IR TERM PROJECT
<u>REPORT - PART 2</u>

ABHISHEK S. PUROHIT
19CH10002
GROUP 13

My contribution in part 2 DEFAULT-IR term project:

PART 2A (TF-IDF vectorization) :

I developed the code for generating the term frequency of the corpus. Owing to a large set of documents, we faced difficulty in developing the term frequency, as every time we ran out of memory. I have even consulted Shounak sir regarding the same, to which he suggested parallelizing the TF generation, which is what I exactly did. I have written a code, which first stores the file path of all the files present in the CORD-19 folder and then divides it into chunks of 5000 files. Further I have written a code which iterates upon each chunk, processing all the files in that chunk, generating the term frequency, and storing it to a temporary folder, “temp_tf”, and releases the memory before moving to the next chunk. So in this way, I have created 11 chunks of term frequency. Further, I have written a code which takes this chunk one by one and merges them accordingly, keeping RAM usage highly under user’s control. As a design spec, I took chunk size as 5000, as it trades off well between the memory usage and time taken to run the code. I have tried 2000 as well, but in that case, it was taking more time as we needed to go through a lot of iterations saving files and calling python’s garbage collector manually. If we go above 5000, then it may negatively impact the system by consuming a lot of RAM. Later on, I figured out that our TF matrix would be sparse as a lot of vocabulary terms may not be present in a particular doc, but still we were storing that as zero, which is why it was taking a lot of memory, so I corrected that code and our final code runs pretty well. Now using the improved way of generating term frequency using parallelization (for which I am highly thankful to Shounak sir), our code is much more robust and is capable of processing even more than provided 52471 documents. Further I developed the readme file for the project.