**Comcast Telecom Consumer Complaints**

* **Description**

Comcast is an American global telecommunication company. The firm has been providing terrible customer service. They continue to fall short despite repeated promises to improve. Only last month (October 2016) the authority fined them a $2.3 million, after receiving over 1000 consumer complaints.  
The existing database will serve as a repository of public customer complaints filed against Comcast.  
It will help to pin down what is wrong with Comcast's customer service.

|  |  |
| --- | --- |
| **Column Name** | **description** |
| Ticket #: | Ticket number assigned to each complaint |
| Customer Complaint: | Description of complaint |
| Date: | Date of complaint |
| Time: | Time of complaint |
| Received Via: | Mode of communication of the complaint |
| City: | Customer city |
| State: | Customer state |
| Zipcode: | Customer zip |
| Status: | Status of complaint |
| Filing on behalf of someone | Filing on behalf of someone |

* **Data Dictionary**
* Analysis Task

Problem1.0: Load data using pandas

Problem 1.1: Provide the trend chart for the number of complaints at monthly and daily granularity levels.

Problem 1.2: Provide a table with the frequency of complaint types. Which complaint types are maximum i.e., around internet, network issues, or across any other domains.

Problem 1.3: Create a new categorical variable with value as **Open**and **Closed**. Open & Pending is to be categorized as Open and Closed & Solved is to be categorized as Closed.  
 Problem 1.4: Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on:

1. Which state has the maximum complaints
2. Which state has the highest percentage of unresolved complaints

Problem 1.5: Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

Problem 1.1, 1.3, 1.4 are 1.5 are EDA (exploratory data analysis) while Problem 1.2 is a NLP which will solve by Topic modeling approach.

* What is Topic Modeling?

**Topic modeling** is a type of statistical modeling for discovering the abstract “topics” that occur in a collection of documents. Latent Dirichlet Allocation (LDA) is an example of topic model and is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions.

**What LDA does?**

LDA’s approach to topic modeling is it considers each document as a collection of topics in a certain proportion and each topic as a collection of keywords, again, in a certain proportion.

Once you provide the algorithm with the number of topics, all it does it to rearrange the topics distribution within the documents and keywords distribution within the topics to obtain a good composition of topic-keywords distribution.

When I say topic, what is it actually and how it is represented?

A topic is nothing but a collection of dominant keywords that are typical representatives. Just by looking at the keywords, you can identify what the topic is all about.

The following are key factors to obtaining good segregation topics:

1. The quality of text processing.
2. The variety of topics the text talks about.
3. The choice of topic modeling algorithm.
4. The number of topics fed to the algorithm.
5. The algorithms tuning parameters.

**Data Pre-processing for topic modeling**

We will perform the following steps:

* **Tokenization**: Split the text into sentences and the sentences into words. Lowercase the words and remove punctuation.
* All **stopwords** are removed.
* Words are **lemmatized** — words in third person are changed to first person and verbs in past and future tenses are changed into present.
* Words are **stemmed** — words are reduced to their root form.

Steps for prepare model, evaluate and visualize results

## Prerequisites – Download nltk stopwords

We will need the stopwords from NLTK and wordnet for text pre-processing. Later, we will be using the WordNetLemmatizer model for lemmatization.

Lemmatization is nothing but converting a word to its root word. For example: the lemma of the word ‘machines’ is ‘machine’. Likewise, ‘walking’ –> ‘walk’, ‘mice’ –> ‘mouse’ and so on.

## Import Packages

The core packages used in this tutorial are re, gensim, spacy and pyLDAvis. Besides this we will also using matplotlib, numpy and pandas for data handling and visualization.

## Prepare Stopwords

We have already downloaded the stopwords. Let’s import them and make it available in stop\_words.

## Remove newline characters

As you can see there are many emails, newline and extra spaces that is quite distracting. Let’s get rid of them using [regular expressions](https://www.machinelearningplus.com/python/python-regex-tutorial-examples/). After removing the emails and extra spaces, the text still looks messy. It is not ready for the LDA to consume. You need to break down each sentence into a list of words through tokenization, while clearing up all the messy text in the process.

Gensim’s simple\_preprocess is great for this

## Tokenize words and Clean-up text

Tokenize each sentence into a list of words, removing punctuations and unnecessary characters altogether.

Gensim’s simple\_preprocess() is great for this. Additionally I have set deacc=True to remove the punctuations

## Creating Bigram and Trigram Models

Bigrams are two words frequently occurring together in the document. Trigrams are 3 words frequently occurring.

Gensim’s Phrases model can build and implement the bigrams, trigrams, quadgrams and more. The two important arguments to Phrases are min\_count and threshold. The higher the values of these param, the harder it is for words to be combined to bigrams.

## Remove Stopwords, Make Bigrams and Lemmatize

The bigrams model is ready. Now define the functions to remove the stopwords, make bigrams and lemmatization and call them sequentially.

## Create the Dictionary and Corpus needed for Topic Modeling

The two main inputs to the LDA topic model are the dictionary(id2word) and the corpus. We create in this steps

## Building the Topic Model

We have everything required to train the LDA model. In addition to the corpus and dictionary, you need to provide the number of topics as well.

Apart from that, alpha and eta are hyperparameters that affect sparsity of the topics. According to the Gensim docs, both defaults to 1.0/num\_topics prior.chunksize is the number of documents to be used in each training chunk. update\_everydetermines how often the model parameters should be updated and passes is the total number of training passes.

## View the topics in LDA model

LDA model is built with 10 different topics where each topic is a combination of keywords and each keyword contributes a certain weightage to the topic.

You can see the keywords for each topic and the weightage(importance) of each keyword using lda\_model.print\_topics()

## Compute Model Perplexity and Coherence Score

Model **perplexity** and**topic coherenc**e provide a convenient measure to judge how good a given topic model is.

## Visualize the topics-keywords

DA model is built, the next step is to examine the produced topics and the associated keywords. There is a tool pyLDAvis package’s interactive chart and is designed to work well with jupyter notebooks.