E-commerce Product Analysis and Recommendation System

Introduction:

Welcome to the world of electronic commerce, more commonly known as e-commerce. In recent years, e-commerce has revolutionized the way we shop, conduct business, and interact with goods and services. This digital marketplace transcends geographical boundaries, offering consumers and businesses unprecedented convenience, accessibility, and flexibility in buying and selling. E-commerce has come a long way since its inception in the early 1990s. What began as simple online transactions has evolved into a sophisticated ecosystem encompassing a diverse array of platforms, technologies, and business models. From online retail giants like Amazon and Alibaba to niche boutique stores and digital marketplaces, e-commerce has democratized commerce, empowering businesses of all sizes to reach global audiences and thrive in the digital economy.

The e-commerce Product Analysis and Recommendation System represents a comprehensive solution tailored to elevate user engagement and streamline product organization within an e-commerce platform. Through the integration of cutting-edge natural language processing (NLP) techniques, advanced clustering algorithms, and sophisticated topic modeling approaches, the system endeavors to revolutionize the user experience by facilitating seamless navigation, personalized recommendations, and insightful product categorization. This documentation serves as a comprehensive guide, offering a detailed exploration of the project's overarching objectives, intricate methodology, illustrative code snippets, and illuminating results. By delving into the intricacies of each component, this document provides invaluable insights into the innovative strategies employed to enrich user interactions and optimize product discovery in the dynamic landscape of online retail.

Problem Statement:

In the ever-expanding landscape of e-commerce, Inefficient categorization and utilization of product descriptions hinder effective decision-making and users often face challenges in discovering relevant products and navigating through extensive product catalogues.

Objectives:

1. Cluster similar products based on their descriptions to identify ideal product groups.
2. Develop a recommender system that suggests similar products to users based on their interactions with a particular product.
3. Extract latent topics from product descriptions using topic modeling techniques to enhance categorization and search functionalities.

Dataset:

Dataset was obtained from Kaggle. It contains 500 actual stock keeping units from an outdoor apparel brand's product catalog.

Out[4]:

	id	description
0	1	Active classic boxers - There's a reason why o...
1	2	Active sport boxer briefs - Skinning up Glory ...
2	3	Active sport briefs - These superbreathable no...
3	4	Alpine guide pants - Skin in, climb ice, switc...
4	5	Alpine wind jkt - On high ridges, steep ice an...
...	...	...
495	496	Cap 2 bottoms - Cut loose from the maddening c...
496	497	Cap 2 crew - This crew takes the edge off fick...
497	498	All-time shell - No need to use that morning T...
498	499	All-wear cargo shorts - All-Wear Cargo Shorts ...
499	500	All-wear shorts - Time to simplify? Our All-We...

500 rows × 2 columns

Methodology:

Methodology involves comparing DBSCAN and K-Means for clustering similar products based on descriptions, employing Truncated SVD for extracting latent topics from TF-IDF matrices, generating word clouds to visualize frequent terms within clusters, and implementing a recommendation system to provide personalized product suggestions based on clustering results.

1. **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** DBSCAN groups together closely packed data points based on two main parameters: epsilon (ε) and min pts. It starts by selecting a random point in the dataset and finds all neighbouring points within a specified distance ε. If the number of neighbouring points exceeds a threshold min pts the point is considered a core point and forms the centre of a cluster. Additional points within ε of the core point are added to the same cluster. In the clustering phase, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is utilized to group similar products based on their descriptions. DBSCAN is chosen due to its ability to identify clusters in arbitrary shapes and its robustness to noise. This is particularly suitable for text data where clusters may not exhibit clear

geometric shapes and noise can arise from irrelevant or uncommon terms in product descriptions. Additionally, DBSCAN's parameter tuning capability ensures that meaningful clusters are obtained by adjusting parameters such as epsilon (eps) and minimum points (min samples) to optimize clustering results.

2. **K-Means Clustering**: K-Means is a centroid-based clustering algorithm that partitions data into k clusters by iteratively assigning each data point to the nearest centroid and updating centroids until convergence. It starts by randomly initializing k centroids in the feature space. Then, each data point is assigned to the nearest centroid, forming k clusters. After the initial assignment, centroids are recalculated as the mean of all data points in each cluster. This process continues until centroids no longer change significantly or a specified number of iterations is reached. K-Means is chosen for its simplicity, scalability, and effectiveness in handling large datasets. However, it may struggle with non-linear cluster shapes and is sensitive to the initial centroid positions, requiring multiple initializations to avoid local minima. Despite these limitations, K-Means remains a popular choice for clustering tasks in various domains due to its computational efficiency and ease of implementation.

3. **Visualization:** Word clouds are generated for each cluster to visualize the most frequent words within each cluster.

4. **Recommender System:** A function `find similar items` is defined to provide personalized recommendations based on the cluster to which a product belongs.

5. **SVD (Singular Value Decomposition):** Singular Value Decomposition (SVD) is a powerful matrix factorization technique used in various applications such as dimensionality reduction, data compression, and collaborative filtering in recommendation systems. In the topic modelling phase, Truncated SVD (Singular Value Decomposition) is applied for Latent Semantic Analysis (LSA) to extract latent topics from the TF-IDF matrix. Truncated SVD is chosen for its ability to reduce the dimensionality of the TF-IDF matrix while preserving important relationships between terms and documents, making subsequent analyses more efficient. Moreover, Truncated SVD enhances interpretability by extracting latent topics that represent sets of words frequently co-occurring across documents, facilitating a deeper understanding of underlying themes within the textual data.

Code Implementation and Explanation:

Data Pre-processing: Data pre-processing involves cleaning product descriptions by converting them to lowercase, removing HTML tags, special characters, and stopwords, followed by tokenization using spaCy to prepare the text data for clustering and topic modeling.

In [239]:
```
catalog["clean_documents"] =  catalog['description'].fillna('').apply(lambda x: x.lower())

catalog["clean_documents"] = catalog["clean_documents"].\
    apply(lambda x :
    x.replace("<li>",'')\
    .replace("</li>",'')\
    .replace("<br>",'')\
    .replace("</br>",'')\
    .replace("<b>",'')\
    .replace("</b>",'')\
    .replace("<ul>",'')\
    .replace("</ul>",''))

catalog["clean_documents"] = catalog["clean_documents"].str.replace(r"[^A-Za-z0-9]+", " ")


catalog.head()
```

Out[239]:

	id	description	clean_documents
0	1	Active classic boxers - There's a reason why o...	active classic boxers - there's a reason why o...
1	2	Active sport boxer briefs - Skinning up Glory ...	active sport boxer briefs - skinning up glory ...
2	3	Active sport briefs - These superbreathable no...	active sport briefs - these superbreathable no...
3	4	Alpine guide pants - Skin in, climb ice, switc...	alpine guide pants - skin in, climb ice, switc...
4	5	Alpine wind jkt - On high ridges, steep ice an...	alpine wind jkt - on high ridges, steep ice an...

In [240]:
```
import en_core_web_sm
nlp = en_core_web_sm.load()
```

In [241]:
```
catalog["tokenized_documents"] = catalog["clean_documents"]\
                        .fillna('')\
                        .apply(lambda x : [token.lemma_ for token in nlp(x) if token.text not in STOP_WORDS])
catalog.head()
```

Out[241]:

	id	description	clean_documents	tokenized_documents
0	1	Active classic boxers - There's a reason why o...	active classic boxers - there's a reason why o...	[active, classic, boxer, -, reason, boxer, cul...
1	2	Active sport boxer briefs - Skinning up Glory ...	active sport boxer briefs - skinning up glory ...	[active, sport, boxer, brief, -, skin, glory, ...
2	3	Active sport briefs - These superbreathable no...	active sport briefs - these superbreathable no...	[active, sport, brief, -, superbreathable, -, ...
3	4	Alpine guide pants - Skin in, climb ice, switc...	alpine guide pants - skin in, climb ice, switc...	[alpine, guide, pant, -, skin, ,, climb, ice, ...
4	5	Alpine wind jkt - On high ridges, steep ice an...	alpine wind jkt - on high ridges, steep ice an...	[alpine, wind, jkt, -, high, ridge, ,, steep, ...

In [242]:
```
catalog["nlp_ready"] = catalog["tokenized_documents"].apply(lambda x : (' '.join(x)).strip())
catalog.head()
```

Out[242]:

	id	description	clean_documents	tokenized_documents	nlp_ready
0	1	Active classic boxers - There's a reason why o...	active classic boxers - there's a reason why o...	[active, classic, boxer, -, reason, boxer, cul...	active classic boxer - reason boxer cult favor...
1	2	Active sport boxer briefs - Skinning up Glory ...	active sport boxer briefs - skinning up glory ...	[active, sport, boxer, brief, -, skin, glory, ...	active sport boxer brief - skin glory require ...
2	3	Active sport briefs - These superbreathable no...	active sport briefs - these superbreathable no...	[active, sport, brief, -, superbreathable, -, ...	active sport brief - superbreathable - fly bri...
3	4	Alpine guide pants - Skin in, climb ice, switc...	alpine guide pants - skin in, climb ice, switc...	[alpine, guide, pant, -, skin, ,, climb, ice, ...	alpine guide pant - skin , climb ice , switch ...
4	5	Alpine wind jkt - On high ridges, steep ice an...	alpine wind jkt - on high ridges, steep ice an...	[alpine, wind, jkt, -, high, ridge, ,, steep, ...	alpine wind jkt - high ridge , steep ice alpin...

Creating a TF_IDF matrix:

In [244]:
```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(stop_words='english')
X = vectorizer.fit_transform(catalog['nlp_ready'])

dense = X.toarray()
dense
```

Out[244]:
```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

This code segment performs text pre-processing, including the removal of HTML tags, punctuation, and stop words, tokenization, and lemmatization. It then generates a TF-IDF matrix to represent the cleaned documents, which can be further used for clustering and topic modeling.

Clustering and finding optimal eps and min_samples for DBSCAN:

```
In [245]: db = DBSCAN(eps=1, min_samples=180, metric="cosine")

          db.fit(X)

Out[245]: DBSCAN
          DBSCAN(eps=1, metric='cosine', min_samples=180)
```

```
In [246]: np.unique(db.labels_)

Out[246]: array([0], dtype=int64)
```

```
In [247]: def count_val(val,T):
              c = 0
              for i in T:
                  if i == val:
                      c += 1
              return c
```

```
In [248]: T_eps = np.arange(0.4,0.71,0.01)
          T_min_samples = list(range(3,10))

          for i in T_eps:
              for j in T_min_samples:
                  dbscan = DBSCAN(eps=i, min_samples=j, metric="cosine")
                  dbscan.fit(X)
                  clusters = np.unique(dbscan.labels_)
                  label = list(dbscan.labels_)
                  outliers_percent = count_val(-1,label)/len(label)*100

                  print(f"eps = {round(i,3)} - min_samples = {j} | {round(outliers_percent,2)}% outliers ({count_val(-1,label)}) ")
                  print(f"{len(clusters)} clusters created")
                  print("-"*50)
```

```
18 clusters created
--------------------------------------------------
eps = 0.7 - min_samples = 4 | 11.0% outliers (55)
18 clusters created
--------------------------------------------------
eps = 0.7 - min_samples = 5 | 16.2% outliers (81)
14 clusters created
--------------------------------------------------
eps = 0.7 - min_samples = 6 | 19.8% outliers (99)
14 clusters created
--------------------------------------------------
eps = 0.7 - min_samples = 7 | 21.4% outliers (107)
14 clusters created
--------------------------------------------------
eps = 0.7 - min_samples = 8 | 30.4% outliers (152)
14 clusters created
--------------------------------------------------
eps = 0.7 - min_samples = 9 | 38.6% outliers (193)
9 clusters created
--------------------------------------------------
```

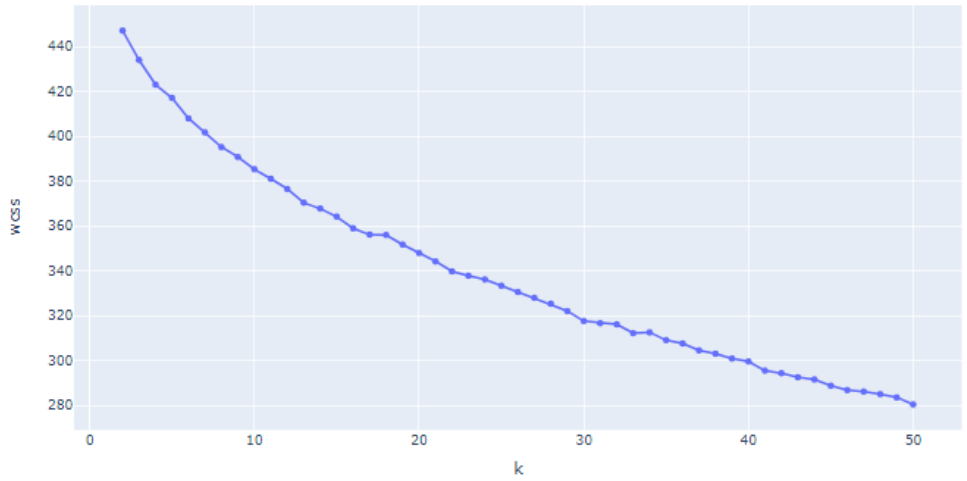
The best hyperparameters with less outliers is eps=0.7 and min_samples=3

```
In [249]: db = DBSCAN(eps=0.7, min_samples=3, metric="cosine")
          db.fit(X)
          np.unique(db.labels_,return_counts=True)

Out[249]: (array([-1,  0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
                  16], dtype=int64),
           array([ 41,  64, 187,  28,  22,  10,  22,  56,   8,   4,   7,  24,   4,
                    5,   7,   3,   4,   4], dtype=int64))
```

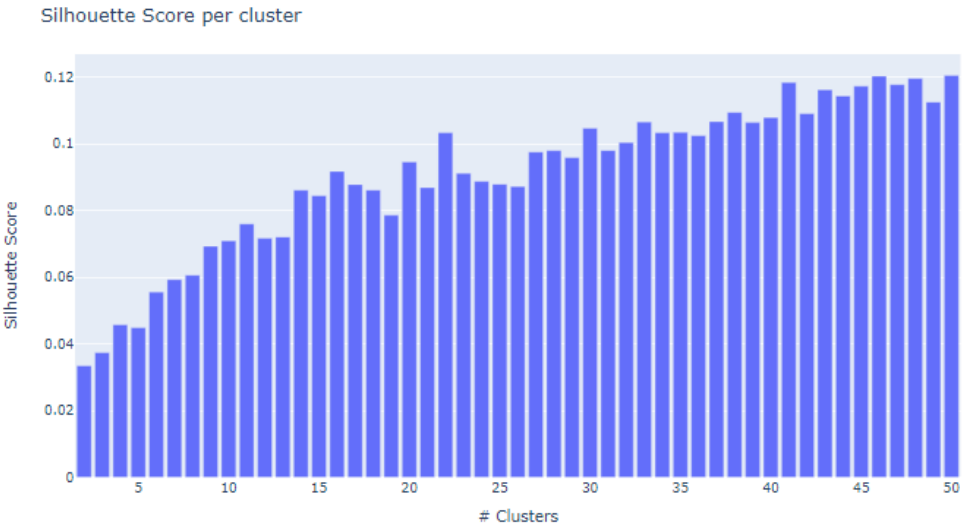
This code segment utilizes DBSCAN algorithm on the TF-IDF matrix to cluster similar documents. It iteratively adjusts the parameters (epsilon and min_samples) to find the optimal values by evaluating the number of unique labels which are minPts = 3 and eps = 0.7 which detect less outliers with 8%. After clustering, it visualizes each cluster using word clouds to show the most frequent terms within each cluster.

K-Means:

In [254]:
```
fig = px.line(kmeans, x="k", y="wcss",markers=True)
fig.update_traces(textposition="bottom right")
fig.show(renderer="iframe") # if using workspace
```



In [255]:
```
fig = px.bar(data_frame=kmeans, x=k, y=sil)

fig.update_layout(
    yaxis_title="Silhouette Score",
    xaxis_title="# Clusters",
    title="Silhouette Score per cluster"
)

fig.show(renderer="iframe")
```



In [256]:
```
kmeans = KMeans(n_clusters= 41, random_state = 42)
kmeans.fit(X)
```

```
D:\conda\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warnin
g
```

Out[256]:

KMeans

```
KMeans(n_clusters=41, random_state=42)
```

This code segment involves finding the optimal k using elbow plot and silhouette score and fitting K-Means model. Optimal k is 41.

Recommender System:

In [262]:
```
def find_similar_items(product_id):
    cluster_id = int(catalog.loc[catalog["id"] == product_id,'kmeans_cluster'])
    similar_items = catalog.loc[catalog["kmeans_cluster"] == cluster_id,:].loc[catalog["id"] != product_id,:]["id"]
    sample = similar_items.sample(min(5,len(similar_items))) # security if 5 is greater than the max
    return sample
```

In [263]:
```
def reco_system():
    product_id = int(input("What product would you like to buy ? "))

    print()
    try:
        item_desc = catalog.loc[catalog['id']==product_id, 'clean_documents'].values[0]
    except:
        print('Product not found in database. Please enter a valid product id.')
    else:
        print(f"Product found in database, description below :")
        print(item_desc)
        print()

        print(f"You might also be interested by the following products : ")
        print()

        for i in find_similar_items(product_id):
            print(f"Item #", i)
            print(catalog.loc[catalog['id']==i, 'clean_documents'].values[0])
            print('-'*50)
```

In [*]:
```
reco_system()
```

```
What product would you like to buy ? 5
```

This function, `find_similar_items`, retrieves 5 products belonging to same cluster. It retrieves the cluster ID of the input product using KMeans clustering results. Another function reco_system() is the recommendation system function. It prompts the user to input a product ID they would like to purchase. It then retrieves the product description from the dataframe based on the input ID and displays it. After that, it calls the find_similar_items() function to find similar products based on clustering results and displays their descriptions as recommendations to the user.

Topic Modeling and Visualization:

In [268]:
```
svd_model = TruncatedSVD(n_components=15, algorithm='randomized',random_state=122)
lsa = svd_model.fit_transform(X)

topic_encoded_df = pd.DataFrame(lsa, columns = [f"topic_{i}" for i in range(len(lsa[0]))])
topic_encoded_df["documents"] = catalog['clean_documents']

topic_encoded_df.head(3)
```

Out[268]:

	topic_0	topic_1	topic_2	topic_3	topic_4	topic_5	topic_6	topic_7	topic_8	topic_9	topic_10	topic_11	topic_12	topic_13	topic_14
0	0.254368	-0.048457	0.186109	0.079333	-0.124193	-0.015025	-0.045512	-0.032078	-0.143735	0.047001	0.000634	0.036199	0.049939	0.027308	0.038711
1	0.263908	-0.074984	0.101777	0.097923	-0.030702	0.087509	-0.055816	-0.031384	-0.138312	-0.048295	-0.040141	0.177507	-0.009287	0.064457	-0.028708
2	0.238223	-0.093284	0.107353	0.020145	-0.029418	0.092321	0.007608	0.046380	-0.081098	-0.011711	-0.042994	0.206832	-0.076280	0.079859	-0.023887

This code segment uses Truncated SVD to reduce the dimensionality of a TF-IDF matrix derived from product descriptions. It then generates topic-encoded data and visualizes topic distributions. Finally, it extracts word-topic associations and creates word clouds to illustrate the most prevalent terms within each topic, aiding in the interpretation of latent themes in the descriptions.

Evaluation:

DBSCAN and K-Means were both evaluated on Silhouette score.

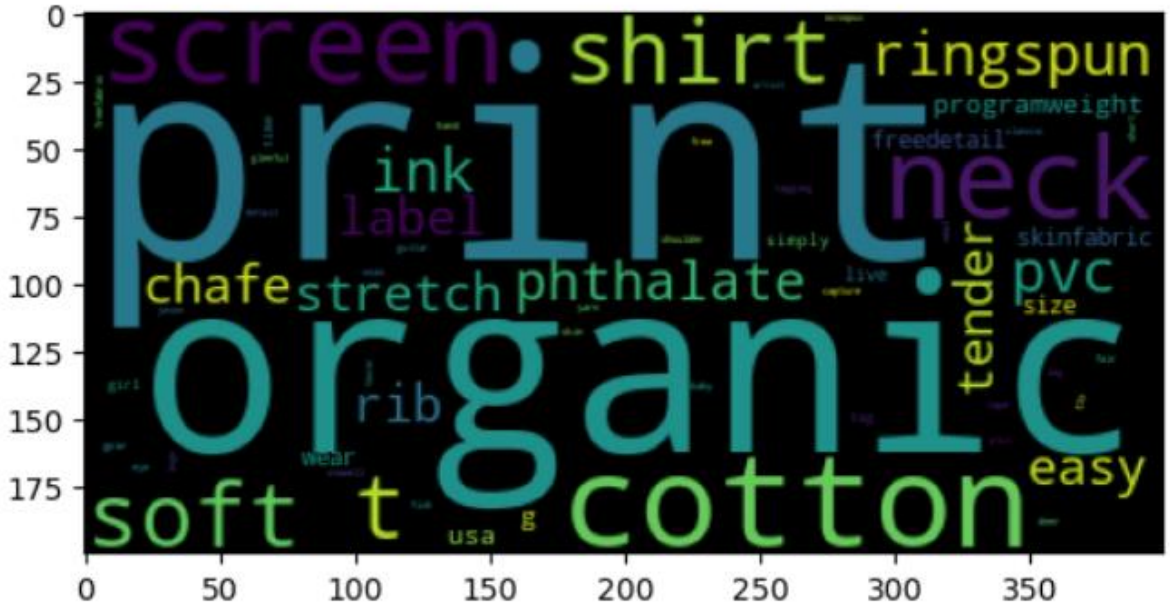
```
In [258]: silhouette_db = silhouette_score(X, catalog['dbscan_cluster'])
          print("Silhouette score for DBSCAN:", silhouette_db)

          # Silhouette score for KMeans
          silhouette_kmeans = silhouette_score(X, catalog['kmeans_cluster'])
          print("Silhouette score for KMeans:", silhouette_kmeans)

          Silhouette score for DBSCAN: 0.05073804549943108
          Silhouette score for KMeans: 0.11836405514804428
```

Results:

DBSCAN vs. K-Means: The silhouette scores indicate the performance of DBSCAN and K-Means, with K-Means chosen for its higher silhouette score.

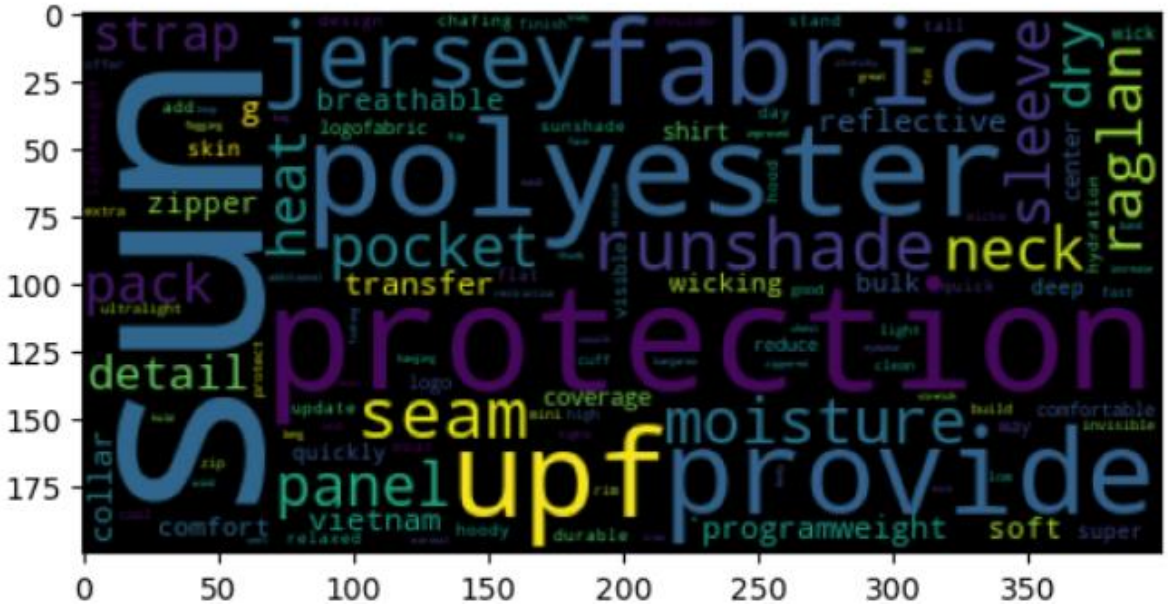
K-Means effectively groups similar products. Here are some of the wordclouds for the clusters:

```
Cluster No. 1
```



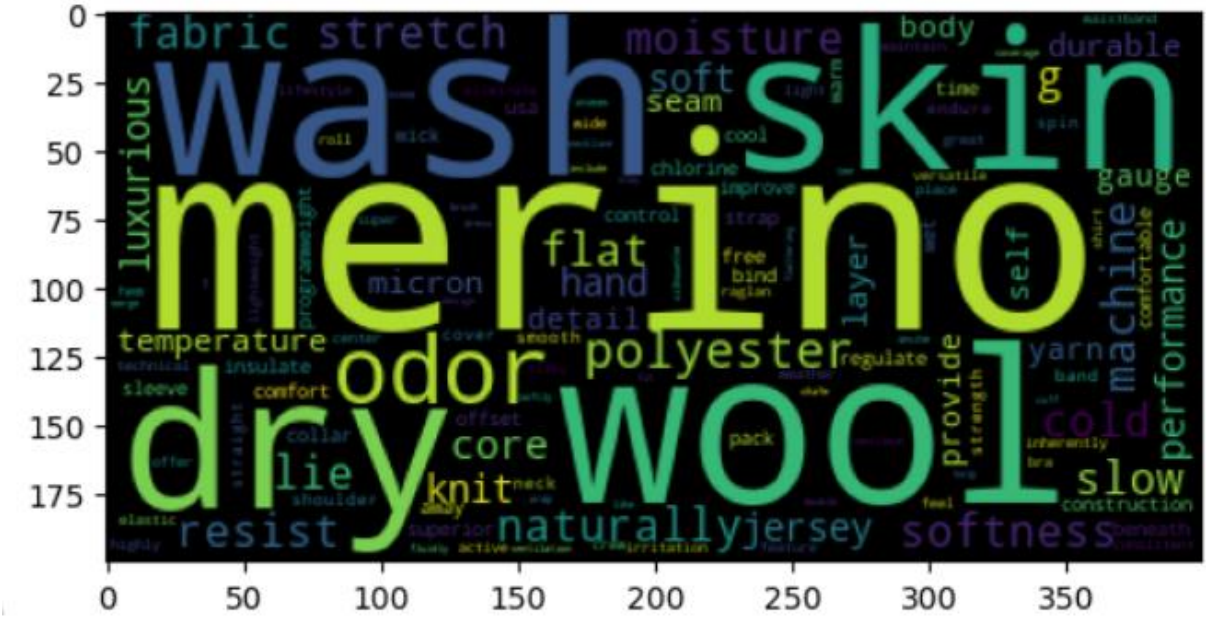
This wordcloud for cluster 1 represents a group of products associated with soft, organic materials such as cotton or bamboo, often used for printed shirts, with potential emphasis on eco-friendly attributes.

Cluster No. 2



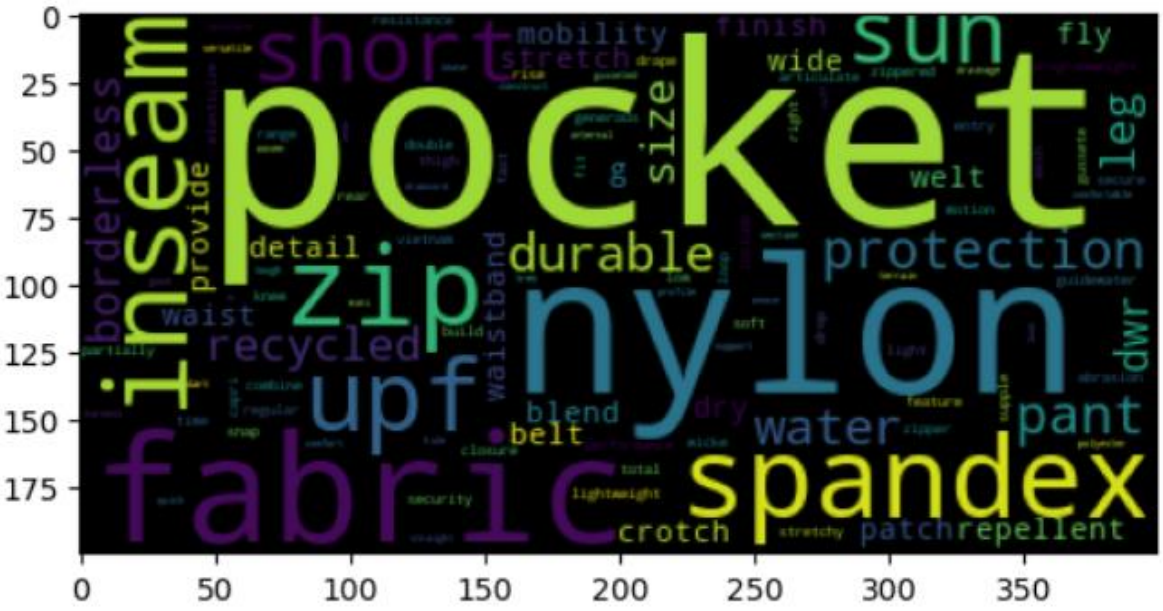
This wordcloud for cluster 2 represents products made from polyester or jersey fabric, designed for sun protection, suggesting a focus on performance wear or outdoor apparel.

Cluster No. 3



This wordcloud for cluster 3 represents items crafted from merino wool, known for its odor-resistant and skin-friendly properties, suggesting a cluster of high-quality, easy-care garments..

Cluster No. 4



This wordcloud for cluster 4 represents products featuring functional details such as pockets and zip closures, crafted from stretchy materials like spandex and nylon, likely representing a range of activewear or performance-oriented garments.

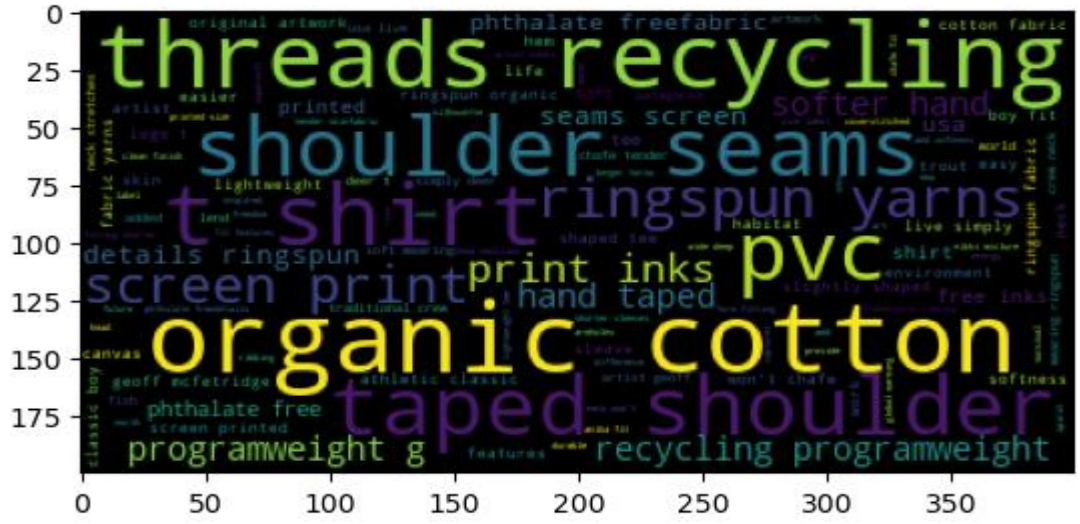
Recommendation System: The recommender system accurately provides personalized recommendations by leveraging the clustering results.

```
What product would you like to buy ? 5

Product found in database, description below :
alpine wind jkt - on high ridges, steep ice and anything alpine, this jacket serves as a true "best of all worlds" staple. it e
xcels as a stand-alone shell for blustery rock climbs, cool-weather trail runs and high-output ski tours. and then, when condit
ions have you ice and alpine climbing, it functions as a lightly insulated windshirt on the approach, as well as a frictionless
midlayer when it's time to bundle up and tie-in. the polyester ripstop shell with a deluge dwr (durable water repellent) treate
nt sheds snow and blocks wind, while the smooth, lightly brushed hanging mesh liner wicks moisture, dries fast, and doesn't bi
nd to your baselayers. superlight stretch-woven underarm panels enhance breathability and allow for unimpaired arm motion, and
the two hand pockets close with zippers. a drawcord hem, elastic cuffs, a heat-transfer reflective logo and a regular-coil, cen
ter-front zipper with dwr finish round out the features. updated this season for an improved fit. recyclable through the common
threads recycling program.details: "lightweight, breathable polyester ripstop fabric with deluge dwr (durable water repellent)
finish; slightly brushed polyester-mesh liner wicks moisture and dries fast" "stretch-woven underarm panels provide breathabili
ty, stretch for unimpeded range of motion" dwr finish on center-front zipper elastic cuffs "pockets: exterior chest, zippered h
andwarmers" drawcord hem reflective heat-transfer logofabric: shell: 1.3-oz 20-denier 100% polyester ripstop. panels: 4.6-oz 75
-denier 90% all-recycled polyester/10% spandex. shell and panels have deluge dwr (durable water repellent) finish. lining: 100%
polyester brushed tricot mesh. recyclable through the common threads recycling programweight: (331 g 11.5 oz)made in china.

You might also be interested by the following products :

Item # 97
nine trails shorts - for those who view trails as their sanctuary, running means more than freedom of movement and lightness of
being. our versatile, pared-down men's nine trails shorts are made from a lightweight 75-denier 91% all-recycled polyester/9% s
pandex blend that has 4-way stretch, is tough yet highly breathable, and is treated with a deluge dwr (durable water repellent)
finish. the styling has ample coverage, the stitched-through pocket bags stay put, and the pockets (two front, one back) close
with zippers for secure storage. an open-knit crotch gusset vents excess heat and eliminates chafing; the lightweight, breathab
le, moisture-wicking built-in liner extends into the waistband; and an interior drawcord fine-tunes the fit. inseam (size m) is
8". reflective heat-transfer logo and reflective graphic keep you visible.details: "supple, soft, stretch-woven breathable fabr
ic resists snagging and is recyclable; deluge dwr (durable water repellent) finish for weather protection" "lightweight, breath
able, moisture-wicking, built-in liner extends into waistband for a comfortable fit" interior drawcord two front and one back p
ocket with zippered closures and stitched through pocket bags for secure storage open-knit crotch gusset stays dry and puts an
end to chafing reflective heat-transfer logo and graphic "8"" inseam"fabric: body: 3.5-oz 75-denier 91% all-recycled polyester/
9% spandex with 4-way stretch and deluge dwr finish. lining: 3.8-oz microdenier polyester crepe with moisturewicking  performan
ce. recyclable through the common threads recycling programweight: (181 g 6.3 oz)made in vietnam.
----------
Item # 485
houdini full-zip jkt - now you see it, now you don't. just like the sun on blustery days, our ultralight, ultrapackable houdini
can shed fickle weather and then disappear when the skies clear. it's also breathable enough to keep on as you crank through th
e next squall. the textured triple-ripstop nylon fabric proves surprisingly tough, while the deluge dwr (durable water repellen
t) finish wards off flurries six pitches from the deck. the lower-volume hood snugs around the face and head three-dimensionall
y with a single-pull system. the jacket stuffs into its own chest pocket, which has a secure zipper closure and a burly-strong
carabiner clip-in loop. with a drawcord hem.details: "highly breathable, incredibly light, textured soft-shell fabric has a str
ong, triple ripstop pattern and a subtle, slightly transparent appearance; deluge dwr (durable water repellent) finish" dwr-coa
ted center-front zipper "zippered exterior chest pocket doubles as a stuff sack, with a reinforced carabiner clip loop" hood op
ening and field-of-vision adjustment in one pull may be worn over baselayers and light midlayersfabric: 1-oz 15-denier 100% nyl
on ripstop with deluge dwr finish. recyclable through the common threads recycling programweight: (124 g 4.3 oz)made in china.
----------
Item # 308
alpine wind jkt - on high ridges, steep ice and anything alpine, this jacket serves as a true "best of all worlds" staple. it e
xcels as a stand-alone shell for blustery rock climbs, cool-weather trail runs and high-output ski tours. and then, when condit
ions have you ice and alpine climbing, it functions as a lightly insulated windshirt on the approach, as well as a frictionless
midlayer when it's time to bundle up and tie-in. the polyester ripstop shell with a deluge dwr (durable water repellent) treate
nt sheds snow and blocks wind, while the smooth, lightly brushed hanging mesh liner wicks moisture, dries fast, and doesn't bi
nd to your baselayers. superlight stretch-woven underarm panels enhance breathability and allow for unimpaired arm motion, and
the two hand pockets close with zippers. a drawcord hem, elastic cuffs, a heat-transfer reflective logo and a regular-coil, cen
ter-front zipper with dwr finish round out the features. updated this season for an improved fit. recyclable through the common
threads recycling program.details: "lightweight, breathable polyester ripstop fabric with a deluge dwr (durable water repellen
t) finish; slightly brushed polyester-mesh liner wicks moisture and dries fast" "stretch-woven underarm panels provide breathab
ility, stretch for unimpeded range of motion" center-front zipper has dwr finish elastic cuffs zippered handwarmer pockets draw
cord hem reflective heat-transfer logofabric: shell: 1.3-oz 20-denier 100% polyester ripstop. panels: 4.6-oz 75-denier 90% all-
recycled polyester/10% spandex. shell and panels have deluge dwr (durable water repellent) finish. lining: 100% polyester brush
ed tricot mesh. recyclable through the common threads recycling programweight: (274 g 9.5 oz)made in china.
----------
```

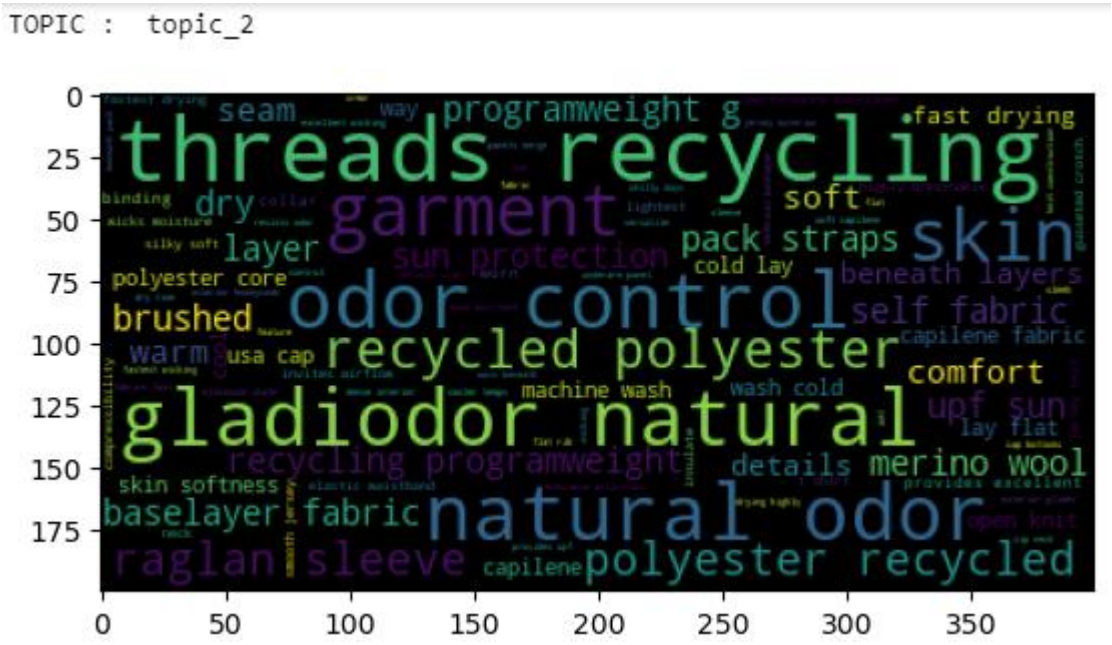
Topic modelling: Topic modelling effectively extracts meaningful topics from product descriptions, contributing to a deeper understanding of product themes and attributes. These are some of the topics formed:

```
 topic_0
['pocket', 'recycle', 'organic', 'cotton', 'polyester']

 topic_1
['organic', 'shirt', 'cotton', 'ringspun', 'print']

 topic_2
['merino', 'odor', 'pocket', 'wool', 'control']

 topic_3
['organic', 'inseam', 'cotton', 'button', '32']

 topic_4
['merino', 'wool', 'wash', 'sun', 'gladiodor']

 topic_5
['82', 'nylon', 'coverage', '18', 'organic']

 topic_6
['cotton', 'organic', '82', 'tencel', 'upf']

 topic_7
['waterproof', 'strap', 'barrier', 'h2no', 'deni']

 topic_8
['sun', 'protection', 'upf', '30', 'collar']

 topic_9
['fleece', 'waterproof', 'strap', 'wind', 'barrier']

 topic_10
['photo', 'poster', 'outside', 'fleece', 'tribute']
```

```
TOPIC :  topic_1
```



This wordcloud for topic 1 highlights products made from organic cotton, possibly emphasizing printed designs and premium quality.



This wordcloud for topic 2 indicates the incorporation of recycled materials and odor-control features, suggesting a cluster of products prioritizing sustainability and enhanced wearer comfort.

This wordcloud for topic 3 highlights products made from organic cotton with specific inseam measurements, catering to consumers seeking both comfort and precise fit in their clothing.

Inferences:

The evaluation of DBSCAN and K-Means clustering methods based on silhouette scores reveals K-Means' superior performance in grouping similar products. This selection is further validated by the effective clustering achieved by K-Means, as demonstrated in wordclouds showcasing distinct product characteristics within each cluster. For instance, cluster 1 represents products crafted from soft, organic materials like cotton or bamboo, emphasizing eco-friendly attributes and commonly used in printed shirts. Cluster 2 depicts products made from polyester or jersey fabric designed for sun protection, catering to consumers seeking performance wear or outdoor apparel. Additionally, cluster 3 showcases items crafted from merino wool, known for its odor-resistant and skin-friendly properties, suggesting a cluster of high-quality, easy-care garments. Cluster 4 illustrates products featuring functional details such as pockets and zip closures, crafted from stretchy materials like spandex and nylon, likely appealing to those seeking activewear or performance-oriented garments. These findings underscore the effectiveness of K-Means clustering in accurately grouping products, thereby enhancing the performance of the recommendation system.

The utilization of topic modelling further enriches the understanding of product descriptions, revealing key themes and attributes across clusters. For instance, topic 1 highlights products made from organic cotton, potentially emphasizing printed designs and premium quality. Meanwhile, topic 2 indicates the incorporation of recycled materials and odor-control features, suggesting a cluster of products prioritizing sustainability and enhanced wearer comfort. Finally, topic 3 focuses on products made from organic cotton with specific inseam measurements, catering to consumers seeking both comfort and precise fit in their clothing choices. These insights offer valuable guidance for product development, marketing strategies, and addressing consumer preferences within specific thematic categories.

Limitations:

Limitations of the e-commerce Product Analysis and Recommendation System include challenges in ensuring data representativeness, scalability concerns, sensitivity to parameter tuning, interpretability issues with topic modelling, lack of contextual information, and potential limitations in generalizing the approach to other product domains. The system's efficacy heavily relies on the representativeness and quality of the dataset, with biased or limited data potentially leading to skewed clustering results and inaccurate recommendations. Which Item id represents which product is not known prior to user making it hard to get recommendations. While the implemented algorithms demonstrate effectiveness on the current dataset size, scalability to larger datasets remains a concern, posing challenges in real-time application and system scalability. Moreover, the sensitivity of clustering algorithms to parameter tuning requires careful experimentation and validation to identify optimal configurations. Interpretability of topics extracted through topic modelling may vary, impacting the system's ability to derive actionable insights for product categorization. Additionally, the absence of contextual information such as user reviews or historical sales data limits the system's ability to provide comprehensive recommendations. Generalizing the approach to other product domains requires careful adaptation and validation due to variations in product attributes and consumer behaviour across different industries.

Conclusion:

In conclusion, the e-commerce Product Analysis and Recommendation System offers a promising solution for enhancing user engagement and streamlining product discovery in online retail. Leveraging advanced natural language processing, clustering algorithms, and topic modelling, the system provides valuable insights into product categorization, personalized recommendations, and thematic analysis. Despite limitations in data representativeness, scalability, and interpretability, the system demonstrates effectiveness in grouping similar products, generating personalized recommendations, and extracting meaningful topics from descriptions. Addressing these limitations and refining algorithms will be pivotal for maximizing utility and adaptability across diverse e-commerce platforms and product domains.