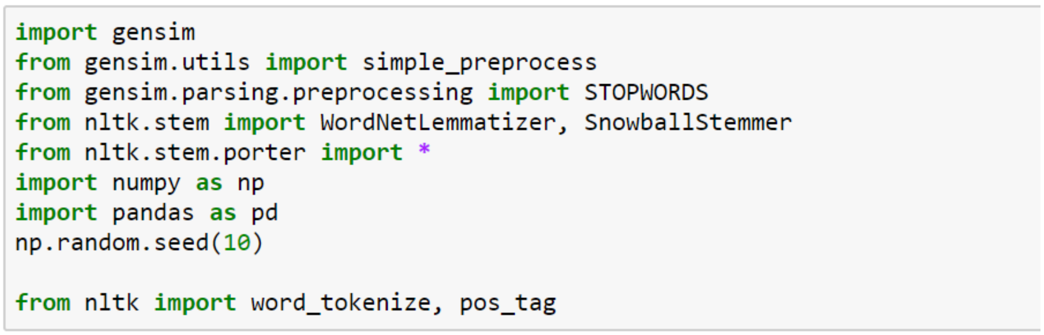
Text Analytics (Topic Modelling)

A general overview of analysis:

1. Start with the question (Objective)
2. Get and clean the data (mostly time-consuming)
3. Perform Exploratory Data Analysis
4. Apply some NLP techniques (Sentiment Analysis, Topic Modelling, Text Generation)
5. Share insights (Interpretations of result)

Useful python libraries

- Related data: Pandas, NumPy, Sklearn, re
- Related NLP: nltk, genism



```
import gensim
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import STOPWORDS
from nltk.stem import WordNetLemmatizer, SnowballStemmer
from nltk.stem.porter import *
import numpy as np
import pandas as pd
np.random.seed(10)

from nltk import word_tokenize, pos_tag
```

Objective: Get topic out of headlines from an Indian newspaper

Get and clean the data

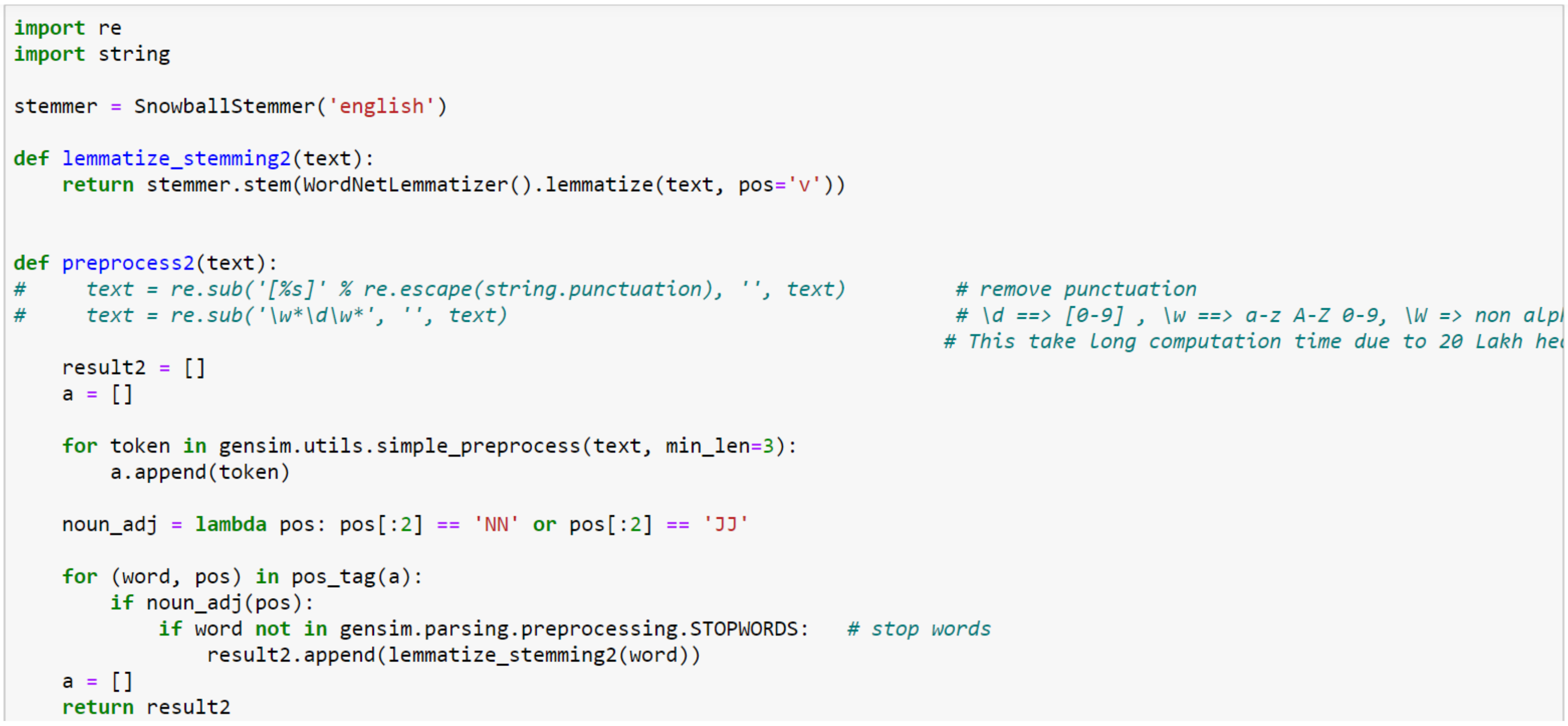
My data: https://www.kaggle.com/therohk/india-headlines-news-dataset (download CSV file) - 18 Years of headlines focusing on India (It contains approximately 2.9 million events published by Times of India. start-2001 to end-2018)

