In [1]:

```
from google.colab import drive
drive.mount('/content/MyDrive/')
```

```
Mounted at /content/MyDrive/
```

DESCRIPTION

Help a leading mobile brand understand the voice of the customer by analyzing the reviews of their product on Amazon and the topics that customers are talking about. You will perform topic modeling on specific parts of speech. You'll finally interpret the emerging topics.

Problem Statement:

A popular mobile phone brand, Lenovo has launched their budget smartphone in the Indian market. The client wants to understand the VOC (voice of the customer) on the product. This will be useful to not just evaluate the current product, but to also get some direction for developing the product pipeline. The client is particularly interested in the different aspects that customers care about. Product reviews by customers on a leading e-commerce site should provide a good view.

Domain: Amazon reviews for a leading phone brand

Analysis to be done: POS tagging, topic modeling usin

In [3]:

```
import warnings
warnings.filterwarnings('ignore', category=DeprecationWarning)
```

In [29]:

```
import pandas as pd
import nltk
nltk.download('all')
```

```
[nltk_data] Downloading collection 'all'
[nltk_data]    | 
[nltk_data]    | Downloading package abc to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/abc.zip.
[nltk_data]    | Downloading package alpino to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/alpino.zip.
[nltk_data]    | Downloading package averaged_perceptron_tagger to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data]    | Downloading package averaged_perceptron_tagger_ru to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping
[nltk_data]    |       taggers/averaged_perceptron_tagger_ru.zip.
[nltk_data]    | Downloading package basque_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/basque_grammars.zip.
[nltk_data]    | Downloading package biocreative_ppi to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/biocreative_ppi.zip.
[nltk_data]    | Downloading package bllip_wsj_no_aux to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping models/bllip_wsj_no_aux.zip.
[nltk_data]    | Downloading package book_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/book_grammars.zip.
[nltk_data]    | Downloading package brown to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/brown.zip.
[nltk_data]    | Downloading package brown_tei to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/brown_tei.zip.
[nltk_data]    | Downloading package cess_cat to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cess_cat.zip.
[nltk_data]    | Downloading package cess_esp to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cess_esp.zip.
[nltk_data]    | Downloading package chat80 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/chat80.zip.
[nltk_data]    | Downloading package city_database to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/city_database.zip.
[nltk_data]    | Downloading package cmudict to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cmudict.zip.
[nltk_data]    | Downloading package comparative_sentences to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/comparative_sentences.zip.
[nltk_data]    | Downloading package comtrans to /root/nltk_data...
[nltk_data]    | Downloading package conll2000 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2000.zip.
[nltk_data]    | Downloading package conll2002 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2002.zip.
[nltk_data]    | Downloading package conll2007 to /root/nltk_data...
[nltk_data]    | Downloading package crubadan to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/crubadan.zip.
[nltk_data]    | Downloading package dependency_treebank to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/dependency_treebank.zip.
[nltk_data]    | Downloading package dolch to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/dolch.zip.
[nltk_data]    | Downloading package europarl_raw to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/europarl_raw.zip.
[nltk_data]    | Downloading package extended_omw to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    | Downloading package floresta to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/floresta.zip.
[nltk_data]    | Downloading package framenet_v15 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/framenet_v15.zip.
[nltk_data]    | Downloading package framenet_v17 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/framenet_v17.zip.
[nltk_data]    | Downloading package gazetteers to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/gazetteers.zip.
[nltk_data]    | Downloading package genesis to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/genesis.zip.
[nltk_data]    | Downloading package gutenberg to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/gutenberg.zip.
[nltk_data]    | Downloading package ieer to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/ieer.zip.
[nltk_data]    | Downloading package inaugural to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/inaugural.zip.
[nltk_data]    | Downloading package indian to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/indian.zip.
[nltk_data]    | Downloading package jeita to /root/nltk_data...
[nltk_data]    | Downloading package kimmo to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/kimmo.zip.
[nltk_data]    | Downloading package knbc to /root/nltk_data...
[nltk_data]    | Downloading package large_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/large_grammars.zip.
[nltk_data]    | Downloading package lin_thesaurus to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/lin_thesaurus.zip.
[nltk_data]    | Downloading package mac_morpho to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/mac_morpho.zip.
[nltk_data]    | Downloading package machado to /root/nltk_data...
[nltk_data]    | Downloading package masc_tagged to /root/nltk_data...
[nltk_data]    | Downloading package maxent_ne_chunker to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping chunkers/maxent_ne_chunker.zip.
[nltk_data]    | Downloading package maxent_treebank_pos_tagger to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping taggers/maxent_treebank_pos_tagger.zip.
[nltk_data]    | Downloading package moses_sample to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping models/moses_sample.zip.
[nltk_data]    | Downloading package movie_reviews to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/movie_reviews.zip.
[nltk_data]    | Downloading package mte_teip5 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/mte_teip5.zip.
[nltk_data]    | Downloading package mwa_ppdb to /root/nltk_data...
[nltk_data]    |   Unzipping misc/mwa_ppdb.zip.
[nltk_data]    | Downloading package names to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/names.zip.
[nltk_data]    | Downloading package nombank.1.0 to /root/nltk_data...
