*Band + 0.003*State + 0.003*Set in th-
e film industry + 0.003*Mysterious + 0.003*Marriage'),
(18,
    u'0.005*Oppression of women + 0.004*Preparations + 0.004*Compilation + 0.004*Back-to-back + 0.004*Escape + 0.004*Priest + 0.004*Planets + 0.004*Hail from a b-
roken family + 0.004*Influence + 0.004*Magician + 0.004*Vintage + 0.004*Recipe + 0.004*Heiress + 0.004*Bollywood + 0.004*Hanuman + 0.004*Underworld + 0.003*Abu-
ndon + 0.003*Virus outbreak + 0.003*Singer + 0.003*Tax'),
(19,
    u'0.006*Scholar + 0.006*Sai Baba + 0.006*Gang Wars + 0.006*Law + 0.005*Contest + 0.005*Mountain climbing + 0.005*Revolt + 0.004*Activists + 0.004*College +
0.004*Suitable for children under 5 + 0.004*Scientist + 0.004*Scandal + 0.004*Domestic help + 0.004*News headlines + 0.004*Performer + 0.004*Action + 0.004*Sho-
rt Story + 0.004*Astronomy + 0.003*Paranormal + 0.003*Espionage')
)

```

```
In [49]: cmhdः=CoherenceModel(model=hdpmodel, texts=showKeywords2['keywords'], dictionary=dictionary_LDA, coherence='c_v')

print cmhdः.get_coherence()

0.7558297733799685
```

```
In [50]: def evaluate_graph(dictionary, corpus, texts, limit):
    """
    Function to display num_topics - LDA graph using c_v coherence

    Parameters:
    -----
    dictionary : Gensim dictionary
    corpus : Gensim corpus
    limit : topic limit

    Returns:
    -----
    lm_list : List of LDA topic models
    c_v : Coherence values corresponding to the LDA model with respective number of topics
    """
    c_v = []
    lm_list = []
    for num_topics in range(1, limit):
        print(num_topics)
        lm = LdaModel(corpus=corpus, num_topics=num_topics, id2word=dictionary, passes=4, alpha=[0.01]*num_topics, \
                      eta=[0.01]*len(dictionary_LDA.keys()))
        lm_list.append(lm)
        cm = CoherenceModel(model=lm, texts=texts, dictionary=dictionary, coherence='c_v')
        c_v.append(cm.get_coherence())

    # Show graph
    x = range(1, limit)
    plt.plot(x, c_v)
    plt.xlabel("num_topics")
    plt.ylabel("Coherence score")
    plt.legend(("c_v"), loc='best')
    plt.show()

    return lm_list, c_v
```

```
In [51]: pyLDAvis.gensim.prepare(lm_list[9], corpus, dictionary_LDA)
```

```
In [52]: lm_list[9].show_topics(formatted=True, num_topics=16, num_words=10)
```

```
Out[52]: [(0,
    u'0.042*Marriage' + 0.039*Family' + 0.036*Family Drama' + 0.030*Children' + 0.028*Competition' + 0.027*Praising' + 0.027*Goddess' + 0.024*Relationships',
    p' + 0.022*Married couple' + 0.021*Contestant'),
(1,
    u'0.127*Updates' + 0.084*News headlines' + 0.076*Breaking News' + 0.067*Developments' + 0.065*Latest' + 0.058*Politics' + 0.043*Current Affairs' + 0.039*Social Issues' +
    0.038*News' + 0.035*Host'),
(2,
    u'0.058*Healthcare' + 0.042*Expert opinion' + 0.039*Health' + 0.026*Healthy diet' + 0.024*Expert analysis' + 0.023*Culture' + 0.021*Doctor' + 0.021*Village' +
    0.020*Tips' + 0.019*Host'),
(3,
    u'0.044*Love' + 0.034*Couple in love' + 0.032*Lovers' + 0.030*Criminal' + 0.028*Criminal offence' + 0.028*Newspaper' + 0.024*Family' + 0.023*Friends' +
    ip' + 0.020*Love-struck' + 0.019*Romance'),
(4,
    u'0.099*Religious leader' + 0.090*Religious discourse' + 0.088*Religious verses' + 0.072*Religion' + 0.059*Spirituality' + 0.056*Spiritual' + 0.054*De-
    votion' + 0.048*Devotees' + 0.045*Spiritual Leader' + 0.043*Religious'),
(5,
    u'0.100*God' + 0.092*God almighty' + 0.080*God created' + 0.073*God is love' + 0.061*God is light' + 0.051*God is love' + 0.041*God is love' + 0.031*God is love' + 0.021*God is love' + 0.011*God is love')
```

```

        u'0.128* Songs + 0.090* Compilation + 0.060* Popular + 0.057* Musician + 0.051* Singer + 0.047* Movie + 0.044* Artist + 0.041* Musical + 0.036* Musi
c" + 0.032*"Singing"),
(6,
u'0.074*Christianity" + 0.062*Christian" + 0.047*Collection" + 0.044*Food" + 0.037*Recipe" + 0.037*Predictions" + 0.033*Chef" + 0.028*Future" + 0.026
*Priest" + 0.026*Cuisine"),
(7,
u'0.252*Set in a newsroom" + 0.026*Technology" + 0.023*Importance" + 0.017*Crime" + 0.017*Investigation" + 0.016*Business" + 0.015*Science" + 0.014*N
ews headlines" + 0.014*Lawyer" + 0.014*Facts"),
(8,
u'0.064*Expert opinion" + 0.057*Expert analysis" + 0.056*Discussions" + 0.055*Social Issues" + 0.043*Politician" + 0.040*Politics" + 0.032*Host" + 0.0
31*Economical" + 0.029*Political party" + 0.027*Panel Discussion"),
(9,
u'0.040*Celebrity" + 0.036*Actor" + 0.032*Host" + 0.028*Personalities" + 0.027*Interaction" + 0.024*Yoga" + 0.024*Interview" + 0.024*Movie" + 0.022
*Life journey" + 0.018*Meditation"))