Data cleaning:

- Make text all lower case
- Remove punctuation
- Remove numerical values
- Remove common non-sensical text (/n)

- Tokenize text
- Stemming/lemmatization (If needed)
- Parts of speech tagging
- Create bi-grams or tri-grams (If required)
- Remove stop words
- Many more….

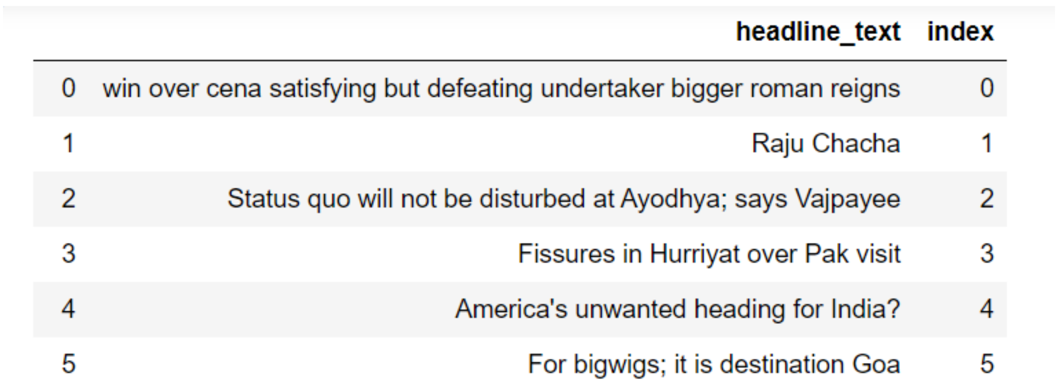
After looking data, necessary first step data cleaning



```
import re
import string

stemmer = SnowballStemmer('english')

def lemmatize_stemming2(text):
    return stemmer.stem(WordNetLemmatizer().lemmatize(text, pos='v'))


def preprocess2(text):
#     text = re.sub('[%s]' % re.escape(string.punctuation), '', text)         # remove punctuation
#     text = re.sub('\w*\d\w*', '', text)                                     # \d ==> [0-9] , \w ==> a-z A-Z 0-9, \W => non alp
                                                                             # This take long computation time due to 20 Lakh he
    result2 = []
    a = []

    for token in gensim.utils.simple_preprocess(text, min_len=3):
        a.append(token)

    noun_adj = lambda pos: pos[:2] == 'NN' or pos[:2] == 'JJ'

    for (word, pos) in pos_tag(a):
        if noun_adj(pos):
            if word not in gensim.parsing.preprocessing.STOPWORDS:    # stop words
                result2.append(lemmatize_stemming2(word))
    a = []
    return result2
```