[nltk_data]    | Downloading package nonbreaking_prefixes to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/nonbreaking_prefixes.zip.
[nltk_data]    | Downloading package nps_chat to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/nps_chat.zip.
[nltk_data]    | Downloading package omw to /root/nltk_data...
[nltk_data]    | Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]    | Downloading package opinion_lexicon to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/opinion_lexicon.zip.
[nltk_data]    | Downloading package panlex_swadesh to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    | Downloading package paradigms to /root/nltk_data...
```

```
[nltk_data]    |   Unzipping corpora/paradigms.zip.
[nltk_data]    | Downloading package pe08 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/pe08.zip.
[nltk_data]    | Downloading package perluniprops to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping misc/perluniprops.zip.
[nltk_data]    | Downloading package pil to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/pil.zip.
[nltk_data]    | Downloading package pl196x to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/pl196x.zip.
[nltk_data]    | Downloading package porter_test to /root/nltk_data...
[nltk_data]    |   Unzipping stemmers/porter_test.zip.
[nltk_data]    | Downloading package ppattach to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/ppattach.zip.
[nltk_data]    | Downloading package problem_reports to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/problem_reports.zip.
[nltk_data]    | Downloading package product_reviews_1 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/product_reviews_1.zip.
[nltk_data]    | Downloading package product_reviews_2 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/product_reviews_2.zip.
[nltk_data]    | Downloading package propbank to /root/nltk_data...
[nltk_data]    | Downloading package pros_cons to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/pros_cons.zip.
[nltk_data]    | Downloading package ptb to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/ptb.zip.
[nltk_data]    | Downloading package punkt to /root/nltk_data...
[nltk_data]    |   Package punkt is already up-to-date!
[nltk_data]    | Downloading package qc to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/qc.zip.
[nltk_data]    | Downloading package reuters to /root/nltk_data...
[nltk_data]    | Downloading package rslp to /root/nltk_data...
[nltk_data]    |   Unzipping stemmers/rslp.zip.
[nltk_data]    | Downloading package rte to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/rte.zip.
[nltk_data]    | Downloading package sample_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/sample_grammars.zip.
[nltk_data]    | Downloading package semcor to /root/nltk_data...
[nltk_data]    | Downloading package senseval to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/senseval.zip.
[nltk_data]    | Downloading package sentence_polarity to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/sentence_polarity.zip.
[nltk_data]    | Downloading package sentiwordnet to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/sentiwordnet.zip.
[nltk_data]    | Downloading package shakespeare to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/shakespeare.zip.
[nltk_data]    | Downloading package sinica_treebank to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/sinica_treebank.zip.
[nltk_data]    | Downloading package smultron to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/smultron.zip.
[nltk_data]    | Downloading package snowball_data to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    | Downloading package spanish_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/spanish_grammars.zip.
[nltk_data]    | Downloading package state_union to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/state_union.zip.
[nltk_data]    | Downloading package stopwords to /root/nltk_data...
[nltk_data]    |   Package stopwords is already up-to-date!
[nltk_data]    | Downloading package subjectivity to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/subjectivity.zip.
[nltk_data]    | Downloading package swadesh to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/swadesh.zip.
[nltk_data]    | Downloading package switchboard to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/switchboard.zip.
[nltk_data]    | Downloading package tagsets to /root/nltk_data...
[nltk_data]    |   Unzipping help/tagsets.zip.
[nltk_data]    | Downloading package timit to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/timit.zip.
[nltk_data]    | Downloading package toolbox to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/toolbox.zip.
[nltk_data]    | Downloading package treebank to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/treebank.zip.
[nltk_data]    | Downloading package twitter_samples to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/twitter_samples.zip.
[nltk_data]    | Downloading package udhr to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/udhr.zip.
[nltk_data]    | Downloading package udhr2 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/udhr2.zip.
[nltk_data]    | Downloading package unicode_samples to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/unicode_samples.zip.
[nltk_data]    | Downloading package universal_tagset to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping taggers/universal_tagset.zip.
[nltk_data]    | Downloading package universal_treebanks_v20 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    | Downloading package vader_lexicon to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    | Downloading package verbnet to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/verbnet.zip.
[nltk_data]    | Downloading package verbnet3 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/verbnet3.zip.
[nltk_data]    | Downloading package webtext to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/webtext.zip.
[nltk_data]    | Downloading package wmt15_eval to /root/nltk_data...
[nltk_data]    |   Unzipping models/wmt15_eval.zip.
[nltk_data]    | Downloading package word2vec_sample to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping models/word2vec_sample.zip.
[nltk_data]    | Downloading package wordnet to /root/nltk_data...
[nltk_data]    |   Package wordnet is already up-to-date!
[nltk_data]    | Downloading package wordnet2021 to /root/nltk_data...
[nltk_data]    | Downloading package wordnet31 to /root/nltk_data...
[nltk_data]    | Downloading package wordnet_ic to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/wordnet_ic.zip.
[nltk_data]    | Downloading package words to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/words.zip.
[nltk_data]    | Downloading package ycoe to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/ycoe.zip.
[nltk_data]    |
[nltk_data]  Done downloading collection all
```

```
Out[29]: True
```

```
In [9]: df = pd.read_csv('/content/MyDrive/MyDrive/NLP Simplilearn/Proj1/K8 Reviews v0.2.csv')
In [10]: df.head()
```

Out[10]:

	sentiment	review
0	1	Good but need updates and improvements
1	0	Worst mobile i have bought ever, Battery is dr...
2	1	when I will get my 10% cash back.... its alrea...
3	1	Good
4	0	The worst phone everThey have changed the last...

```
In [14]: df = df.drop(['sentiment'],axis=1)
```

```
In [15]: df.head()
```

Out[15]:

	review
0	Good but need updates and improvements
1	Worst mobile i have bought ever, Battery is dr...
2	when I will get my 10% cash back.... its alrea...
3	Good
4	The worst phone everThey have changed the last...

```
In [16]: df.shape
Out[16]: (14675, 1)
```

Data Pre-Processing

Replacing/Dropping NULL values

```
In [17]: df.isnull().sum()
Out[17]: review    0
         dtype: int64
```

Converting to LOWER case

```
In [18]: df['clean_review'] = df['review'].apply(lambda x: str(x).lower())
         df.head()
```

Out[18]:

	review	clean_review
0	Good but need updates and improvements	good but need updates and improvements
1	Worst mobile i have bought ever, Battery is dr...	worst mobile i have bought ever, battery is dr...
2	when I will get my 10% cash back.... its alrea...	when i will get my 10% cash back.... its alrea...
3	Good	good
4	The worst phone everThey have changed the last...	the worst phone everthey have changed the last...

REMOVE NON-ALPHA DATA(DIGITS,PUNCTUATIONS,DIACRITICS)

```
In [19]: df['clean_review'] = df['clean_review'].str.replace(r'[^a-zA-Z\s]', ' ',regex=True)
         df.head()
```

Out[19]:

	review	clean_review
0	Good but need updates and improvements	good but need updates and improvements
1	Worst mobile i have bought ever, Battery is dr...	worst mobile i have bought ever battery is dr...
2	when I will get my 10% cash back.... its alrea...	when i will get my cash back its alrea...
3	Good	good
4	The worst phone everThey have changed the last...	the worst phone everthey have changed the last...

REMOVING WHITE SPACE

```
In [20]: df['clean_review'] = df['clean_review'].str.replace(r'\s{2,}', ' ',regex=True)
         df.head()
```

Out[20]:

	review	clean_review
0	Good but need updates and improvements	good but need updates and improvements
1	Worst mobile i have bought ever, Battery is dr...	worst mobile i have bought ever battery is dra...
2	when I will get my 10% cash back.... its alrea...	when i will get my cash back its already january
3	Good	good
4	The worst phone everThey have changed the last...	the worst phone everthey have changed the last...

WORD TOKENIZATION

```
In [21]: import nltk
         from nltk.tokenize import word_tokenize
         nltk.download('punkt')

         [nltk_data] Downloading package punkt to /root/nltk_data...
         [nltk_data]   Unzipping tokenizers/punkt.zip.

Out[21]: True
```

In [22]:
```
df['clean_review'] = df['clean_review'].apply(lambda x: word_tokenize(x))
df.head()
```

Out[22]:

	review	clean_review
0	Good but need updates and improvements	[good, but, need, updates, and, improvements]
1	Worst mobile i have bought ever, Battery is dr...	[worst, mobile, i, have, bought, ever, battery...
2	when I will get my 10% cash back.... its alrea...	[when, i, will, get, my, cash, back, its, alre...
3	Good	[good]
4	The worst phone everThey have changed the last...	[the, worst, phone, everthey, have, changed, t...

REMOVE UNNECESSARY WORDS

In [24]:
```
from nltk.corpus import stopwords
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

Out[24]:
```
True
```

In [25]:
```
df['clean_review'] = df['clean_review'].apply\
(lambda x:[word for word in x if word not in stopwords.words("english") and len(word) > 3 and word.isalpha()])
df.head()
```

Out[25]:

	review	clean_review
0	Good but need updates and improvements	[good, need, updates, improvements]
1	Worst mobile i have bought ever, Battery is dr...	[worst, mobile, bought, ever, battery, drainin...
2	when I will get my 10% cash back.... its alrea...	[cash, back, already, january]
3	Good	[good]
4	The worst phone everThey have changed the last...	[worst, phone, everthey, changed, last, phone,...

In [26]:
```
df = df[df['clean_review'].map(lambda x: len(x)) > 1].reset_index(drop=True)
#Keeping records with more than single words
```

In [27]:
```
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

Out[27]:
```
True
```

LEMMATIZATION

In [30]:
```
df['clean_review'] = df['clean_review'].apply\
(lambda x: [WordNetLemmatizer().lemmatize(word) for word in x])
df.head()
```

Out[30]:

	review	clean_review
0	Good but need updates and improvements	[good, need, update, improvement]
1	Worst mobile i have bought ever, Battery is dr...	[worst, mobile, bought, ever, battery, drainin...
2	when I will get my 10% cash back.... its alrea...	[cash, back, already, january]
3	The worst phone everThey have changed the last...	[worst, phone, everthey, changed, last, phone,...
4	Only I'm telling don't buyI'm totally disappoi...	[telling, buyi, totally, disappointedpoor, bat...

Extracting only NOUN

In [31]:
```
df['clean_review'] = df['clean_review'].apply\
(lambda x: [word for word in x if nltk.pos_tag([word])[0][1] == 'NN'])
```

In [32]:
```
df = df[df['clean_review'].map(lambda x: len(x)) > 1].reset_index(drop=True)
# Keeping records with more than single words
```

In [34]:
```
df.head()
```

Out[34]:

	review	clean_review
0	Good but need updates and improvements	[need, update, improvement]
1	Worst mobile i have bought ever, Battery is dr...	[mobile, bought, battery, hell, backup, hour, ...
2	when I will get my 10% cash back.... its alrea...	[cash, january]
3	The worst phone everThey have changed the last...	[phone, everthey, phone, problem, amazon, phon...
4	Only I'm telling don't buyI'm totally disappoi...	[buyi, disappointedpoor, batterypoor, camerawa...

Document Term Matrix

In [49]:
```
import gensim
from gensim import corpora
```

In [37]:
```
dictionary = corpora.Dictionary(df['clean_review'])
print(dictionary)

# We have 6724 unique tokens
```

```
Dictionary(6724 unique tokens: ['improvement', 'need', 'update', 'amazon', 'backup']...)
```

In [38]:
```
doc_term_matrix = df['clean_review'].apply(lambda x: dictionary.doc2bow(x))
doc_term_matrix[:10]

# Each tokenized words has been assigned index value and thier count in corpus
```

Out[38]:
```
0                             [(0, 1), (1, 1), (2, 1)]
1    [(3, 1), (4, 1), (5, 2), (6, 1), (7, 1), (8, 1...
2                                   [(19, 1), (20, 1)]
3                  [(3, 2), (21, 1), (22, 3), (23, 1)]
4        [(24, 1), (25, 1), (26, 1), (27, 1), (28, 1)]
5    [(14, 1), (22, 1), (29, 1), (30, 1), (31, 1), ...
6                           [(5, 1), (36, 1), (37, 1)]
7    [(14, 2), (22, 2), (23, 2), (34, 1), (38, 1), ...
8                 [(44, 1), (45, 1), (46, 1), (47, 1)]
9                  [(8, 1), (22, 1), (48, 1), (49, 1)]
Name: clean_review, dtype: object
```

LDA

In [39]:
```
from IPython.display import clear_output
```

In [40]:
```
Lda = gensim.models.ldamodel.LdaModel
ldamodel = Lda(corpus=doc_term_matrix, num_topics=12, id2word=dictionary, passes=10,random_state=45)
clear_output()

# corpus requires document term matrix
# num_topics is used to define number of topics to create from corpus
# id2word requires mapping of words
# passes is used to define number of iterations
```

In [41]:
```
ldamodel.print_topics()

# We have printed all 12 topics and their keywords generated by LDA
```

Out[41]:
```
[(0,
  '0.199*"camera" + 0.099*"quality" + 0.041*"phone" + 0.031*"sound" + 0.026*"front" + 0.025*"battery" + 0.022*"mode" + 0.019*"depth" + 0.017*"rear" + 0.016*"feature"'),
 (1,
  '0.057*"android" + 0.042*"phone" + 0.034*"feature" + 0.031*"stock" + 0.028*"card" + 0.026*"contact" + 0.022*"user" + 0.021*"memory" + 0.020*"headphone" + 0.017*"slot"')
 (2,
  '0.315*"mobile" + 0.162*"problem" + 0.091*"heating" + 0.031*"battery" + 0.022*"heat" + 0.014*"network" + 0.012*"game" + 0.008*"month" + 0.007*"class" + 0.007*"hang"'),
 (3,
  '0.062*"phone" + 0.060*"screen" + 0.058*"charger" + 0.048*"turbo" + 0.039*"feature" + 0.027*"glass" + 0.018*"gorilla" + 0.017*"time" + 0.017*"charge" + 0.015*"core"'),
 (4,
  '0.120*"update" + 0.053*"phone" + 0.049*"software" + 0.034*"need" + 0.034*"system" + 0.028*"oreo" + 0.026*"problem" + 0.019*"lenovo" + 0.013*"bill" + 0.012*"please"'),
 (5,
  '0.196*"phone" + 0.101*"battery" + 0.057*"price" + 0.052*"camera" + 0.050*"awesome" + 0.047*"performance" + 0.044*"backup" + 0.027*"range" + 0.027*"life" + 0.020*"super
b"'),
 (6,
  '0.129*"battery" + 0.100*"issue" + 0.057*"heating" + 0.047*"fast" + 0.042*"phone" + 0.042*"drain" + 0.039*"hour" + 0.038*"charge" + 0.028*"time" + 0.021*"usage"'),
 (7,
  '0.297*"product" + 0.036*"price" + 0.029*"excellent" + 0.018*"performance" + 0.013*"awesome" + 0.013*"till" + 0.013*"amazon" + 0.012*"expectation" + 0.012*"feature" + 0
0*"lenovo"'),
 (8,
  '0.063*"call" + 0.049*"phone" + 0.036*"network" + 0.035*"device" + 0.030*"work" + 0.029*"screen" + 0.026*"speaker" + 0.025*"issue" + 0.025*"support" + 0.022*"cast"'),
 (9,
  '0.149*"note" + 0.144*"lenovo" + 0.072*"phone" + 0.020*"redmi" + 0.016*"killer" + 0.013*"review" + 0.011*"game" + 0.011*"model" + 0.010*"bought" + 0.009*"feature"'),
 (10,
  '0.104*"phone" + 0.071*"amazon" + 0.043*"service" + 0.035*"lenovo" + 0.034*"return" + 0.023*"day" + 0.022*"please" + 0.020*"product" + 0.020*"problem" + 0.019*"custome
r"'),
 (11,
  '0.160*"money" + 0.086*"waste" + 0.074*"worth" + 0.061*"value" + 0.032*"delivery" + 0.017*"super" + 0.013*"buying" + 0.012*"facility" + 0.009*"dont" + 0.008*"iron"')]
```

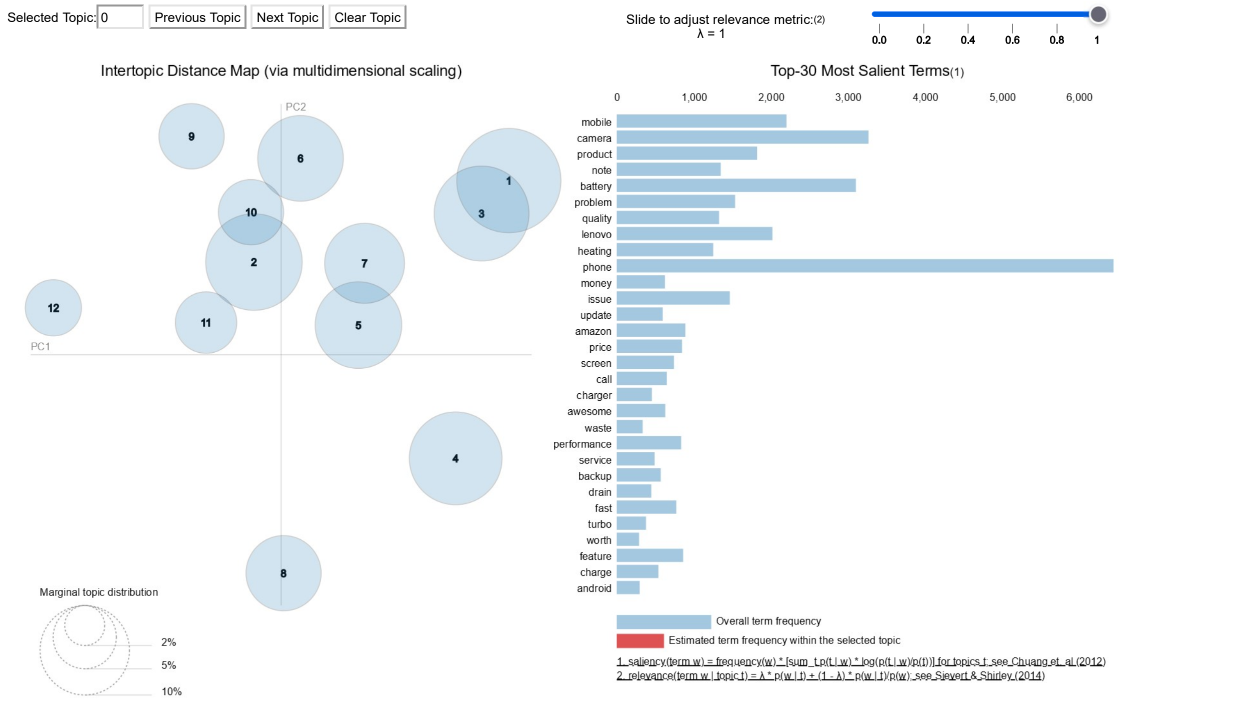
Visualizing LDA model topics

In [45]:
```
import pyLDAvis
import pyLDAvis.gensim_models as gensim

pyLDAvis.enable_notebook()
vis = gensim.prepare(ldamodel,doc_term_matrix,dictionary)
vis
```

```
/usr/local/lib/python3.8/dist-packages/pyLDAvis/_prepare.py:246: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'lab
' will be keyword-only
  default_term_info = default_term_info.sort_values(
```

Out[45]:



Since, some topics in above graph are overlapping each other we will try to find optimal number of topics.

In [46]:
```
from gensim.models.coherencemodel import CoherenceModel
coherence_model_lda = CoherenceModel(model=ldamodel,texts=df['clean_review'],\
                                     dictionary=dictionary , coherence='c_v')
print('\nCoherence Score: ', coherence_model_lda.get_coherence())

# Compute Coherence Score
```

```
Coherence Score:  0.5758709646389434
```

In [50]:
```
from gensim.models import LdaModel
```

In [51]:
```
# Computing coherence score for different size of topic

def calculate_topic_cv(ldamodel,texts,dictionary,topic_range):
  cv_score =[]
  topic_num = []
  for i in range(2,topic_range):
    topic_num.append(i)
    ldamodel = LdaModel(doc_term_matrix, num_topics=i, id2word=dictionary, passes=10,random_state=45)
    cv_score.append(CoherenceModel(model=ldamodel,texts=texts,\
                                 dictionary=dictionary , coherence='c_v').get_coherence())
    clear_output()
  return topic_num,cv_score
```

In [52]:
```
topic_num,cv_score = calculate_topic_cv(ldamodel,df['clean_review'],dictionary,15)
```

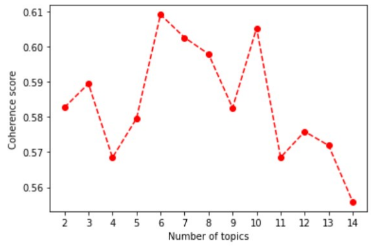
In [53]:
```
pd.DataFrame(zip(topic_num,cv_score),columns=['Topic','Coherence_Score']).set_index\
('Topic').sort_values('Coherence_Score',ascending=False)
```

Out[53]:

	Coherence_Score
Topic	
6	0.609116
10	0.605270
7	0.602581
8	0.597894
3	0.589486
2	0.582708
9	0.582576
5	0.579579
12	0.575871
13	0.571891
11	0.568505
4	0.568473
14	0.555884

In [55]:
```
import matplotlib.pyplot as plt

plt.plot(topic_num,cv_score,color='red', marker='o', linestyle='dashed')
plt.xticks(range(2,15))
plt.xlabel('Number of topics')
plt.ylabel('Coherence score')
plt.show()
```



we will be going with number of topic 6 as with 8 topics there will be many overlaps .

In [56]:
```
# Creating LDA model with number of topics as 6

Lda = gensim.models.ldamodel.LdaModel
ldamodel = Lda(doc_term_matrix, num_topics=6, id2word=dictionary, passes=10,random_state=45)
clear_output()
print(CoherenceModel(model=ldamodel,texts=df['clean_review'],\
                     dictionary=dictionary , coherence='c_v').get_coherence())
```

```
0.6091161154634883
```

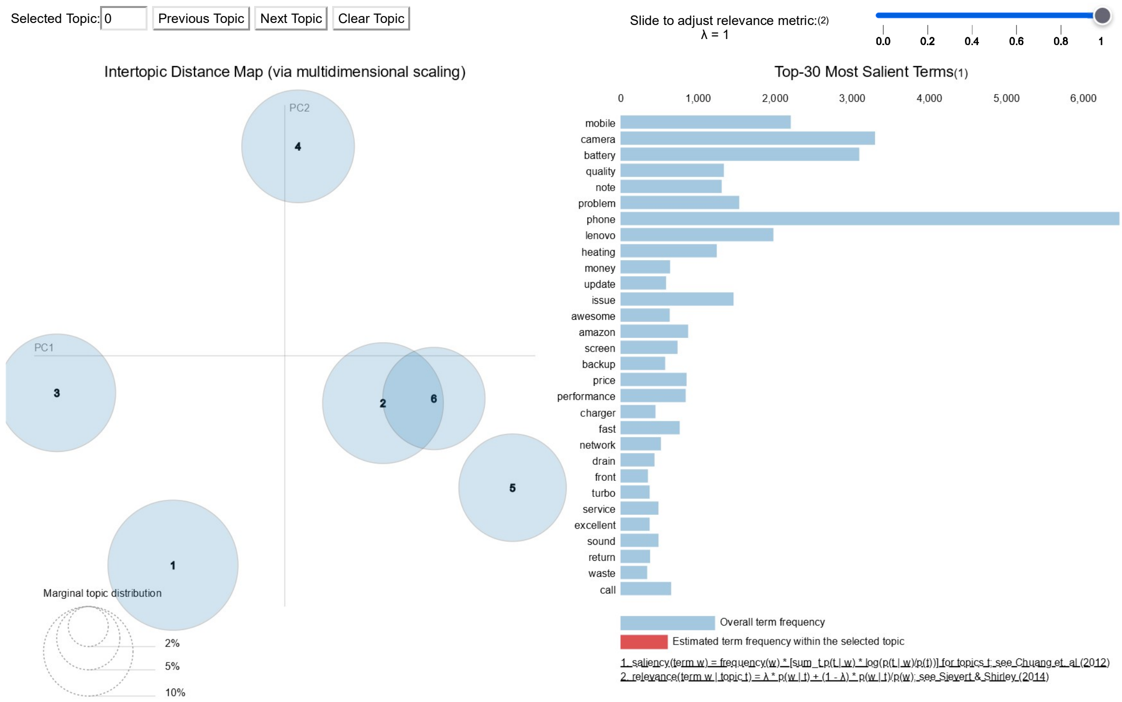
In [57]:
```
ldamodel.print_topics()
```

Out[57]:
```
[(0,
  '0.156*"camera" + 0.078*"quality" + 0.039*"phone" + 0.022*"sound" + 0.021*"front" + 0.018*"mode" + 0.015*"depth" + 0.014*"performance" + 0.014*"display" + 0.014*"rear"'
 (1,
  '0.082*"note" + 0.070*"lenovo" + 0.039*"phone" + 0.025*"call" + 0.024*"feature" + 0.017*"android" + 0.016*"product" + 0.012*"option" + 0.011*"speaker" + 0.011*"stock"')
 (2,
  '0.154*"mobile" + 0.084*"problem" + 0.046*"heating" + 0.038*"product" + 0.038*"amazon" + 0.034*"issue" + 0.022*"return" + 0.018*"network" + 0.015*"lenovo" + 0.013*"tim
e"'),
 (3,
  '0.076*"phone" + 0.036*"money" + 0.034*"screen" + 0.025*"charger" + 0.022*"product" + 0.021*"lenovo" + 0.021*"turbo" + 0.019*"waste" + 0.014*"amazon" + 0.013*"value"'),
 (4,
  '0.086*"phone" + 0.040*"update" + 0.038*"issue" + 0.026*"problem" + 0.025*"service" + 0.022*"lenovo" + 0.021*"network" + 0.021*"software" + 0.013*"volta" + 0.012*"call"
 (5,
  '0.135*"battery" + 0.126*"phone" + 0.035*"price" + 0.030*"awesome" + 0.030*"camera" + 0.030*"fast" + 0.029*"performance" + 0.027*"backup" + 0.023*"product" + 0.022*"hea
g"')]
```

In [60]:
```
pyLDAvis.gensim_models.prepare(ldamodel,doc_term_matrix,dictionary)
```

```
/usr/local/lib/python3.8/dist-packages/pyLDAvis/_prepare.py:246: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'lab
' will be keyword-only
  default_term_info = default_term_info.sort_values(
```

Out[60]:


Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
4
3
2
6
5
1
Marginal topic distribution
2%
5%
10%
Top-30 Most Salient Terms(1)
0
1,000
2,000
3,000
4,000
5,000
6,000
mobile
camera
battery
quality
note
problem
phone
lenovo
heating
money
update
issue
awesome
amazon
screen
backup
price
performance
charger
fast
network
drain
front
turbo
service
excellent
sound
return
waste
call
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

In [61]:
```
df.head()
```

Out[61]:

	review	clean_review
0	Good but need updates and improvements	[need, update, improvement]
1	Worst mobile i have bought ever, Battery is dr...	[mobile, bought, battery, hell, backup, hour, ...
2	when I will get my 10% cash back.... its alrea...	[cash, january]
3	The worst phone everThey have changed the last...	[phone, everthey, phone, problem, amazon, phon...
4	Only I'm telling don't buyI'm totally disappoi...	[buyi, disappointedpoor, batterypoor, camerawa...

Creating a lookup table for topics

In [62]:
```
topic_lookup_data = pd.DataFrame((ldamodel.print_topics()),columns=['Topic_Number','Top_Keywords'])
topic_lookup_data['Topic_Name'] = ['Camera, Sound','Mixed issues','Heating issue','turbo charger','Connectivity','Battery']
topic_lookup_data = topic_lookup_data[['Topic_Number','Topic_Name','Top_Keywords']]
topic_lookup_data['Top_Keywords'] = topic_lookup_data.Top_Keywords.str\
.replace(r'[^a-z]',' ',regex=True).apply(lambda x: x.split())
topic_lookup_data.style.set_properties(subset=['Top_Keywords'], **{'width': '300px'})
```

Out[62]:

	Topic_Number	Topic_Name	Top_Keywords
0	0	Camera, Sound	['camera', 'quality', 'phone', 'sound', 'front', 'mode', 'depth', 'performance', 'display', 'rear']
1	1	Mixed issues	['note', 'lenovo', 'phone', 'call', 'feature', 'android', 'product', 'option', 'speaker', 'stock']
2	2	Heating issue	['mobile', 'problem', 'heating', 'product', 'amazon', 'issue', 'return', 'network', 'lenovo', 'time']
3	3	turbo charger	['phone', 'money', 'screen', 'charger', 'product', 'lenovo', 'turbo', 'waste', 'amazon', 'value']
4	4	Connectivity	['phone', 'update', 'issue', 'problem', 'service', 'lenovo', 'network', 'software', 'volta', 'call']
5	5	Battery	['battery', 'phone', 'price', 'awesome', 'camera', 'fast', 'performance', 'backup', 'product', 'heating']

Creating new columns and inserting topic numbers and names

In [63]:
```
for index,sent in enumerate(ldamodel[doc_term_matrix]):
  topic_num =[]
  topic_details = sorted(sent,key=lambda x: x[1], reverse=True)[:2] # Getting top 2 topics in descending order
  topic_num.append(topic_details[0][0]) # Appending top topic
  if len(topic_details) > 1:
    if topic_details[1][1] > 0.35: # Appending second topic only if it has more than 35% influence on current row
      topic_num.append(topic_details[1][0])
  df.loc[index,'Topic_Number'] = ','.join(str(x) for x in sorted(topic_num))
```

In [65]:
```
for index,topic_num in enumerate(df.Topic_Number):
  topic_name_list=[]
  for single_topic_num in topic_num.split(','):
    single_topic_num=int(single_topic_num)
    topic_name_list.append(topic_lookup_data.loc\
                           [topic_lookup_data.Topic_Number == single_topic_num,'Topic_Name'][single_topic_num])
  # Extracting topic names from lookup table
  df.loc[index,'Topic_Name'] =' & '.join(topic_name_list)
```

In [66]:
```
df.head()
```

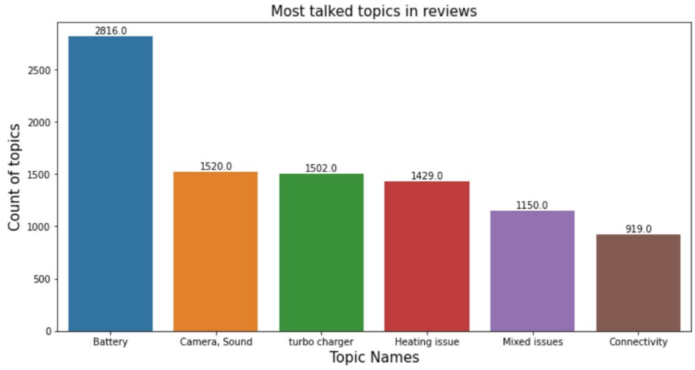
Out[66]:

	review	clean_review	Topic_Number	Topic_Name
0	Good but need updates and improvements	[need, update, improvement]	0,4	Camera, Sound & Connectivity
1	Worst mobile i have bought ever, Battery is dr...	[mobile, bought, battery, hell, backup, hour, ...	3	turbo charger
2	when I will get my 10% cash back.... its alrea...	[cash, january]	0	Camera, Sound
3	The worst phone everThey have changed the last...	[phone, everthey, phone, problem, amazon, phon...	3	turbo charger
4	Only I'm telling don't buyI'm totally disappoi...	[buyi, disappointedpoor, batterypoor, camerawa...	4	Connectivity

Visualization

In [67]:
```
import seaborn as sns
```

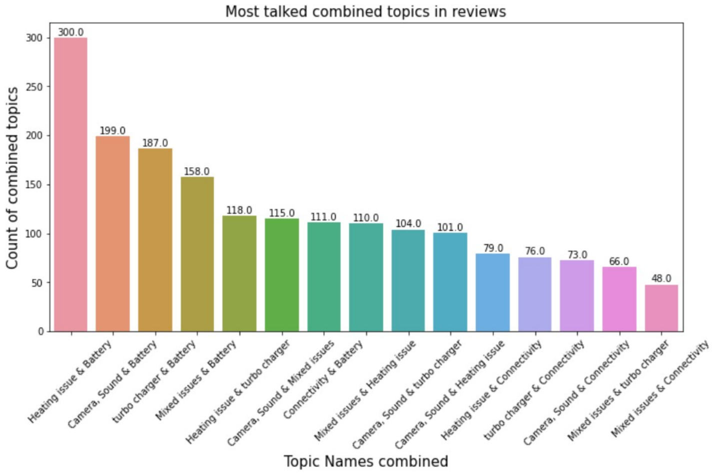
In [68]:
```
plt.figure(figsize=(12,6))
ax = sns.barplot(x=df.Topic_Name.value_counts()[:6].index,y=df.Topic_Name.value_counts()[:6].values)
for p in ax.patches:
    ax.annotate(p.get_height(), (p.get_x() + p.get_width() / 2., p.get_height()+50),ha = 'center', va = 'center')
plt.xlabel('Topic Names',size=15)
plt.ylabel('Count of topics',size=15)
plt.title('Most talked topics in reviews',size=15)
plt.show()
```



From above graph we can say that most of customers had issues with Battery of mobile

In [69]:

```
plt.figure(figsize=(12,6))
ax = sns.barplot(x=df.Topic_Name.value_counts()[6:].index,y=df.Topic_Name.value_counts()[6:].values)
for p in ax.patches:
    ax.annotate(p.get_height(), (p.get_x() + p.get_width() / 2., p.get_height()+5),ha = 'center', va = 'center')
plt.xlabel('Topic Names combined',size=15)
plt.ylabel('Count of combined topics',size=15)
plt.title('Most talked combined topics in reviews',size=15)
plt.xticks(rotation=45)
plt.show()
```



From above graph we can say that most of customers had combined issues with,

1. Heating issue & Battery
2. Camera, Sound & Battery
3. turbo charger & Battery

In [70]:

```
#Extracting reviews of 5 topic(review of battery)
df.loc[df.Topic_Number.str.contains('5'),['review','Topic_Name']].head(10)\
.style.set_properties(subset=['review'], **{'width': '300px'})
```

Out[70]:

	review	Topic_Name
5	Phone is awesome. But while charging, it heats up allot..Really a genuine reason to hate Lenovo k8 note	Battery
10	Don't purchase this item, It is so much of heating &Battery life is very poor	Heating issue & Battery
12	Very good phone slim good battry backup good screen love it	Battery
15	Battery draining very rapidly I don't know why..Tell me possible solutions for battery life	Heating issue & Battery
17	Excellent camera , excellent speed.excellent features.excelent battery.	Battery
18	It is not a very good product camera are very poor ...Os is not good...Battery draining very quickly...Like a odinary phone..It was fully unexpected product from Lenovo..	Battery
21	Awesome phone in this price and this is my second mobile from lenovo. It is fast and display has been improved.	Heating issue & Battery
24	Before the new update of 8.0 Oreo, it worked superbly, the battery back-up is also superb and there is not that much heating problem...But... After that update, my phone got heating up simply, battery is also draining unnecessarily... really very much disappointed after that update of 8.0 Oreo...😒😒☹	Connectivity & Battery
26	Good performance but the battery gets oveheated	Battery
27	Best camera and better backup is very bestIn this priceFull passa wasole phone	Battery

In [ ]: