Q2.3. Provide a pdf document which contains:

performance comparison between Q1 and Q2.1
describe how you tune the model parameters, e.g. alpha, max_iter etc. in Q2.1.
discuss how effective the method in Q2.2 is to find similar documents, compared with the tfidf weight cosine similarity we used before.

1) performance comparison between Q1 and Q2.1

K-Mean Clustering-

Using K-Mean clustering(both methods) we found the precision and recall to vary widely for the given data.

Using LDA we found the precision and recall both high constantly during every execution. Therefore I found LDA to better at text classification when compared to k-means(both methods).

2) describe how you tune the model parameters, e.g. alpha, max_iter etc

The alpha controls the mixture of topics for any given document. Turn it down, and the documents will likely have less of a mixture of topics. Turn it up, and the documents will likely have more of a mixture of topics

The beta hyperparameter controls the distribution of words per topic. Turn it down, and the topics will likely have less words. Turn it up, and the topics will likely have more words.

Ideally, we want our composites to be made up of only a few topics and our parts to belong to only some of the topics. With this in mind, alpha and beta are typically set below one.

n_components **–** gives the number of topics in the corpus. Should be set as needed.

max_iter gives the maximum number of iterations the algorithm is applied. We should select a value depending on the size of the dataset in order to avoid overfitting and underfitting.

evaluate_every- used to evaluate perplexity. Only used in fit method. set it to 0 or negative number to not evaluate perplexity in training at all. Evaluating perplexity can help you check convergence in training process, but it will also increase total training time. Evaluating perplexity in every iteration might increase training time up to two-fold.

Most of the hyperparameters do not have correct or wrong option and need to be evaluated on a case by case basis on what is required by the problem. If we have no clue about what is to be done we change each parameter little by little every time and see if the performance improves compared to the last run.

3) discuss how effective the method in Q2.2 is to find similar documents, compared with the tfidf weight cosine similarity we used before.

The method used in Q2.2 would give a better because the `topic_mix array` has the probability that each topic belongs to the given topic using LDA. This makes sure that finding the similarity between the documents is accurate.

I.e the major difference is that at tfidf matrix is at the word level. Therefore, a document about a particular topic such as “car” might be far from documents about “traffic” when represented in a tf-idf. But using a LDA we can figure out that those 2 words “traffic” and “car” co-occur in articles and hence are likely to come from the same topic. So the LDA representation of these documents would be close and therefore more similar.