IR Term Project Report Part 2
Sanskar Patni 19CH10046

The contribution I had in the Part 2 of default term project are:

TASK 2A(tf-idf vectorization)

I created the **term frequency and document frequency matrix** for the corpus. The main problem faced here was the size of the tf matrix.Even after reducing it to 20k vocabulary the tf matrix(20k*50k) was too large to be loaded onto RAM. The solution that I came up with was to store only the non-zero values, this way the previous sparces tf matrix was reduced a lot in size.
One more design choice was for **defining the vocabulary**, I chose to take the words which occur in maximum number of documents. Therefore the 20k vocabulary is for the words that occur in most number of documents.

Using the tf and df values, I wrote the code for calculating **tf-idf for lnc.ltc and lnc.lpc scheme**. The problems faced here was dealing with words which were not in tf, I wrote the code such that all these instances are given a 0 value. I wrote the function to normalize the matrix, to make the code more clean.
Finally after calculating the scores for all query-doc combinations, I saved the result(top 50 documents for each query) in respective CSV files.

TASK 2B(Evaluation)

I wrote the code for calculating **MAP@10, MAP@20, NDCG10, NDCG20.**
First I sorted the qRels file, so that I take the query-doc pair with higher iteration value. I then created a dict to store the judgement value for query-doc pair.
I created a function **compute_values** to compute map10, map20, ndcg10, ndcg20 for **any given ranking**.
For calculating MAP, I kept a count of relevant docs and total docs, which I then used in the formula to calculate MAP. For NDCG, I first sorted the given ranking to compute the ideal dcg which I divided to the dcg to give NDCG.
The main problem I faced during this task was to deal with MAP and NDCG logic as was taught in class and dealing with math errors(log zero, divide by zero) due to non relevant documents.