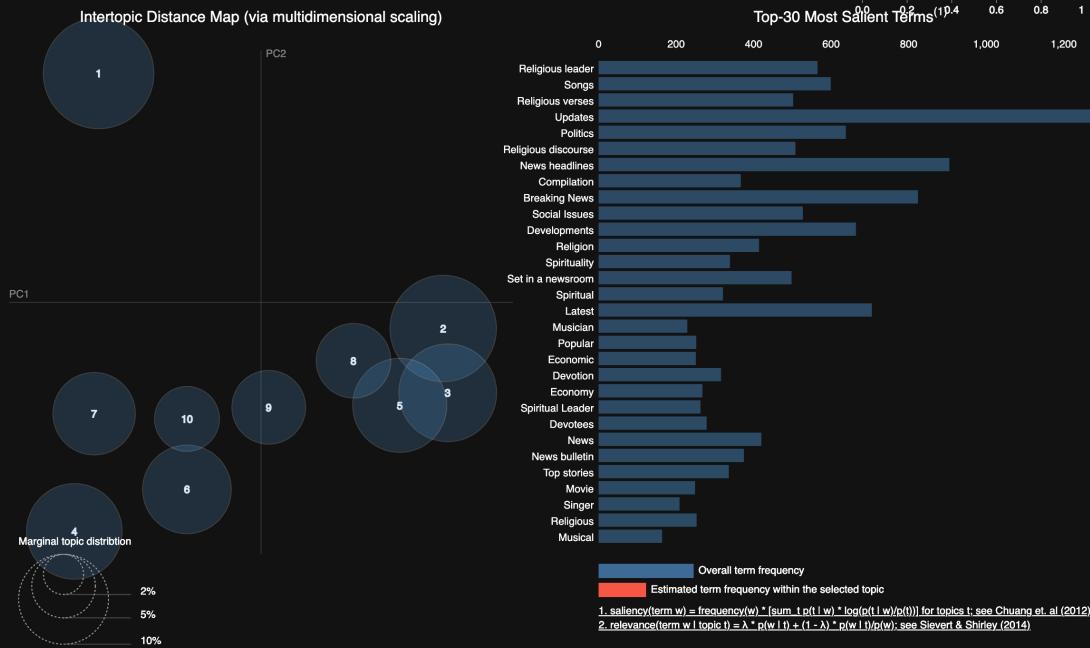
```

In [53]: pyLDAvis.gensim.prepare(lmlist[9], corpus, dictionary\_LDA)

Out[53]: Selected Topic: 0 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2) 0.0 0.2 0.4 0.6 0.8 1

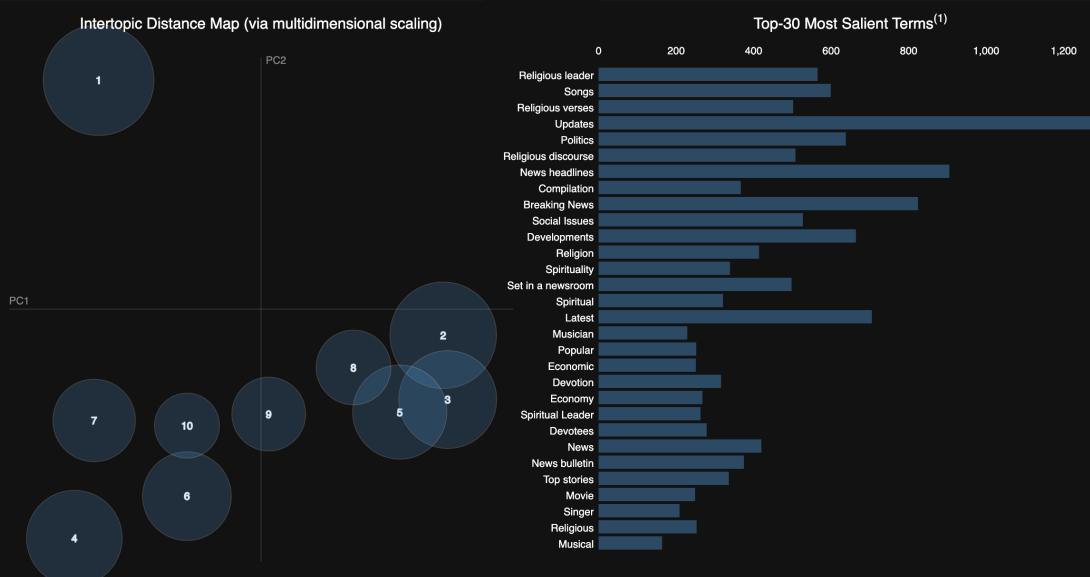
Top-30 Most Salient Terms<sup>(1)</sup> 0.0 0.2 0.4 0.6 0.8 1



Selected Topic: 0 Previous Topic Next Topic Clear Topic

$\lambda = 1$

Top-30 Most Salient Terms<sup>(1)</sup> 0.0 200 400 600 800 1,000 1,200



In [ ]: