Latent Dirichlet Allocation:

LDA is a “generative probabilistic model”. It represents documents as a mixture of topics that spits out words with certain probabilities for belonging to that topic.

LDA goes through all the documents and randomly assigns each word to the K topics (3), giving a random topic representation of the documents and words distribution of each topic.

Assuming all the words, except the word in question is correct, LDA will go through each word in a document and compute the probability of topic t given the document, and the probability of each word given a topic t. Reassign the word to a new topic by multiplying the two probabilities computed.

This is iterated numerous times till a steady state of topic assignments are found.

Assigning Topics to Models Description:

First, we apply our LDA model to our documents corpus which has all the term frequencies and word ids. Then, based on a threshold of 1/3, which is basically 1/ the number of clusters, I made a list of text documents for each topic. Then I checked if the text documents were in any of the topics list and labelled the texts if they were.

Aspect Based Sentiment Analysis Algorithm Description:

To do the aspect-based sentiment analysis, I used 2 algorithms. First, the text was parsed into sentences.

For each sentence, the sentiment score and aspect term were extracted and put into a dictionary as key-value pairs. This was done by checking if words in the sentence belonged in the opinion lexicon and assigning a score of +1 or -1 depending on whether the word was positive or negative. Words with adjectives were given more weight at 1.5, and it also looked for negation terms, and switched the sign based on the negation.

* Pseudo Code:
  + For token in each sentence:
    - Check if the token is in the Opinions Lexicon
      * Assign a +1 if it’s a positive term or -1 if it’s a negative term
      * Add more weight to tokens with adjectives (1.5 for pos, -1.5 for neg)
    - Check if negation on a sentiment term
      * Switch the sign of the score
    - Check for nouns
      * Assign sentiment scores to nouns

Then I used a combination of our Word2Vec model and SVM to assign our aspects and sentiments to the topics. I checked the similarity for each of our terms to our topics and assigned those terms to their respective topics in a dictionary if they were higher than a certain threshold. Otherwise, I assigned the terms based on the ML-SVM model.

* Pseudo Code:
  + For term in sentiment dictionary:
    - If similarity between term and topic using Word2Vec Model is higher than a certain threshold
      * Assign term, sentiment to topic (can be assigned to more than one topic)
    - Otherwise, use the predicted labels from SVM
      * Assign term, sentiment to topic (can be assigned to more than one topic)