Section 13: Topic Modeling (Unsupervised)

Key Insights from Topic Modeling Setup

1. **Data Input Size**:
   - You have **5,400 text records** (e.g., responses or comments) in your dataset.
   - These were likely preprocessed (tokenized, cleaned, etc.) before modeling.
2. **TF-IDF Representation**:
   - The **TF-IDF matrix shape is (5400, 342)**, indicating:
     - **5,400 rows = documents** (responses)
     - **342 columns = unique terms/features** retained after filtering (stopword removal, frequency thresholding, etc.).
   - This representation helps capture **term importance** across documents in a sparse format.
3. **SBERT Embeddings**:
   - The **SBERT (Sentence-BERT) embeddings have shape (5400, 384)**:
     - Each of the 5,400 responses is converted into a **384-dimensional dense vector** using SBERT.
   - SBERT captures **semantic meaning** of entire sentences or phrases, unlike TF-IDF which focuses on word-level frequency.
4. **Model Readiness**:
   - The data is now in two powerful formats: **TF-IDF for interpretable keywords**, and **SBERT for contextual clustering**.
   - You're ready to apply **topic modeling algorithms** such as BERTopic or LDA with semantic clustering and interpretability.

What This Enables:

- **More nuanced topic modeling**: Using SBERT allows for better grouping of similar meanings even if the words differ.
- **Deeper insights extraction**: Topics found using SBERT will reflect **themes and sentiments**, not just co-occurrence of keywords.
- **Effective stakeholder reporting**: Combined with TF-IDF, you can extract **top keywords per topic** for easy explanation.

1. **Topic Summary Table**
   - Columns: Topic_ID, Top_Keywords, Interpretative Label.

Column Name	Description
`id`	**Unique Respondent ID** — Each row corresponds to a specific respondent.
`Question_ID`	**Question Identifier** — Indicates which open-ended question the text is from (e.g., `Q19`, `Q25`).
`Cleaned_Text`	**Preprocessed Text** — The cleaned and lemmatized version of the original open-ended response.
`Topic`	**Topic Number** — The numeric ID assigned by BERTopic to this text snippet based on semantic similarity. Example: `0`, `1`, `-1` (where `-1` typically indicates an outlier or unclustered text).
`Topic_Probability`	**Topic Membership Probability** — A float between 0 and 1 representing how confidently the model assigned this response to its topic. Higher values mean stronger topic association.

2. **Topic Assignment File**
    - Columns: Respondent_ID, Question_ID, Topic_ID, Topic_Probability.

https://docs.google.com/spreadsheets/d/1ejnuioXFQa2MBZ4P95fUspL6CNPP3AqmDqG_eoeqtSc/edit?gid=889103237#gid=889103237

3. **Prevalence Report**
    - Tables & charts showing topic distribution across Age, Gender, Segment.

https://docs.google.com/spreadsheets/d/1ejnuioXFQa2MBZ4P95fUspL6CNPP3AqmDqG_eoeqtSc/edit?gid=1853506090#gid=1853506090

Key Statistical Test Results & Insights

Variable	p-value	Interpretation
Gender vs. Topic Distribution	0.9849	❌ **No significant relationship** between gender and the distribution of topics. Both male and female respondents discuss similar themes.
Age vs. Topic Distribution	0.8248	❌ **No significant association** between age groups and topics. Younger and older respondents show **no major difference** in the themes they mention.
NCCS vs. Topic Distribution	0.8818	❌ **No significant difference** in topic preferences across socioeconomic segments. Consumers across NCCS levels talk about similar themes.

🧠 Overall Insight:

Demographic factors like **gender**, **age**, and **NCCS** do **not significantly influence** the kinds of topics people mention in open-ended responses. This suggests:

- Themes such as product usage, packaging, trust, effectiveness, etc., are **consistently mentioned across audience segments**.
- Your product communication and positioning messages might **resonate uniformly**, regardless of demographic variation.

Description:

Objectives

- Extract latent topics from responses related to brand choice, packaging feedback, memorable attributes, preference reasons, and format motivations.
- Use BERTopic (or fallback to LDA) to identify 8–12 coherent topics.
- Analyze topic prevalence across Age, Gender, and key consumer segments.

Analysis Tasks

Task	Details	Method
1. Prepare Input	- Load cleaned and lemmatized text from Section 12 - Optionally aggregate by respondent - Choose input: TF-IDF for LDA or sentence embeddings for BERTopic	`python import pandas as pd from sklearn.feature_extraction.text import TfidfVectorizer df = pd.read_csv("openends_cleaned.csv") texts = df['Cleaned_Text'].tolist() tfidf = TfidfVectorizer(max_features=3000) tfidf_matrix = tfidf.fit_transform(texts)`

2. Topic Model Training	- Use BERTopic with Sentence-BERT embeddings + UMAP + HDBSCAN - Extract top keywords per topic - Fallback: LDA using TF-IDF	`python from bertopic import BERTopic from sentence_transformers import SentenceTransformer model = SentenceTransformer('all-MiniLM-L6-v2') embeddings = model.encode(texts, show_progress_bar=True) topic_model = BERTopic(language="english") topics, probs = topic_model.fit_transform(texts, embeddings)`
3. Topic Tuning	- Aim for 8–12 topics - Adjust HDBSCAN parameters (min_cluster_size) - Check coherence and interpretability - Merge/split if needed	`python topic_model.get_topic_info() topic_model.visualize_barchart(top_n_topics=12)`
4. Topic Assignment	- Assign dominant topic to each text - Create topic-response mapping table	`python df['Topic_ID'] = topics df['Topic_Probability'] = probs df[['Respondent_ID', 'Question_ID', 'Topic_ID', 'Topic_Probability']].to_csv("topic_assignment.csv", index=False)`
5. Prevalence Profiling	- Group by Age, Gender, and Segments - Compute frequency or proportion of each topic - Apply chi-square test	`python import scipy.stats as stats crosstab = pd.crosstab(df['Topic_ID'], df['Gender']) chi2, p, _, _ = stats.chi2_contingency(crosstab) print("Chi-square p-value:", p)`
6. Visualization	- Bar charts: Topic prevalence by group - Optional: Word clouds per topic	`python import seaborn as sns import matplotlib.pyplot as plt topic_counts = df['Topic_ID'].value_counts().sort_index() sns.barplot(x=topic_counts.index,`

```
y=topic_counts.values) plt.xlabel("Topic
ID") plt.ylabel("Frequency")
plt.title("Topic Distribution")
plt.show()
```

Deliverables

Topic Modeling Notebook

- Full implementation of data prep, model fitting, tuning, and assignment
- Annotated with justifications and parameter notes

Topic Summary Table

- Columns: `Topic_ID`, `Top_Keywords`, `Interpretative_Label`
- Example:

Topic_ID	Top_Keywords	Interpretative_Label
0	convenient, use, fast	Ease of Use
1	smell, fragrance, nice	Sensory Appeal

Topic Assignment File

- File: `topic_assignment.csv`
- Columns: `Respondent_ID`, `Question_ID`, `Topic_ID`, `Topic_Probability`

Prevalence Report

- Tables of topic frequency across:
    - Age groups (18–25, 26–35, 36+)
    - Gender (Male/Female)
    - Segments (None, Mild, Moderate, Severe)

- Statistical tests (Chi-square, ANOVA) with p-values

Slide Deck Snippets

- Bar charts of top topics
- Keyword visualizations (e.g., barcharts or word clouds)
- Key cross-group differences annotated