Fashion Search AI Project Documentation

- I downloaded Myntra dataset from kaggle.
  Link https://www.kaggle.com/datasets/djagatiya/myntra-fashion-product-dataset
- Dataset has 14214 rows and 11 columns. Two columns with more than 50% Null values are dropped.
- Column "description" has been sanitized using regular expressions.
- Data in the column has been converted to lower case.
- As all the data is in html format, all the tags have been removed.
- Multiple <space> have been replaced with single <space>.
- Punctuations that do not contribute to the model are removed.
- NLTK library is used for all the columns to tokenize the data to see the most frequent words and any unnecessary words that can be removed.
- For p_attributes column, html tags have been removed.
- All the data in the column has been converted to lowercase.
- As this column data are dictionaries converted to strings, keys "body shape id", "body or garment size" have been removed as they do not contribute.
- Data in columns "name" and "p_attributes " have been added to their concerned description columns.
- Columns "products", "price", "colour", "brand" are converted to dictionaries for with each column name as key and its corresponding values as keys.
- This is repeated for each row. Resultant dictionaries are added as metadata column.
- "name", "p_attributes ", "products", "price", "colour", "brand" columns have been removed from dataframe as they are redundant now.
- For embedding, openai model available for chromadb is used.
- Threshold is chosen as 0.15 as anything more than that is considering products that are not very similar to the query.
- For embedding model, a persistant client is chosen as it will avoid api calling everytime a new session is opened.
- For cache, a volatile client is chosen. Cache collection is created as an array of length 5 to store 5 latest queries.
- When a new query is received, all the five caches are checked to see if a match can be found.
- If not found, query is forwarded to vector store to retrieve info and it is saved in cache with least recent query.
- When query is received, the data sent by LLM model, is formatted to suit the query structure of chromadb and forwarded to the query.

- Once the dataframe is created with results received from the query, the dataframe is forwaded to LLM model to display the results in the specified format.