Assessment_Project_01_Topic_Analysis_of_Review_Data1

August 21, 2022

1 Assessment Project-01:- Topic Analysis of Review Data

[1]:
```
import warnings
warnings.filterwarnings('ignore')
```

[3]:
```
# Import require library

import pandas as pd
from nltk.tokenize import word_tokenize
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
import re
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
from string import punctuation
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
from tmtoolkit.topicmod.evaluate import metric_coherence_gensim
import numpy as np
import gensim
from gensim import corpora
from gensim.models.coherencemodel import CoherenceModel
from gensim.corpora import Dictionary
```

[4]:
```
pd.set_option('display.max_colwidth',150)
```

2 1. Read the .csv file using Pandas. Take a look at the top few records.

[5]:
```
data = pd.read_csv("K8 Reviews v0.2.csv")
data.head()
```

[5]:
```
   sentiment  \
0          1
1          0
```

```
2          1
3          1
4          0

                                                                review
0
Good but need updates and improvements
1  Worst mobile i have bought ever, Battery is draining like hell, backup is
only 6 to 7 hours with internet uses, even if I put mobile idle its gett…
2
when I will get my 10% cash back… its already 15 January..
3
Good
4  The worst phone everThey have changed the last phone but the problem is still
same and the amazon is not returning the phone .Highly disappointing…
```

[6]:
```python
data.shape
```

[6]:
```
(14675, 2)
```

[7]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14675 entries, 0 to 14674
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   sentiment  14675 non-null  int64
 1   review     14675 non-null  object
dtypes: int64(1), object(1)
memory usage: 229.4+ KB
```

[8]:
```python
data.isna().sum().any()
```

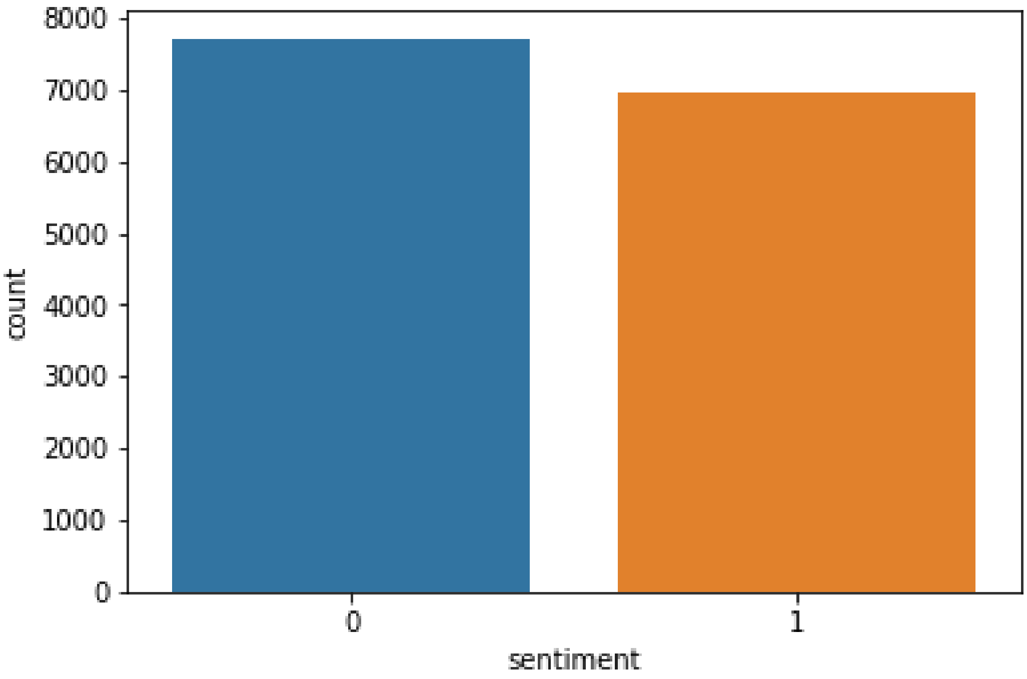
[8]:
```
False
```

[9]:
```python
data['sentiment'].value_counts()
```

[9]:
```
0    7712
1    6963
Name: sentiment, dtype: int64
```

[11]:
```python
sns.countplot(data['sentiment'])
```

[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7eff4078b390>
```



3 2. Normalize casings for the review text and extract the text into a list for easier manipulation.

```
[12]: reviews = data['review'].values
```

```
[13]: reviews[:5]
```

```
[13]: array(['Good but need updates and improvements',
              "Worst mobile i have bought ever, Battery is draining like hell, backup
      is only 6 to 7 hours with internet uses, even if I put mobile idle its getting
      discharged.This is biggest lie from Amazon & Lenove which is not at all
      expected, they are making full by saying that battery is 4000MAH & booster
      charger is fake, it takes at least 4 to 5 hours to be fully charged.Don't know
      how Lenovo will survive by making full of us.Please don;t go for this else you
      will regret like me.",
              'when I will get my 10% cash back… its already 15 January..',
              'Good',
              'The worst phone everThey have changed the last phone but the problem is
      still same and the amazon is not returning the phone .Highly disappointing of
      amazon'],
            dtype=object)
```

```
[14]: review_lower = [term.lower() for term in reviews]
```

[15]:
```
review_lower[:5]
```

[15]:
```
['good but need updates and improvements',
 "worst mobile i have bought ever, battery is draining like hell, backup is only
6 to 7 hours with internet uses, even if i put mobile idle its getting
discharged.this is biggest lie from amazon & lenove which is not at all
expected, they are making full by saying that battery is 4000mah & booster
charger is fake, it takes at least 4 to 5 hours to be fully charged.don't know
how lenovo will survive by making full of us.please don;t go for this else you
will regret like me.",
 'when i will get my 10% cash back… its already 15 january..',
 'good',
 'the worst phone everthey have changed the last phone but the problem is still
same and the amazon is not returning the phone .highly disappointing of amazon']
```

4 3. Tokenize the reviews using NLTKs word_tokenize function.

[17]:
```
review_token = [word_tokenize(token) for token in review_lower]
```

[18]:
```
print(review_token[:5])
```

```
[['good', 'but', 'need', 'updates', 'and', 'improvements'], ['worst', 'mobile',
'i', 'have', 'bought', 'ever', ',', 'battery', 'is', 'draining', 'like', 'hell',
',', 'backup', 'is', 'only', '6', 'to', '7', 'hours', 'with', 'internet',
'uses', ',', 'even', 'if', 'i', 'put', 'mobile', 'idle', 'its', 'getting',
'discharged.this', 'is', 'biggest', 'lie', 'from', 'amazon', '&', 'lenove',
'which', 'is', 'not', 'at', 'all', 'expected', ',', 'they', 'are', 'making',
'full', 'by', 'saying', 'that', 'battery', 'is', '4000mah', '&', 'booster',
'charger', 'is', 'fake', ',', 'it', 'takes', 'at', 'least', '4', 'to', '5',
'hours', 'to', 'be', 'fully', 'charged.do', "n't", 'know', 'how', 'lenovo',
'will', 'survive', 'by', 'making', 'full', 'of', 'us.please', 'don', ';', 't',
'go', 'for', 'this', 'else', 'you', 'will', 'regret', 'like', 'me', '.'],
['when', 'i', 'will', 'get', 'my', '10', '%', 'cash', 'back', '…', 'its',
'already', '15', 'january', '..'], ['good'], ['the', 'worst', 'phone',
'everthey', 'have', 'changed', 'the', 'last', 'phone', 'but', 'the', 'problem',
'is', 'still', 'same', 'and', 'the', 'amazon', 'is', 'not', 'returning', 'the',
'phone', '.highly', 'disappointing', 'of', 'amazon']]
```

5 4. Perform parts-of-speech tagging on each sentence using the NLTK POS tagger.

[20]:
```
review_pos = [nltk.pos_tag(sent) for sent in review_token]
```

[21]:
```
print(review_pos[:5])
```

```
[[('good', 'JJ'), ('but', 'CC'), ('need', 'VBP'), ('updates', 'NNS'), ('and',
'CC'), ('improvements', 'NNS')], [('worst', 'JJS'), ('mobile', 'NN'), ('i',
'NN'), ('have', 'VBP'), ('bought', 'VBN'), ('ever', 'RB'), (',', ','),
('battery', 'NN'), ('is', 'VBZ'), ('draining', 'VBG'), ('like', 'IN'), ('hell',
'NN'), (',', ','), ('backup', 'NN'), ('is', 'VBZ'), ('only', 'RB'), ('6', 'CD'),
('to', 'TO'), ('7', 'CD'), ('hours', 'NNS'), ('with', 'IN'), ('internet', 'JJ'),
('uses', 'NNS'), (',', ','), ('even', 'RB'), ('if', 'IN'), ('i', 'JJ'), ('put',
'VBP'), ('mobile', 'JJ'), ('idle', 'NN'), ('its', 'PRP$'), ('getting', 'VBG'),
('discharged.this', 'NN'), ('is', 'VBZ'), ('biggest', 'JJS'), ('lie', 'NN'),
('from', 'IN'), ('amazon', 'NN'), ('&', 'CC'), ('lenove', 'NN'), ('which',
'WDT'), ('is', 'VBZ'), ('not', 'RB'), ('at', 'IN'), ('all', 'DT'), ('expected',
'VBN'), (',', ','), ('they', 'PRP'), ('are', 'VBP'), ('making', 'VBG'), ('full',
'JJ'), ('by', 'IN'), ('saying', 'VBG'), ('that', 'DT'), ('battery', 'NN'),
('is', 'VBZ'), ('4000mah', 'CD'), ('&', 'CC'), ('booster', 'JJR'), ('charger',
'NN'), ('is', 'VBZ'), ('fake', 'JJ'), (',', ','), ('it', 'PRP'), ('takes',
'VBZ'), ('at', 'IN'), ('least', 'JJS'), ('4', 'CD'), ('to', 'TO'), ('5', 'CD'),
('hours', 'NNS'), ('to', 'TO'), ('be', 'VB'), ('fully', 'RB'), ('charged.do',
'VBP'), ("n't", 'RB'), ('know', 'VB'), ('how', 'WRB'), ('lenovo', 'JJ'),
('will', 'MD'), ('survive', 'VB'), ('by', 'IN'), ('making', 'VBG'), ('full',
'JJ'), ('of', 'IN'), ('us.please', 'JJ'), ('don', 'NN'), (';', ':'), ('t',
'CC'), ('go', 'VB'), ('for', 'IN'), ('this', 'DT'), ('else', 'JJ'), ('you',
'PRP'), ('will', 'MD'), ('regret', 'VB'), ('like', 'IN'), ('me', 'PRP'), ('.',
'.')], [('when', 'WRB'), ('i', 'NN'), ('will', 'MD'), ('get', 'VB'), ('my',
'PRP$'), ('10', 'CD'), ('%', 'NN'), ('cash', 'NN'), ('back', 'RB'), ('…',
'VBZ'), ('its', 'PRP$'), ('already', 'RB'), ('15', 'CD'), ('january', 'JJ'),
('..', 'NN')], [('good', 'JJ')], [('the', 'DT'), ('worst', 'JJS'), ('phone',
'NN'), ('everthey', 'NN'), ('have', 'VBP'), ('changed', 'VBN'), ('the', 'DT'),
('last', 'JJ'), ('phone', 'NN'), ('but', 'CC'), ('the', 'DT'), ('problem',
'NN'), ('is', 'VBZ'), ('still', 'RB'), ('same', 'JJ'), ('and', 'CC'), ('the',
'DT'), ('amazon', 'NN'), ('is', 'VBZ'), ('not', 'RB'), ('returning', 'VBG'),
('the', 'DT'), ('phone', 'NN'), ('.highly', 'RB'), ('disappointing', 'JJ'),
('of', 'IN'), ('amazon', 'NN')]]
```

6 5. For the topic model, we should want to include only nouns.

6.1 i. Find out all the POS tags that correspond to nouns.

6.2 ii. Limit the data to only terms with these tags.

```
[22]: nltk.download('tagsets')
```

```
[nltk_data] Downloading package tagsets to /root/nltk_data…
[nltk_data]   Package tagsets is already up-to-date!
```

```
[22]: True
```

```
[23]: nltk.help.upenn_tagset()
```

```
$: dollar
    $ -$ --$ A$ C$ HK$ M$ NZ$ S$ U.S.$ US$
'': closing quotation mark
    ' ''
(: opening parenthesis
    ( [ {
): closing parenthesis
    ) ] }
,: comma
    ,
--: dash
    --
.: sentence terminator
    . ! ?
:: colon or ellipsis
    : ; …
CC: conjunction, coordinating
    & 'n and both but either et for less minus neither nor or plus so
    therefore times v. versus vs. whether yet
CD: numeral, cardinal
    mid-1890 nine-thirty forty-two one-tenth ten million 0.5 one forty-
    seven 1987 twenty '79 zero two 78-degrees eighty-four IX '60s .025
    fifteen 271,124 dozen quintillion DM2,000 …
DT: determiner
    all an another any both del each either every half la many much nary
    neither no some such that the them these this those
EX: existential there
    there
FW: foreign word
    gemeinschaft hund ich jeux habeas Haementeria Herr K'ang-si vous
    lutihaw alai je jour objets salutaris fille quibusdam pas trop Monte
    terram fiche oui corporis …
IN: preposition or conjunction, subordinating
    astride among uppon whether out inside pro despite on by throughout
    below within for towards near behind atop around if like until below
    next into if beside …
JJ: adjective or numeral, ordinal
    third ill-mannered pre-war regrettable oiled calamitous first separable
    ectoplasmic battery-powered participatory fourth still-to-be-named
    multilingual multi-disciplinary …
JJR: adjective, comparative
    bleaker braver breezier briefer brighter brisker broader bumper busier
    calmer cheaper choosier cleaner clearer closer colder commoner costlier
    cozier creamier crunchier cuter …
JJS: adjective, superlative
    calmest cheapest choicest classiest cleanest clearest closest commonest
    corniest costliest crassest creepiest crudest cutest darkest deadliest
    dearest deepest densest dinkiest …
```

```
LS: list item marker
    A A. B B. C C. D E F First G H I J K One SP-44001 SP-44002 SP-44005
    SP-44007 Second Third Three Two * a b c d first five four one six three
    two
MD: modal auxiliary
    can cannot could couldn't dare may might must need ought shall should
    shouldn't will would
NN: noun, common, singular or mass
    common-carrier cabbage knuckle-duster Casino afghan shed thermostat
    investment slide humour falloff slick wind hyena override subhumanity
    machinist …
NNP: noun, proper, singular
    Motown Venneboerger Czestochwa Ranzer Conchita Trumplane Christos
    Oceanside Escobar Kreisler Sawyer Cougar Yvette Ervin ODI Darryl CTCA
    Shannon A.K.C. Meltex Liverpool …
NNPS: noun, proper, plural
    Americans Americas Amharas Amityvilles Amusements Anarcho-Syndicalists
    Andalusians Andes Andruses Angels Animals Anthony Antilles Antiques
    Apache Apaches Apocrypha …
NNS: noun, common, plural
    undergraduates scotches bric-a-brac products bodyguards facets coasts
    divestitures storehouses designs clubs fragrances averages
    subjectivists apprehensions muses factory-jobs …
PDT: pre-determiner
    all both half many quite such sure this
POS: genitive marker
    ' 's
PRP: pronoun, personal
    hers herself him himself hisself it itself me myself one oneself ours
    ourselves ownself self she thee theirs them themselves they thou thy us
PRP$: pronoun, possessive
    her his mine my our ours their thy your
RB: adverb
    occasionally unabatingly maddeningly adventurously professedly
    stirringly prominently technologically magisterially predominately
    swiftly fiscally pitilessly …
RBR: adverb, comparative
    further gloomier grander graver greater grimmer harder harsher
    healthier heavier higher however larger later leaner lengthier less-
    perfectly lesser lonelier longer louder lower more …
RBS: adverb, superlative
    best biggest bluntest earliest farthest first furthest hardest
    heartiest highest largest least less most nearest second tightest worst
RP: particle
    aboard about across along apart around aside at away back before behind
    by crop down ever fast for forth from go high i.e. in into just later
    low more off on open out over per pie raising start teeth that through
    under unto up up-pp upon whole with you
```

```
SYM: symbol
    % & ' '' ''. ) ). * + ,. < = > @ A[fj] U.S U.S.S.R * ** ***
TO: "to" as preposition or infinitive marker
    to
UH: interjection
    Goodbye Goody Gosh Wow Jeepers Jee-sus Hubba Hey Kee-reist Oops amen
    huh howdy uh dammit whammo shucks heck anyways whodunnit honey golly
    man baby diddle hush sonuvabitch …
VB: verb, base form
    ask assemble assess assign assume atone attention avoid bake balkanize
    bank begin behold believe bend benefit bevel beware bless boil bomb
    boost brace break bring broil brush build …
VBD: verb, past tense
    dipped pleaded swiped regummed soaked tidied convened halted registered
    cushioned exacted snubbed strode aimed adopted belied figgered
    speculated wore appreciated contemplated …
VBG: verb, present participle or gerund
    telegraphing stirring focusing angering judging stalling lactating
    hankerin' alleging veering capping approaching traveling besieging
    encrypting interrupting erasing wincing …
VBN: verb, past participle
    multihulled dilapidated aerosolized chaired languished panelized used
    experimented flourished imitated reunifed factored condensed sheared
    unsettled primed dubbed desired …
VBP: verb, present tense, not 3rd person singular
    predominate wrap resort sue twist spill cure lengthen brush terminate
    appear tend stray glisten obtain comprise detest tease attract
    emphasize mold postpone sever return wag …
VBZ: verb, present tense, 3rd person singular
    bases reconstructs marks mixes displeases seals carps weaves snatches
    slumps stretches authorizes smolders pictures emerges stockpiles
    seduces fizzes uses bolsters slaps speaks pleads …
WDT: WH-determiner
    that what whatever which whichever
WP: WH-pronoun
    that what whatever whatsoever which who whom whosoever
WP$: WH-pronoun, possessive
    whose
WRB: Wh-adverb
    how however whence whenever where whereby whereever wherein whereof why
``: opening quotation mark
    ` ``
```

[24]:
```
review_nouns = []
for term in review_pos:
    review_nouns.append([token for token in term if re.search('NN.*',token[1])])
```

[25]:
```
print(review_nouns[:5])
```

```
[[('updates', 'NNS'), ('improvements', 'NNS')], [('mobile', 'NN'), ('i', 'NN'),
('battery', 'NN'), ('hell', 'NN'), ('backup', 'NN'), ('hours', 'NNS'), ('uses',
'NNS'), ('idle', 'NN'), ('discharged.this', 'NN'), ('lie', 'NN'), ('amazon',
'NN'), ('lenove', 'NN'), ('battery', 'NN'), ('charger', 'NN'), ('hours', 'NNS'),
('don', 'NN')], [('i', 'NN'), ('%', 'NN'), ('cash', 'NN'), ('..', 'NN')], [],
[('phone', 'NN'), ('everthey', 'NN'), ('phone', 'NN'), ('problem', 'NN'),
('amazon', 'NN'), ('phone', 'NN'), ('amazon', 'NN')]]
```

7 6. Lemmatize.

7.1 i. Different forms of the terms need to be treated as one.

7.2 ii. No need to provide POS tag to lemmatizer for now.

[28]:
```
lemmatizer = WordNetLemmatizer()
review_lemma = []
for term in review_nouns:
    review_lemma.append([lemmatizer.lemmatize(sent[0]) for sent in term])
```

[29]:
```
print(review_lemma[:5])
```

```
[['update', 'improvement'], ['mobile', 'i', 'battery', 'hell', 'backup', 'hour',
'us', 'idle', 'discharged.this', 'lie', 'amazon', 'lenove', 'battery',
'charger', 'hour', 'don'], ['i', '%', 'cash', '..'], [], ['phone', 'everthey',
'phone', 'problem', 'amazon', 'phone', 'amazon']]
```

8 7. Remove stopwords and punctuation (if there are any).

[31]:
```
stop_words = stopwords.words('english')
```

[32]:
```
review_nonstopwords = []
for term in review_lemma:
    review_nonstopwords.append([word for word in term if word not in␣
 ↪stop_words])
```

[33]:
```
print(review_nonstopwords[:5])
```

```
[['update', 'improvement'], ['mobile', 'battery', 'hell', 'backup', 'hour',
'us', 'idle', 'discharged.this', 'lie', 'amazon', 'lenove', 'battery',
'charger', 'hour'], ['%', 'cash', '..'], [], ['phone', 'everthey', 'phone',
'problem', 'amazon', 'phone', 'amazon']]
```

[34]:
```
punct = list(punctuation)
```

[35]:

```
nonpunctuation = []
re_puct = re.compile('[%s]' % re.escape(punctuation))
for sent in review_nonstopwords:
    nonpunctuation.append([re_puct.sub('',word) for word in sent])
```

[36]:

```
nonpunctuation[:5]
```

[36]:

```
[['update', 'improvement'],
 ['mobile',
  'battery',
  'hell',
  'backup',
  'hour',
  'us',
  'idle',
  'dischargedthis',
  'lie',
  'amazon',
  'lenove',
  'battery',
  'charger',
  'hour'],
 ['', 'cash', ''],
 [],
 ['phone', 'everthey', 'phone', 'problem', 'amazon', 'phone', 'amazon']]
```

[37]:

```
remove = []
re_br = re.compile('(br)$')
for term in nonpunctuation:
    remove.append([re_br.sub('',word) for word in term])

review_clean = []
for sent in remove:
    review_clean.append([word for word in sent if word.isalpha()])
```

[38]:

```
review_clean[:5]
```

[38]:

```
[['update', 'improvement'],
 ['mobile',
  'battery',
  'hell',
  'backup',
  'hour',
  'us',
  'idle',
  'dischargedthis',
  'lie',
  'amazon',
  'lenove',
  'battery',
  'charger',
  'hour'],
 ['cash'],
 [],
 ['phone', 'everthey', 'phone', 'problem', 'amazon', 'phone', 'amazon']]
```

9 8. Create a topic model using LDA on the cleaned-up data with 12 topics.

9.1 i. Print out the top terms for each topic.

[39]:

```
# Define topics
n_topics = 12
```

[40]:

```
# Create Dictionary
id2word = corpora.Dictionary(review_clean)

#Create corpus
texts = review_clean
```

[41]:

```
# Term Document Frequency
corpus = [id2word.doc2bow(text) for text in texts]

print(corpus[50])
print(id2word[1])
print([[(id2word[i],freq) for i, freq in corp] for corp in corpus[:2]])
```

```
[(79, 1), (104, 1), (105, 1)]
update
[[('improvement', 1), ('update', 1)], [('amazon', 1), ('backup', 1), ('battery',
2), ('charger', 1), ('dischargedthis', 1), ('hell', 1), ('hour', 2), ('idle',
1), ('lenove', 1), ('lie', 1), ('mobile', 1), ('us', 1)]]
```

[42]:

```
# Build LDA model
lda_model_gensim = gensim.models.ldamodel.
 ↪LdaModel(corpus=corpus,id2word=id2word,num_topics=12,random_state=42,
                                                       ␣
 ↪passes=10,per_word_topics=True)
```

[43]:

```
# Print the Keyword in the 12 topics
lda_model_gensim.print_topics()
```

```
[43]: [(0,
        '0.077*"feature" + 0.062*"heat" + 0.051*"superb" + 0.049*"h" + 0.024*"set" +
      0.024*"cost" + 0.021*"r" + 0.017*"cell" + 0.011*"k" + 0.010*"fine"'),
       (1,
        '0.093*"phone" + 0.053*"lenovo" + 0.042*"screen" + 0.038*"device" +
      0.034*"note" + 0.029*"problem" + 0.027*"option" + 0.025*"service" + 0.022*"day"
      + 0.019*"star"'),
       (2,
        '0.114*"phone" + 0.075*"price" + 0.073*"amazon" + 0.038*"service" +
      0.037*"product" + 0.033*"delivery" + 0.029*"range" + 0.026*"time" +
      0.025*"return" + 0.022*"replacement"'),
       (3,
        '0.112*"issue" + 0.112*"phone" + 0.093*"money" + 0.044*"waste" + 0.035*"value"
      + 0.027*"network" + 0.017*"lot" + 0.016*"worth" + 0.015*"box" + 0.015*"month"'),
       (4,
        '0.256*"problem" + 0.116*"heating" + 0.050*"performance" + 0.023*"network" +
      0.019*"excellent" + 0.018*"smartphone" + 0.017*"ok" + 0.016*"everything" +
      0.014*"awesome" + 0.013*"connection"'),
       (5,
        '0.257*"battery" + 0.056*"camera" + 0.049*"backup" + 0.042*"phone" +
      0.039*"day" + 0.036*"hour" + 0.034*"issue" + 0.028*"life" + 0.024*"time" +
      0.021*"performance"'),
       (6,
        '0.085*"charger" + 0.051*"call" + 0.025*"volta" + 0.024*"turbo" +
      0.018*"condition" + 0.018*"month" + 0.018*"piece" + 0.017*"speed" +
      0.015*"message" + 0.014*"notification"'),
       (7,
        '0.424*"phone" + 0.033*"hai" + 0.014*"ho" + 0.012*"plz" + 0.009*"hi" +
      0.008*"month" + 0.008*"color" + 0.007*"bhi" + 0.007*"hang" + 0.006*"charge"'),
       (8,
        '0.315*"mobile" + 0.043*"speaker" + 0.038*"glass" + 0.020*"gorilla" +
      0.017*"display" + 0.014*"class" + 0.013*"work" + 0.013*"screen" + 0.012*"cover"
      + 0.011*"gud"'),
       (9,
        '0.096*"note" + 0.036*"sim" + 0.031*"network" + 0.025*"phone" + 0.023*"system"
      + 0.021*"call" + 0.020*"card" + 0.019*"sensor" + 0.019*"jio" + 0.018*"budget"'),
       (10,
        '0.222*"product" + 0.043*"update" + 0.031*"lenovo" + 0.031*"software" +
      0.021*"handset" + 0.018*"issue" + 0.017*"review" + 0.015*"apps" +
      0.014*"feature" + 0.014*"memory"'),
       (11,
        '0.183*"camera" + 0.086*"quality" + 0.072*"phone" + 0.025*"price" +
      0.022*"feature" + 0.022*"performance" + 0.021*"mode" + 0.020*"sound" +
      0.019*"processor" + 0.013*"depth"')]
```

9.2 ii. What is the coherence of the model with the c_v metric?

[44]:
```
# Compute Coherence Score
coherence_lda_gensim =␣
 ↪CoherenceModel(model=lda_model_gensim,texts=review_clean,dictionary=id2word,coherence='c_v'
```

[45]:
```
coherence_lda = coherence_lda_gensim.get_coherence()
```

[46]:
```
print('Coherence:',coherence_lda)
```

```
Coherence: 0.5265659302002864
```

[47]:
```
# Compute Perplexity for 12 topics
print('Perplexity:',lda_model_gensim.log_perplexity(corpus))
```

```
Perplexity: -6.653522184369614
```

10 9. Analyze the topics through the business lens.

10.1 i. Determine which of the topics can be combined.

[48]:
```
x = lda_model_gensim.show_topics(num_topics=12,formatted=False)
topics_words = [(tp[0], [wd[0] for wd in tp[1]]) for tp in x]
for topic,words in topics_words:
    print(str(topic) + " --> " + str(words))
print()
```

```
0 --> ['feature', 'heat', 'superb', 'h', 'set', 'cost', 'r', 'cell', 'k',
'fine']
1 --> ['phone', 'lenovo', 'screen', 'device', 'note', 'problem', 'option',
'service', 'day', 'star']
2 --> ['phone', 'price', 'amazon', 'service', 'product', 'delivery', 'range',
'time', 'return', 'replacement']
3 --> ['issue', 'phone', 'money', 'waste', 'value', 'network', 'lot', 'worth',
'box', 'month']
4 --> ['problem', 'heating', 'performance', 'network', 'excellent',
'smartphone', 'ok', 'everything', 'awesome', 'connection']
5 --> ['battery', 'camera', 'backup', 'phone', 'day', 'hour', 'issue', 'life',
'time', 'performance']
6 --> ['charger', 'call', 'volta', 'turbo', 'condition', 'month', 'piece',
'speed', 'message', 'notification']
7 --> ['phone', 'hai', 'ho', 'plz', 'hi', 'month', 'color', 'bhi', 'hang',
'charge']
8 --> ['mobile', 'speaker', 'glass', 'gorilla', 'display', 'class', 'work',
'screen', 'cover', 'gud']
9 --> ['note', 'sim', 'network', 'phone', 'system', 'call', 'card', 'sensor',
'jio', 'budget']
10 --> ['product', 'update', 'lenovo', 'software', 'handset', 'issue', 'review',
'apps', 'feature', 'memory']
11 --> ['camera', 'quality', 'phone', 'price', 'feature', 'performance', 'mode',
'sound', 'processor', 'depth']
```

Coherence measures the interpretability of the topics. Good value for Coherence measure c_V is 0.5.

Topic 5 and 6 are vaguely about the battery life and charger.

Topic 3 and 4 are vaguely about the product heating and problems.

Topic 8, 9 and 11 are vaguely about the product functions.

Above topics can be combined.

After we combined some topics, we are left with 8 topics. Lets repeat the LDA for 8 topics and see the result.

11 10. Create a topic model using LDA with what you think is the optimal number of topics

[49]:

```
# Build LDA model with 8 topics
lda_model_8 = gensim.models.ldamodel.
↪LdaModel(corpus=corpus,id2word=id2word,num_topics=8,random_state=42,passes=10,
                                                   per_word_topics=True)
```

[50]:

```
# Print the Keyword in the 12 topics
lda_model_8.print_topics()
```

[50]:

```
[(0,
  '0.170*"mobile" + 0.050*"charger" + 0.044*"feature" + 0.029*"device" +
0.026*"battery" + 0.023*"turbo" + 0.018*"hour" + 0.017*"charging" + 0.015*"day"
+ 0.013*"issue"'),
 (1,
  '0.073*"phone" + 0.039*"service" + 0.032*"screen" + 0.024*"problem" +
0.020*"amazon" + 0.020*"product" + 0.019*"day" + 0.019*"speaker" +
0.018*"option" + 0.017*"lenovo"'),
 (2,
  '0.163*"product" + 0.083*"price" + 0.082*"phone" + 0.030*"range" +
0.026*"amazon" + 0.022*"delivery" + 0.019*"feature" + 0.017*"superb" +
0.017*"return" + 0.017*"glass"'),
 (3,
  '0.091*"money" + 0.056*"issue" + 0.043*"waste" + 0.034*"value" +
0.030*"update" + 0.027*"h" + 0.027*"software" + 0.026*"system" + 0.021*"box" +
0.017*"work"'),
 (4,
  '0.178*"problem" + 0.052*"heating" + 0.034*"hai" + 0.019*"network" +
0.017*"please" + 0.014*"ho" + 0.014*"excellent" + 0.013*"smartphone" +
0.011*"plz" + 0.009*"message"'),
 (5,
  '0.191*"camera" + 0.098*"battery" + 0.081*"quality" + 0.039*"performance" +
0.034*"backup" + 0.016*"mode" + 0.016*"display" + 0.014*"sound" +
0.012*"everything" + 0.011*"depth"'),
 (6,
  '0.084*"note" + 0.050*"phone" + 0.037*"lenovo" + 0.032*"network" +
0.030*"call" + 0.023*"sim" + 0.019*"feature" + 0.013*"card" + 0.012*"jio" +
0.011*"stock"'),
 (7,
  '0.291*"phone" + 0.075*"battery" + 0.035*"issue" + 0.030*"time" + 0.025*"day"
+ 0.022*"month" + 0.019*"hour" + 0.018*"heat" + 0.015*"life" + 0.015*"use"')]
```

11.1 i. What is the coherence of the model?

[51]:
```
# Compute Coherence Score for 8 topics
coherence_lda_gensim =␣
 ↪CoherenceModel(model=lda_model_8,texts=review_clean,dictionary=id2word,coherence='c_v')
```

[52]:
```
coherence_lda = coherence_lda_gensim.get_coherence()
```

[53]:
```
print('Coherence:',coherence_lda)
```

```
Coherence: 0.5724932854996856
```

[54]:
```
# Compute Perplexity for 8 topics
print('Perplexity:',lda_model_8.log_perplexity(corpus))
```

```
Perplexity: -6.585275005141235
```

12 11. The business should be able to interpret the topics.

12.1 i. Name each of the identified topics.

[55]:
```
x = lda_model_8.show_topics(formatted=False)
topics_words = [(tp[0], [wd[0] for wd in tp[1]]) for tp in x]
for topic,words in topics_words:
    print(str(topic)+ "::"+ str(words))
print()
```

```
0::['mobile', 'charger', 'feature', 'device', 'battery', 'turbo', 'hour',
'charging', 'day', 'issue']
1::['phone', 'service', 'screen', 'problem', 'amazon', 'product', 'day',
'speaker', 'option', 'lenovo']
2::['product', 'price', 'phone', 'range', 'amazon', 'delivery', 'feature',
'superb', 'return', 'glass']
3::['money', 'issue', 'waste', 'value', 'update', 'h', 'software', 'system',
'box', 'work']
4::['problem', 'heating', 'hai', 'network', 'please', 'ho', 'excellent',
'smartphone', 'plz', 'message']
5::['camera', 'battery', 'quality', 'performance', 'backup', 'mode', 'display',
'sound', 'everything', 'depth']
6::['note', 'phone', 'lenovo', 'network', 'call', 'sim', 'feature', 'card',
'jio', 'stock']
7::['phone', 'battery', 'issue', 'time', 'day', 'month', 'hour', 'heat', 'life',
'use']
```

12.2 ii. Create a table with the topic name and the top 10 terms in each to present to the business.

12.3 Topic - Business Name

Topic 0 :- Battery Related

Topic 1 :- Customer Service

Topic 2 :- Performace of eshopping platform

Topic 3 :- Feedback of the product

Topic 4 :- Features on which pricing depend

Topic 5 :- Option to be considere for shopping

Topic 6 :- Related to Communication or connectivity

Topic 7 :- Phone Performance

[55]: