

Project Report

Myntra Fashion Helper

Your Name

Aparana Bhatt

Table of Contents

1. Ideation
2. Data details
3. Implementation Details
4. System Design
5. Challenges and Future Scope
6. Conclusion

1. Ideation

There are plethora of clothes products available online for women with various options.

We do have multiple options to explore via our applications like myntra,amazon etc.

If we know the correct keyword , we can search for good products with less effort but if we have a requirement and our search query is very long then we do not get required suggestions and end up wasting a lot of time searching for the right product.

This is where fashion helpmate can be of extreme use because we can write our needs to it and it will understand and return good suggestions in a short time and we would need less time to search for the product online.

It will greatly enhance our shopping experience.

2. All About Data

I have used a publicly available Myntra dataset and sampled it according to the project requirement. The data set is available at [Myntra Fashion Product Dataset \(kaggle.com\)](https://www.kaggle.com/datasets/myntra-fashion-product-dataset)

Data consists of a list of women products with description, name, brand and other metadata . Some of which have been used to create a minimal instruction/decision set for the helpmate to take help with.

```
In [15]: import pandas as pd;

df=pd.read_csv(content_path)

df.head()
```

Out[15]:

Unnamed: 0	p_id	name	price	colour	brand	img	ratingCount	avg_rating	description	p_attributes
0	17048614.0	Khushal K Women Black Ethnic Motifs Printed Kurta	5099.0	Black	Khushal K	http://assets.myntassets.com/assets/images/170...	4522.0	4.418399	Black printed Kurta with Palazzos with dupatta...	{'Add-Ons': 'NA', 'Body Shape ID': '443,333,32...
1	16524740.0	InWeave Women Orange Solid Kurta with Palazzos	5899.0	Orange	InWeave	http://assets.myntassets.com/assets/images/165...	1081.0	4.119334	Orange solid Kurta with Palazzos with dupatta<...	{'Add-Ons': 'NA', 'Body Shape ID': '443,333,32...
2	16331376.0	Anubhutee Women Navy Blue Ethnic Motifs Embroi...	4899.0	Navy Blue	Anubhutee	http://assets.myntassets.com/assets/images/163...	1752.0	4.161530	Navy blue embroidered Kurta with Trousers with...	{'Add-Ons': 'NA', 'Body Shape ID': '333,424', ...}
3	14709966.0	Nayo Women Red Floral Printed Kurta With Trous...	3699.0	Red	Nayo	http://assets.myntassets.com/assets/images/147...	4113.0	4.088986	Red printed kurta with trouser and dupatta ...	{'Add-Ons': 'NA', 'Body Shape ID': '333,424', ...}
4	11056154.0	AHIKA Women Black & Green Printed Straight Kurta	1350.0	Black	AHIKA	http://assets.myntassets.com/assets/images/110...	21274.0	3.978377	Black and green printed straight kurta, has a ...	{'Body Shape ID': '424', 'Body or Garment Size': ...}

```
In [40]: ## filter rows where description is not string type.
df.loc[df['description'].apply(lambda x: isinstance(x, str)), 'Text_Length'] = df['description'].apply(lambda x: len(x.split(' '))
```

The data is processed and sampled to 500 entries only for simplicity. We have also filtered the rows having empty description columns as description is the major source of semantic context in the process. We have also added MetaData column, text length column into the dataset during processing and saved this sampled data in a final csv file, which is being used in the next steps.

```
In [40]: ## filter rows where description is not string type.
df.loc[df['description'].apply(lambda x: isinstance(x, str)), 'Text_Length'] = df['description'].apply(lambda x: len(x.split(' ')))
df.head()
```

Out[40]:

	Unnamed: 0	p_id	name	price	colour	brand	img	ratingCount	avg_rating	description	p_attribut
0	0	17048614.0	Khushal K Women Black Ethnic Motifs Printed Kurta	5099.0	Black	Khushal K	http://assets.myntassets.com/assets/images/170...	4522.0	4.418399	Black printed Kurta with Palazzos with dupatta...	{'Add-Or': 'NA', 'Bc Shape I': '443,333,3;
1	1	16524740.0	InWeave Women Orange Solid Kurta with Palazzos...	5899.0	Orange	InWeave	http://assets.myntassets.com/assets/images/165...	1081.0	4.119334	Orange solid Kurta with Palazzos with dupatta<...	{'Add-Or': 'NA', 'Bc Shape I': '443,333,3;
2	2	16331376.0	Anubhutee Women Navy Blue Ethnic Motifs Embroi...	4899.0	Navy Blue	Anubhutee	http://assets.myntassets.com/assets/images/163...	1752.0	4.161530	Navy blue embroidered Kurta with Trousers with...	{'Add-Or': 'NA', 'Bc Shape I': '333,424';
3	3	14709966.0	Nayo Women Red Floral Printed Kurta With Trous...	3699.0	Red	Nayo	http://assets.myntassets.com/assets/images/147...	4113.0	4.088986	Red printed kurta with trouser and dupatta ...	{'Add-Or': 'NA', 'Bc Shape I': '333,424';
4	4	11056154.0	AHIKA Women Black & Green Printed Straight Kurta	1350.0	Black	AHIKA	http://assets.myntassets.com/assets/images/110...	21274.0	3.978377	Black and green printed straight kurta, has a ...	{'Body Sha ID': '42', 'Body Garm Sizi

3. Implementation Details

The code is written in python and various other components and apis used are mentioned below

- 1) Python3
- 2) Openai library
- 3) Chat completions API
- 4) Chroma DB
- 5) cross-encoder/ms-marco-MiniLM-L-6-v2 cross encoder for reranking
- 6) Gpt-3.5 Turbo used for recurring chats
- 7) Jupyter notebook

4. System Design

System design contains below major layers and steps for a successful response.

- 1) **Data Layer** : This layer is the first layer and first step where relevant data is sourced and preprocessed.
- 2) **Semantic Search Layer**: We use chroma db as a semantic search store here. This is where the document embeddings are stored pooled from the data layer after preprocessing.
- 3) **Cache Layer**: We also have a cache layer which stores the response for all the searched queries, so that when a semantic similar query is asked , it directly returns search result for reranking, reducing processing cost of the system.
- 4) **Reranking Layer**: Here we use cross encoder to rank the results in the semantic search layer .The documents are compared pairwise for the results and relative scores or ranks are found out . These outputs help to sort the results on the basis of relevancy.
- 5) **Generation Layer**: This layer gets as an input the final filtered results after semantic search and reranking. This result is then passed on for chat generation using chat completion API . This layer generates the final response to be shared to the user. It uses one of the prompting techniques for a suitable response for the user.

Code overview and helpmate example

Below is the overview of main methods used in the code which are empowering the Movie Assistant .

- 1) **Create batches for chroma db data addition:** Since openAI api has TPS limit set per minute, it throws error when passed TPS/min exceeds the limit . For the data to be added in the chromadb , it was converted in the batches and with some delay introduction all the batches were processed to add data to the chroma collection.

```
:  
## this method creates 50 batches of sampled data and then adds to chroma db collection . The openAi has token limit per minute  
## restriction hence added some sleep time to process all the data in some batches.  
  
def create_batches(doc_list, metadata_list, batch_size):  
    for i in range(0, len(doc_list), batch_size):  
        yield doc_list[i:i + batch_size], metadata_list[i:i + batch_size]  
  
# Example usage:  
doc_list = documents_list # Replace with your actual docList  
metadata_list = metadata_list  
  
batch_size = 10  
batches = list(create_batches(doc_list, metadata_list, batch_size))  
currentCount=0  
# Displaying the batches  
for batch_number, (doc_batch, metadata_batch) in enumerate(batches, 1):  
    fashionData.add(  
        documents= doc_batch,  
        ids = [str(i) for i in range(currentCount, currentCount+len(doc_batch))],  
        metadatas = metadata_batch  
    )  
    currentCount+=len(doc_batch)+1  
    print("added batch "+ str(batch_number))  
    time.sleep(10)
```

- 2) **Search process:** Search process is 2 step , first we search in the cache collection if the query gives a hit on the cache collection, we return the data from there for reranking , else we search in the main collection called “myntara_fashion”.

```

def searchInCache(query,cache_collection):
    cache_results = cache_collection.query(
        query_texts=query,
        n_results=1
    )
    return cache_results

def searchData(query,threshold,cache_collection,results_df):

    # If the distance is greater than the threshold, then return the results from the main collection.
    cache_results=searchInCache(query,cache_collection)

    if cache_results['distances'][0] == [] or cache_results['distances'][0][0] > threshold:
        # Query the collection against the user query and return the top 10 results
        results = fashionData.query(
            query_texts=query,
            n_results=10
        )

        # Store the query in cache_collection as document w.r.t to ChromaDB so that it can be embedded and searched against Later
        # Store retrieved text, ids, distances and metadatas in cache_collection as metadatas, so that they can be fetched easily
        Keys = []
        Values = []
        #print(results)

        for key, val in results.items():
            if key != 'embeddings':
                for i in range(len(results)):
                    if val is not None:
                        Keys.append(str(key)+str(i))
                        # print(key)
                        Values.append(str(val[0][i]))

        cache_collection.add(
            documents= [query],
            ids = [query], # Or if you want to assign integers as IDs 0,1,2,.., then you can use "len(cache_results['documents'])"
            metadatas = dict(zip(Keys, Values))
        )

        print("Not found in cache. Found in main collection.")

    result dict = {'Metadatas': results['metadatas'][0], 'Documents': results['documents'][0], 'Distances': results['distances'][0]}

```

- 3) **Reranking Process:** Once the data is searched , it is send to the cross encoder for generating relative relevance scores of the results. One the data scores are generated, the data is sorted on the basis of relevance scores in descending order. Then top 3 result from that result is selected and sent as an input to prompt for text generation.

- 4) **Generation Process:**After reranking , we send the data as a prompt for final result generation.

6. Retrieval Augmented Generation

Now that we have the final top search results, we can pass it to an GPT 3.5 along with the user query and a well-engineered prompt, to generate a direct answer to the query along with citations, rather than returning whole pages/chunks.

```
In [117]: # Define the function to generate the response. Provide a comprehensive prompt that passes the user query and the top 3 results to the GPT-3.5 model.

def generate_response(query, results_df):
    """
    Generate a response using GPT-3.5's ChatCompletion based on the user query and retrieved information.
    """
    messages = [
        {"role": "system", "content": "You are a helpful assistant in the women fashion domain who can effectively answer user queries."},
        {"role": "user", "content": f"""You are a helpful assistant in the women fashion domain who can effectively answer user queries.
        You have a question asked by the user in '{query}' and you have some search results from a search engine.

        The column 'documents' inside this dataframe contains the actual description of the products.
        Use the documents in '{top_3_RAG}' to answer the query '{query}'. Frame an informative response.

        Follow the guidelines below when performing the task.
        1. Try to provide relevant/accurate numbers if available.
        2. You don't have to necessarily use all the information in the dataframe. Only use the relevant information.
        3. If the document text has tables with relevant information, please reformat the table into a markdown table.
        3. Use the Metadatas columns in the dataframe to retrieve and cite the product name and price.
        4. If you can't provide the complete answer, please also provide any information you have.
        5. You are a customer facing assistant, so do not provide any information on internal processes.

        The generated response should answer the query directly addressing the user and include citations.
        """,
        }

    ]

    response = openai.chat.completions.create(
        model="gpt-3.5-turbo",
        messages=messages
    )

    return response.choices[0].message.content.split('\n')
```

helpmate example

Query 1 Snapshots

```
n [106]: query = input()

results_df=searchData(query,threshold,cache_collection,results_df)

i am looking for a red kurta
Not found in cache. Found in main collection.
```

```
n [107]: results_df
```

```
ut[107]:
```

	Metadatas	Documents	Distances	IDs
0	{'brand': 'Anouk', 'color': 'Red', 'name': 'An...	Red and blue yoke design kurta with dhoti pant...	0.249005	392
1	{'brand': 'Libas', 'color': 'Black', 'name': '...	Black and red embroidered A-line kurti, has a ...	0.266093	129
2	{'brand': 'DIVYANK', 'color': 'Red', 'name': '...	 Red and beige straight kurti <...	0.272927	53
3	{'brand': 'Yuris', 'color': 'Purple', 'name': '...	Purple printed Kurta with Trousers with dupatt...	0.276227	478
4	{'brand': 'Ishin', 'color': 'Pink', 'name': 'I...	Pink printed Kurta with Sharara with dupatta <...	0.277330	423
5	{'brand': 'PANIT', 'color': 'Yellow', 'name': '...	Yellow and black a-line kurti...	0.279540	524
6	{'brand': 'Inddus', 'color': 'Pink', 'name': '...	Pink embroidered kurta with palazzos and dupat...	0.280090	517
7	{'brand': 'Biba', 'color': 'White', 'name': 'B...	White embroidered Kurta with Churidar with dup...	0.280805	284
8	{'brand': 'Anouk', 'color': 'Blue', 'name': 'A...	Blue self design Kurta with Trousers ...	0.281407	149
9	{'brand': 'heemara', 'color': 'Maroon', 'name': '...	Maroon printed Kurta with Trousers with dupatt...	0.282169	122

```
In [118]: # Generate the response

response = generate_response(query, top_3_RAG)
```

```
In [119]: # Print the response

print("\n".join(response))
```

Sure! I can help you find a red kurta. Here are a few suggestions for you:

1. Anouk Red and Blue Yoke Design Kurta with Dhoti Pants - This kurta features a red and blue yoke design and comes with matching dhoti pants. It is perfect for a traditional yet stylish look. Brand: Anouk
2. Libas Black and Red Embroidered A-line Kurti - This A-line kurti is black with red embroidery, giving it a unique and elegant look. It can be paired with leggings or jeans for a trendy outfit. Brand: Libas
3. Divyank Red and Beige Straight Kurti - This straight kurti is red with beige detailing. It is a versatile piece that can be dressed up or down for different occasions. Brand: Divyank

You can find more details about these kurtas on our website. Let me know if there's anything else I can assist you with!

Citations:

1. Product: Anouk Red and Blue Yoke Design Kurta with Dhoti Pants, Brand: Anouk
2. Product: Libas Black and Red Embroidered A-line Kurti, Brand: Libas
3. Product: Divyank Red and Beige Straight Kurti, Brand: Divyank

```
In [112]: # Return the top 3 results from semantic search
```

```
top_3_semantic = results_df.sort_values(by='Distances')
top_3_semantic[:3]
```

Out[112]:

	Metadatas	Documents	Distances	IDs	Reranked_scores
0	{'brand': 'Anouk', 'color': 'Red', 'name': 'An...	Red and blue yoke design kurta with dhoti pant...	0.249005	392	1.598063
1	{'brand': 'Libas', 'color': 'Black', 'name': '...	Black and red embroidered A-line kurti, has a ...	0.266093	129	0.525208
2	{'brand': 'DIVYANK', 'color': 'Red', 'name': '...	 Red and beige straight kurti <...	0.272927	53	-2.181401

```
In [114]: # Return the top 3 results after reranking
```

```
top_3_rerank = results_df.sort_values(by='Reranked_scores', ascending=False)
top_3_rerank[:3]
```

Out[114]:

	Metadatas	Documents	Distances	IDs	Reranked_scores
0	{'brand': 'Anouk', 'color': 'Red', 'name': 'An...	Red and blue yoke design kurta with dhoti pant...	0.249005	392	1.598063
1	{'brand': 'Libas', 'color': 'Black', 'name': '...	Black and red embroidered A-line kurti, has a ...	0.266093	129	0.525208
2	{'brand': 'DIVYANK', 'color': 'Red', 'name': '...	 Red and beige straight kurti <...	0.272927	53	-2.181401

```
In [115]: top_3_RAG = top_3_rerank[["Documents", "Metadatas"]][:3]
```

```
In [116]: top_3_RAG
```

Out[116]:

	Documents	Metadatas
0	Red and blue yoke design kurta with dhoti pant...	{'brand': 'Anouk', 'color': 'Red', 'name': 'An...
1	Black and red embroidered A-line kurti, has a ...	{'brand': 'Libas', 'color': 'Black', 'name': '...
2	 Red and beige straight kurti <...	{'brand': 'DIVYANK', 'color': 'Red', 'name': '...

Query 2 Snapshots

In [120]:

```
query = input()

results_df=searchData(query,threshold,cache_collection,results_df)
```

Find me an attractive western dress
Not found in cache. Found in main collection.

In [121]:

```
results_df
```

Out[121]:

	Metadatas	Documents	Distances	IDs
0	{'brand': 'Rang Gali', 'color': 'Navy Blue', '...	Highlight your stylish side by donning this cl...	0.389410	346
1	{'brand': 'INDYA', 'color': 'Black', 'name': '...	 Black crop fitted top Eth...	0.412692	225
2	{'brand': 'Shae by SASSAFRAS', 'color': 'Blue'...	Blue and gold-toned crop fitted top</l...	0.412736	411
3	{'brand': 'STREET 9', 'color': 'White', 'name'...	White and black checked culotte jumpsuit with ...	0.413677	206
4	{'brand': 'KASSUALLY', 'color': 'Black', 'name'...	Trendy and versatile, this playsuit is a worth...	0.416538	492
5	{'brand': 'Jompers', 'color': 'Red', 'name': '...	Red and off-white striped basic jumpsuit with ...	0.418487	108
6	{'brand': 'Shae by SASSAFRAS', 'color': 'Green'...	Green and gold-toned printed lehenga choli,<br...	0.420346	75
7	{'brand': 'Libas', 'color': 'Black', 'name': '...	Black and red embroidered A-line kurti, has a ...	0.421255	129
8	{'brand': 'Myshka', 'color': 'Black', 'name': '...	Black and white striped straight kurta, has a ...	0.421427	506
9	{'brand': 'PANIT', 'color': 'Yellow', 'name': '...	Yellow and black a-line kurti...	0.424118	524

In [130]:

```
# Generate the response

response = generate_response(query, top_3_RAG)
```

In [131]:

```
# Print the response

print("\n".join(response))
```

I found some attractive western dress suggestions for you:

1. Rang Gali Navy Blue Dress:
 - Description: Highlight your stylish side by donning this classic navy blue dress. It is made from high-quality fabric to ensure maximum comfort.
 - Brand: Rang Gali
2. KASSUALLY Black Playsuit:
 - Description: This trendy and versatile black playsuit is a worthy addition to your wardrobe. It can be dressed up or down for different occasions.
 - Brand: KASSUALLY
3. Shae by SASSAFRAS Blue Crop Fitted Top:
 - Description: This trendy blue crop fitted top is a stylish addition to your western dress collection. It can be paired with jeans, skirts, or shorts for a chic look.
 - Brand: Shae by SASSAFRAS

Please let me know if you need more information or if there's anything else I can help you with!

In []:

```
In [126]: # Return the top 3 results from semantic search
```

```
top_3_semantic = results_df.sort_values(by='Distances')
top_3_semantic[:3]
```

```
Out[126]:
```

	Metadatas	Documents	Distances	IDs	Reranked_scores
0	{'brand': 'Rang Gali', 'color': 'Navy Blue', '...	Highlight your stylish side by donning this cl...	0.389410	346	-3.770736
1	{'brand': 'INDYA', 'color': 'Black', 'name': '...	 Black crop fitted top Eth...	0.412692	225	-11.266121
2	{'brand': 'Shae by SASSAFRAS', 'color': 'Blue'...	Blue and gold-toned crop fitted top</l...	0.412736	411	-11.190517

```
In [127]: # Return the top 3 results after reranking
```

```
top_3_rerank = results_df.sort_values(by='Reranked_scores', ascending=False)
top_3_rerank[:3]
```

```
Out[127]:
```

	Metadatas	Documents	Distances	IDs	Reranked_scores
0	{'brand': 'Rang Gali', 'color': 'Navy Blue', '...	Highlight your stylish side by donning this cl...	0.389410	346	-3.770736
4	{'brand': 'KASSUALLY', 'color': 'Black', 'name'...	Trendy and versatile, this playsuit is a worth...	0.416538	492	-9.323263
2	{'brand': 'Shae by SASSAFRAS', 'color': 'Blue'...	Blue and gold-toned crop fitted top</l...	0.412736	411	-11.190517

```
In [128]: top_3_RAG = top_3_rerank[["Documents", "Metadatas"]][:3]
```

```
In [129]: top_3_RAG
```

```
Out[129]:
```

	Documents	Metadatas
0	Highlight your stylish side by donning this cl...	{'brand': 'Rang Gali', 'color': 'Navy Blue', '...
4	Trendy and versatile, this playsuit is a worth...	{'brand': 'KASSUALLY', 'color': 'Black', 'name'...
2	Blue and gold-toned crop fitted top</l...	{'brand': 'Shae by SASSAFRAS', 'color': 'Blue'...

Query 3 Snapshots

In [144]:

```
query = input()

results_df=searchData(query,threshold,cache_collection,results_df)

help me find red or white top
Not found in cache. Found in main collection.
```

In [145]:

```
results_df
```

Out[145]:

	Metadatas	Documents	Distances	IDs
0	{'brand': 'Colour Me by Melange', 'color': 'Wh...	 White regular top Solid <...	0.336761	389
1	{'brand': 'Bhama Couture', 'color': 'Red', 'na...	Red solid woven A-line top with gathered and e...	0.353052	152
2	{'brand': 'Remanika', 'color': 'Red', 'name': '...	 Red and cherry tomato crop shirt sty...	0.354479	169
3	{'brand': 'DressBerry', 'color': 'White', 'nam...	White and Black striped woven regular top, ha...	0.361686	113
4	{'brand': 'max', 'color': 'Yellow', 'name': 'm...	Yellow regular a-line topSelf...	0.362786	457
5	{'brand': 'AND', 'color': 'Blue', 'name': 'AND...	This clothing set consists of top and trousers...	0.366097	324
6	{'brand': 'Divena', 'color': 'Yellow', 'name': '...	Peach-coloured, pink and white re...	0.366126	28
7	{'brand': 'RARE', 'color': 'White', 'name': 'R...	White solid woven regular top, has a square ne...	0.367724	319
8	{'brand': 'Bhama Couture', 'color': 'Mustard', '...	Mustard yellow and white regular top</l...	0.367888	130
9	{'brand': 'MANGO', 'color': 'White', 'name': '...	White regular boxy apron topS...	0.367929	194

In [154]:

```
# Generate the response

response = generate_response(query, top_3_RAG)
```

In [155]:

```
# Print the response

print("\n".join(response))

Sure! I can help you find a red or white top. Here are some options for you:

1. Red Solid Woven A-line Top by Bhama Couture
- This top is red in color, with a solid woven fabric. It has a gathered and elasticated detail and a round neck.

2. White and Black Striped Woven Regular Top by DressBerry
- This top is white with black stripes. It is made of woven fabric and has a regular fit.

3. White Solid Woven Regular Top by RARE
- This top is white in color and made of solid woven fabric. It has a square neckline and a regular fit.

Please let me know if you need more information or if you would like to see any other options.

Citations:
- Red Solid Woven A-line Top: Bhama Couture
- White and Black Striped Woven Regular Top: DressBerry
- White Solid Woven Regular Top: RARE
```

In []:

Out[150]:

	Metadatas	Documents	Distances	IDs	Reranked_scores
0	{'brand': 'Colour Me by Melange', 'color': 'Wh...	 White regular top Solid <...	0.336761	389	-5.311004
1	{'brand': 'Bhama Couture', 'color': 'Red', 'na...	Red solid woven A-line top with gathered and e...	0.353052	152	-3.825470
2	{'brand': 'Remanika', 'color': 'Red', 'name': '...	 Red and cherry tomato crop shirt sty...	0.354479	169	-7.225606

In [151]: *# Return the top 3 results after reranking*

```
top_3_rerank = results_df.sort_values(by='Reranked_scores', ascending=False)
top_3_rerank[:3]
```

Out[151]:

	Metadatas	Documents	Distances	IDs	Reranked_scores
1	{'brand': 'Bhama Couture', 'color': 'Red', 'na...	Red solid woven A-line top with gathered and e...	0.353052	152	-3.825470
3	{'brand': 'DressBerry', 'color': 'White', 'nam...	White and Black striped woven regular top, ha...	0.361686	113	-3.984451
7	{'brand': 'RARE', 'color': 'White', 'name': 'R...	White solid woven regular top, has a square ne...	0.367724	319	-4.162737

In [152]: `top_3_RAG = top_3_rerank[["Documents", "Metadatas"]][:3]`

In [153]: `top_3_RAG`

Out[153]:

	Documents	Metadatas
1	Red solid woven A-line top with gathered and e...	{'brand': 'Bhama Couture', 'color': 'Red', 'na...
3	White and Black striped woven regular top, ha...	{'brand': 'DressBerry', 'color': 'White', 'nam...
7	White solid woven regular top, has a square ne...	{'brand': 'RARE', 'color': 'White', 'name': 'R...

4. Challenges and Future Scope

Challenges

- 1) Identifying minimal subset that would yield proper data for semantic search documents in the dataset .
- 2) The support code was written on the older openai version, the common modules using openai apis had to be rewritten to handle response structure and api call changes.
- 3) The score generation and query response is limited by the dataset size and user requirement. Since the sample dataset is small, the query suggestions are not much satisfactory however working well with respect to the kind of content that is present.
- 4) Frequent data updates in the chroma db database for very large datasets is taxing.

Future Scope

- 1) Store user shopping patterns and when the user logs in provide users with similar suggestions even if the user has not yet searched anything. This approach can be used to also refine user likes and dislikes.
- 2) We can also include price in search response in future so that we can provide user with more refined searches. However since price keeps changing , we should not make it part of embeddings but of metadata.

5. Conclusion

This project showcases the utility of semantic search combined with the power of chat api to improve and enhance user experience to more humane and also reduce online shopping time expenditure of the user .