Preprocessing strategy:

1. Lowercase
2. Remove Extra Spaces
3. Lemmatize using WordNetLemmatizer()
4. Tokenize using RegexpTokenizer()
5. Remove stopword

Methodology:

In the first question we have implemented tf-idf using the standard algorithm and to compute query similarity we used two approaches - (1) Sum the tf-idf scores (2) Calculate cosine similarity with the one hot encoded query vector. Top matching documents are returned from both approaches.

In question 2, we have performed calculations for DCG, ideal DCG and nDCG. Where nDCG = DCG/iDCG. To calculate the number of documents, we calculate that by multiplying the factorials of number of items in a group.

The calculations in the third question have been done according to the provided formulas and the code for gaussian naive bayes and evaluation metric is written from scratch.

Assumptions:

1. Metric for document similarity is not provided, we used 2 approaches to compare.
2. Lots of garbage characters were throwing errors in UTF-8 encoding, thus we have used windows-1254 encoding and ignored the errors that were arising.

How to Run:

Upload the .pkl files and run, however for Q2 and Q3, the dictionary and file needs to be uploaded separately.