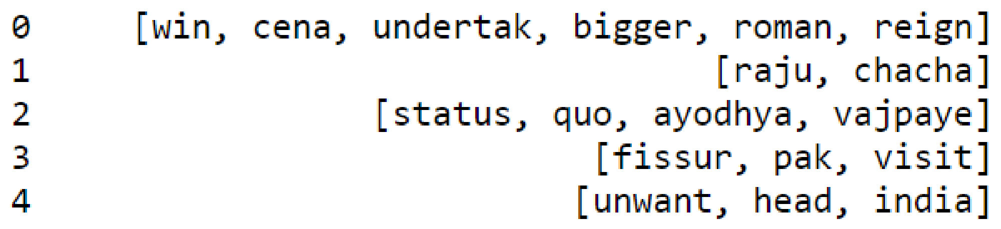
Data Format –

Corpus: Used in sentiment analysis (because the order of words matter)



	headline_text	index
0	win over cena satisfying but defeating undertaker bigger roman reigns	0
1	Raju Chacha	1
2	Status quo will not be disturbed at Ayodhya; says Vajpayee	2
3	Fissures in Hurriyat over Pak visit	3
4	America's unwanted heading for India?	4
5	For bigwigs; it is destination Goa	5

Bag of words:



```
0     [win, cena, undertak, bigger, roman, reign]
1                                  [raju, chacha]
2                 [status, quo, ayodhya, vajpaye]
3                            [fissur, pak, visit]
4                           [unwant, head, india]
```

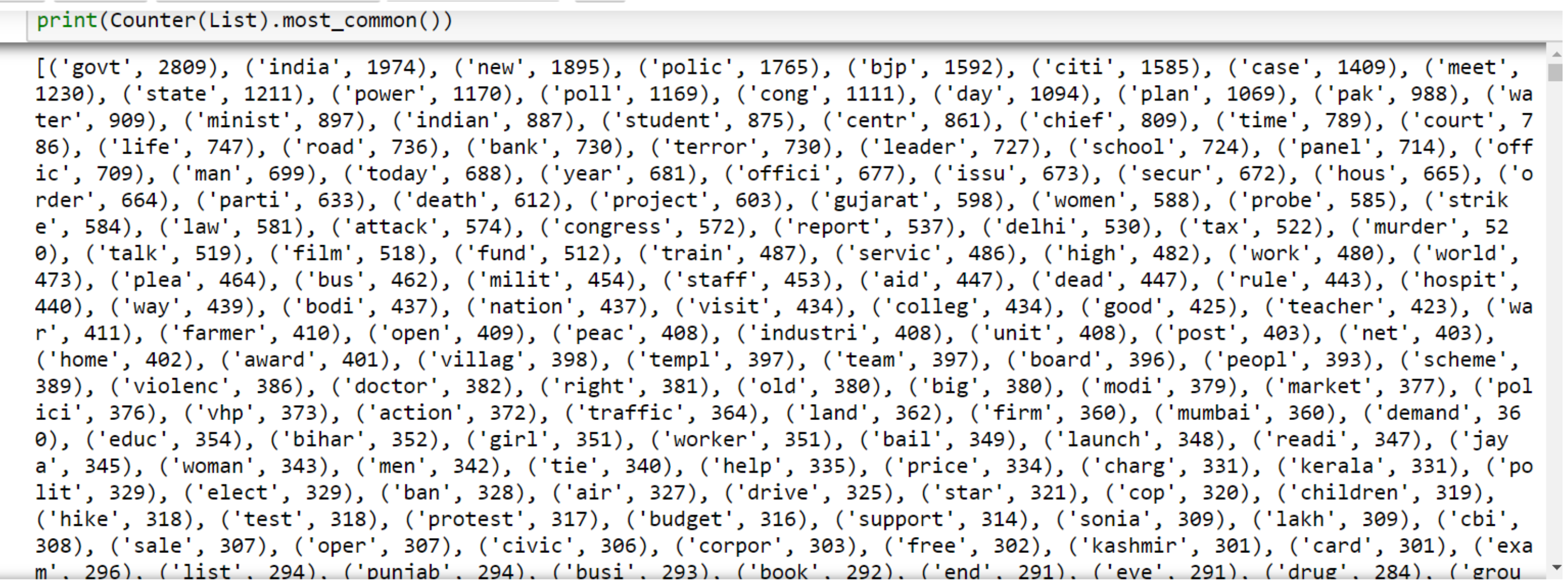
Exploratory data analysis

- Word count

  If words make sense, (i.e. if top words can indicate news topic) then okay.
  But if top words are useless (like – ‘he’, ‘like’, ‘talk’, etc.) then we add those also in stop word!

  I tried to do that, but as data set contains 2969921 rows, and each row has approximately 8-12 words (after clean that data), which took very long (hour) by normal method. So skip it and move on, if the end result require this step then we can come back!

I saw starting “1 Lakh” headlines, the output is:



```
print(Counter(List).most_common())
```

```
[('govt', 2809), ('india', 1974), ('new', 1895), ('polic', 1765), ('bjp', 1592), ('citi', 1585), ('case', 1409), ('meet',
1230), ('state', 1211), ('power', 1170), ('poll', 1169), ('cong', 1111), ('day', 1094), ('plan', 1069), ('pak', 988), ('wa
ter', 909), ('minist', 897), ('indian', 887), ('student', 875), ('centr', 861), ('chief', 809), ('time', 789), ('court', 7
86), ('life', 747), ('road', 736), ('bank', 730), ('terror', 730), ('leader', 727), ('school', 724), ('panel', 714), ('off
ic', 709), ('man', 699), ('today', 688), ('year', 681), ('offici', 677), ('issu', 673), ('secur', 672), ('hous', 665), ('o
rder', 664), ('parti', 633), ('death', 612), ('project', 603), ('gujarat', 598), ('women', 588), ('probe', 585), ('strik
e', 584), ('law', 581), ('attack', 574), ('congress', 572), ('report', 537), ('delhi', 530), ('tax', 522), ('murder', 52
0), ('talk', 519), ('film', 518), ('fund', 512), ('train', 487), ('servic', 486), ('high', 482), ('work', 480), ('world',
473), ('plea', 464), ('bus', 462), ('milit', 454), ('staff', 453), ('aid', 447), ('dead', 447), ('rule', 443), ('hospit',
440), ('way', 439), ('bodi', 437), ('nation', 437), ('visit', 434), ('colleg', 434), ('good', 425), ('teacher', 423), ('wa
r', 411), ('farmer', 410), ('open', 409), ('peac', 408), ('industri', 408), ('unit', 408), ('post', 403), ('net', 403),
('home', 402), ('award', 401), ('villag', 398), ('templ', 397), ('team', 397), ('board', 396), ('peopl', 393), ('scheme',
389), ('violenc', 386), ('doctor', 382), ('right', 381), ('old', 380), ('big', 380), ('modi', 379), ('market', 377), ('pol
ici', 376), ('vhp', 373), ('action', 372), ('traffic', 364), ('land', 362), ('firm', 360), ('mumbai', 360), ('demand', 36
0), ('educ', 354), ('bihar', 352), ('girl', 351), ('worker', 351), ('bail', 349), ('launch', 348), ('readi', 347), ('jay
a', 345), ('woman', 343), ('men', 342), ('tie', 340), ('help', 335), ('price', 334), ('charg', 331), ('kerala', 331), ('po
lit', 329), ('elect', 329), ('ban', 328), ('air', 327), ('drive', 325), ('star', 321), ('cop', 320), ('children', 319),
('hike', 318), ('test', 318), ('protest', 317), ('budget', 316), ('support', 314), ('sonia', 309), ('lakh', 309), ('cbi',
308), ('sale', 307), ('oper', 307), ('civic', 306), ('corpor', 303), ('free', 302), ('kashmir', 301), ('card', 301), ('exa
m', 296), ('list', 294), ('punjab', 294), ('busi', 293), ('book', 292), ('end', 291), ('eve', 291), ('drug', 284), ('grou
```

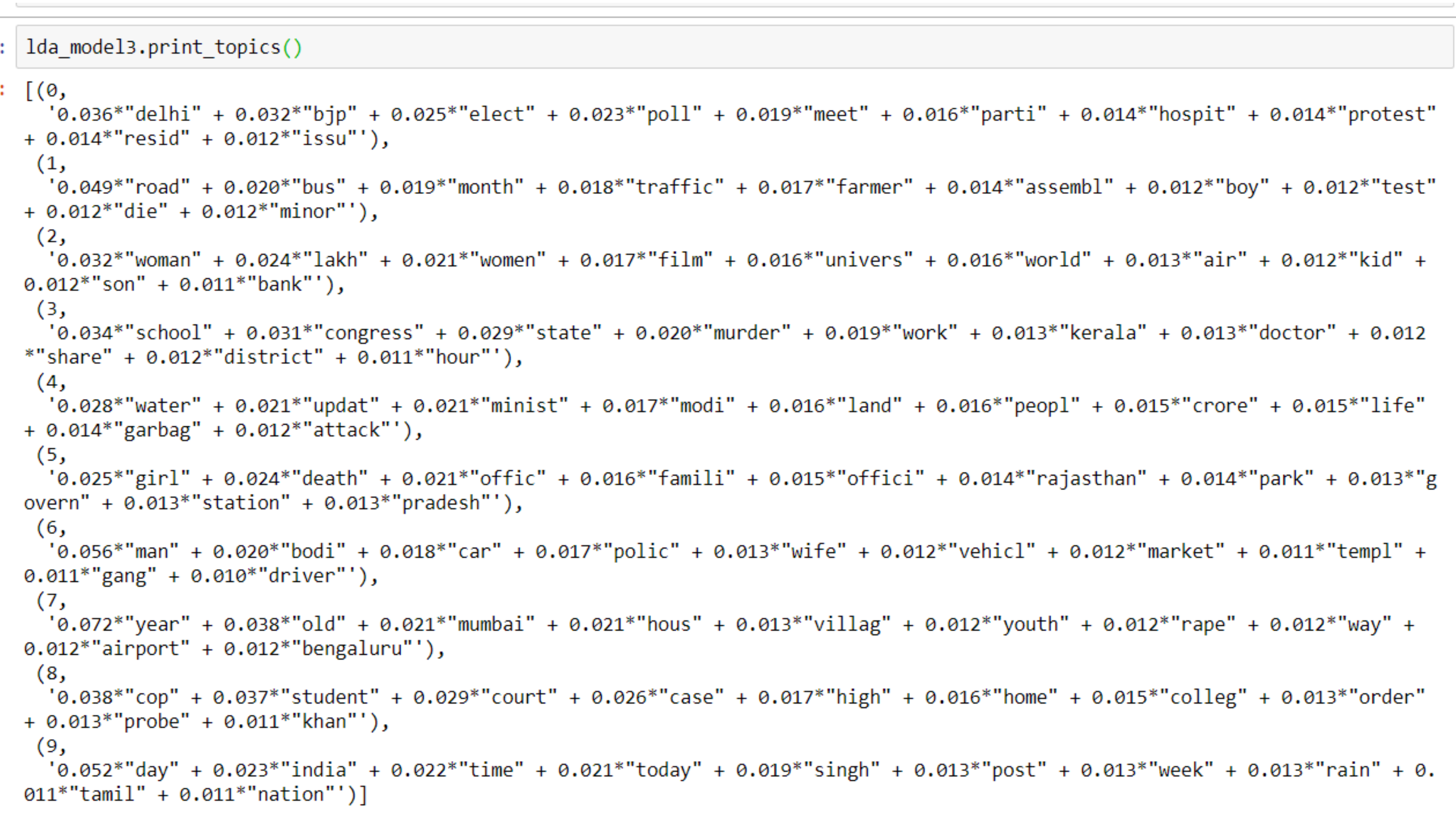
You can see top words are govt and political party related which make sense as its common that ‘political matters’ more in the headline in India.

NLP techniques (Topic Modelling):

Tried with LdaModel but as the dataset has approx 2M row, so it takes a lot of time in computation, so I used LdaMulticore(workers=2)

Firstly apply with all words but the output was not making sense to me, so I filtered out and keep noun and adjective (one method people use in topic modelling – work with the noun and adjective words)

Used different topic numbers but 10 worked best for me!



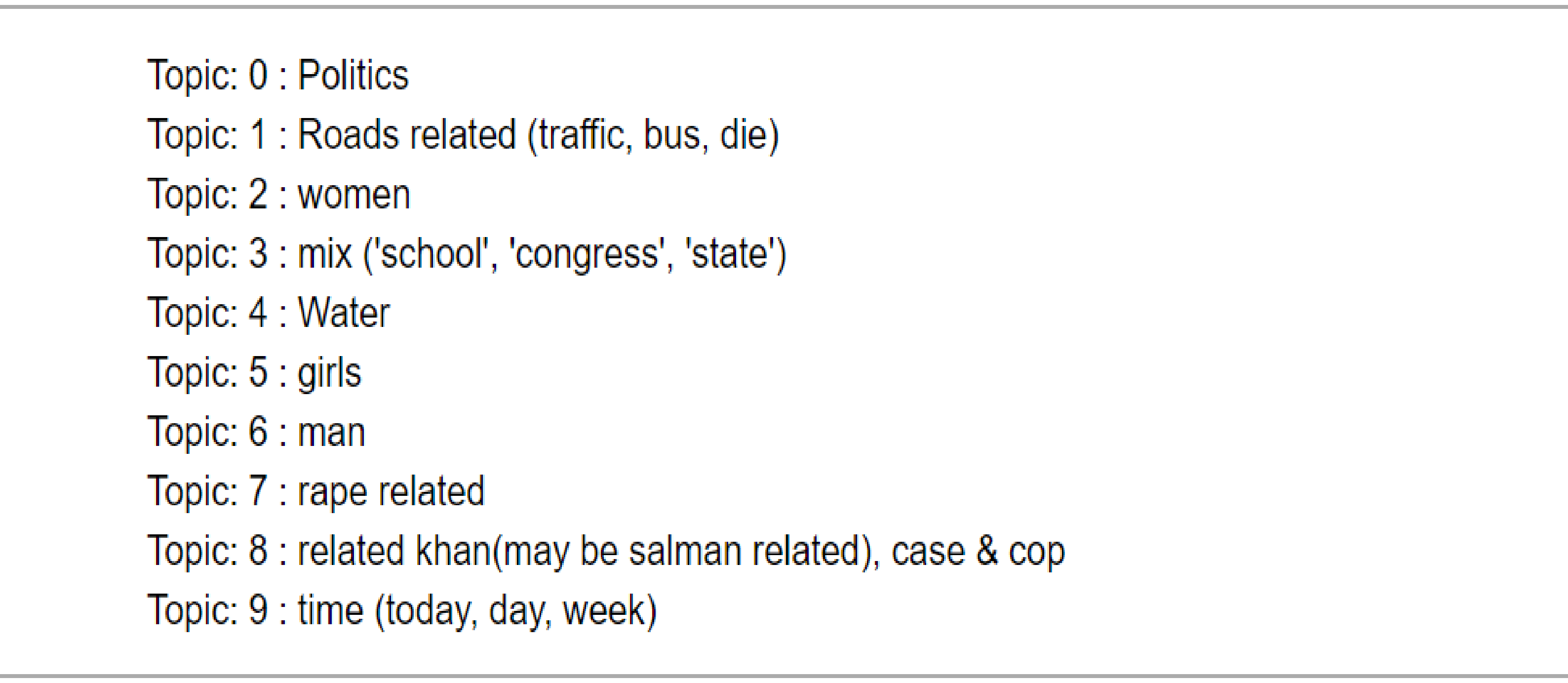
```
:  lda_model3.print_topics()
```

```
: [(0,
   '0.036*"delhi" + 0.032*"bjp" + 0.025*"elect" + 0.023*"poll" + 0.019*"meet" + 0.016*"parti" + 0.014*"hospit" + 0.014*"protest"
 + 0.014*"resid" + 0.012*"issu"'),
  (1,
   '0.049*"road" + 0.020*"bus" + 0.019*"month" + 0.018*"traffic" + 0.017*"farmer" + 0.014*"assembl" + 0.012*"boy" + 0.012*"test"
 + 0.012*"die" + 0.012*"minor"'),
  (2,
   '0.032*"woman" + 0.024*"lakh" + 0.021*"women" + 0.017*"film" + 0.016*"univers" + 0.016*"world" + 0.013*"air" + 0.012*"kid" +
 0.012*"son" + 0.011*"bank"'),
  (3,
   '0.034*"school" + 0.031*"congress" + 0.029*"state" + 0.020*"murder" + 0.019*"work" + 0.013*"kerala" + 0.013*"doctor" + 0.012
 *"share" + 0.012*"district" + 0.011*"hour"'),
  (4,
   '0.028*"water" + 0.021*"updat" + 0.021*"minist" + 0.017*"modi" + 0.016*"land" + 0.016*"peopl" + 0.015*"crore" + 0.015*"life"
 + 0.014*"garbag" + 0.012*"attack"'),
  (5,
   '0.025*"girl" + 0.024*"death" + 0.021*"offic" + 0.016*"famili" + 0.015*"offici" + 0.014*"rajasthan" + 0.014*"park" + 0.013*"g
 overn" + 0.013*"station" + 0.013*"pradesh"'),
  (6,
   '0.056*"man" + 0.020*"bodi" + 0.018*"car" + 0.017*"polic" + 0.013*"wife" + 0.012*"vehicl" + 0.012*"market" + 0.011*"templ" +
 0.011*"gang" + 0.010*"driver"'),
  (7,
   '0.072*"year" + 0.038*"old" + 0.021*"mumbai" + 0.021*"hous" + 0.013*"villag" + 0.012*"youth" + 0.012*"rape" + 0.012*"way" +
 0.012*"airport" + 0.012*"bengaluru"'),
  (8,
   '0.038*"cop" + 0.037*"student" + 0.029*"court" + 0.026*"case" + 0.017*"high" + 0.016*"home" + 0.015*"colleg" + 0.013*"order"
 + 0.013*"probe" + 0.011*"khan"'),
  (9,
   '0.052*"day" + 0.023*"india" + 0.022*"time" + 0.021*"today" + 0.019*"singh" + 0.013*"post" + 0.013*"week" + 0.013*"rain" + 0.
 011*"tamil" + 0.011*"nation"')]
```

### Speed ===> LdaMulticore (workers=3) > LdaMulticore (workers=2) > LdaMulticore (workers=1) > LdaModel

Conclusion:

So, I concluded the topics by looking result are:



Topic: 0 : Politics
Topic: 1 : Roads related (traffic, bus, die)
Topic: 2 : women
Topic: 3 : mix ('school', 'congress', 'state')
Topic: 4 : Water
Topic: 5 : girls
Topic: 6 : man
Topic: 7 : rape related
Topic: 8 : related khan(may be salman related), case & cop
Topic: 9 : time (today, day, week)

And this makes sense to me, as

1. India is concerned about women and girls empowerment with time – eg, enable women to serve as army commanders, i.e. women will now be allowed to command entire military units., some scheme for girls, girls quota.
2. Politics – I saw congress, bjp, Modi in high number – because two national parties and its common that politics related topic in the news.
3. Roads related (traffic, bus, die) – The road construction, potholes in the road, accidents, etc.
4. Rape cases – The sad side of India.
5. Other mix topics - Mumbai, death, man, khan, students, protest, etc.