Section 17: Customer Segmentation

1. **Feature Matrix**
   – Clean, scaled dataset used for clustering

https://docs.google.com/spreadsheets/d/1ejnuioXFQa2MBZ4P95fUspL6CNPP3AqmDqG_eoeqtSc/edit?gid=197570711#gid=197570711

Scope: Segment the consumer base into distinct personas using behavioral, demographic, and attitudinal data to drive marketing and product strategies.

Objectives

- Segment respondents into 4–6 unique personas using quantitative clustering methods.
- Characterize each persona by demographics, behavioral features, and attitudinal indicators.
- Validate segmentation effectiveness using silhouette scores and GMM-based model fit indices.

Analysis Tasks

Task	Details	Method / Tools

1. Feature Matrix Construction	- Extract numeric variables from Q1–Q30: — Dandruff severity, hair fall, usage frequency — Purchase behavior, engagement flags (eye-tracking) — Sentiment/excitement scores from open-ended responses	`python import pandas as pd import numpy as np df = pd.read_csv("survey_data_cleaned.csv") features = df[["Dandruff_Score", "HairFall_Freq", "Purchase_Cadence", "Sentiment_Score", "EyeTracking_Flag"]] X = features.copy()`
2. Preprocessing & Scaling	- Impute missing values using mean or flags - Standardize features before clustering	`python from sklearn.impute import SimpleImputer from sklearn.preprocessing import StandardScaler imputer = SimpleImputer(strategy="mean") X_imputed = imputer.fit_transform(X) scaler = StandardScaler() X_scaled = scaler.fit_transform(X_imputed)`
3. Clustering Models	- Apply KMeans with k = 4 to 6 - Fit Gaussian Mixture Models for soft segmentation	`python from sklearn.cluster import KMeans from sklearn.mixture import GaussianMixture kmeans = KMeans(n_clusters=5, random_state=42) kmeans_labels = kmeans.fit_predict(X_scaled) gmm = GaussianMixture(n_components=5, random_state=42) gmm_labels = gmm.fit_predict(X_scaled)`

4. Cluster Validity & Selection	- Evaluate silhouette scores and GMM BIC/AIC - Visualize results with elbow and silhouette plots	`python from sklearn.metrics import silhouette_score import matplotlib.pyplot as plt silhouette_scores = [] bic_scores = [] aic_scores = [] for k in range(2, 8): km = KMeans(n_clusters=k, random_state=42) labels = km.fit_predict(X_scaled) silhouette_scores.append(silhouette_score(X_scaled, labels)) gmm_k = GaussianMixture(n_components=k, random_state=42).fit(X_scaled) bic_scores.append(gmm_k.bic(X_scaled)) aic_scores.append(gmm_k.aic(X_scaled)) plt.plot(range(2, 8), silhouette_scores, label="Silhouette") plt.plot(range(2, 8), bic_scores, label="BIC") plt.plot(range(2, 8), aic_scores, label="AIC") plt.legend(); plt.title("Cluster Evaluation Metrics"); plt.xlabel("Number of Clusters"); plt.show()`
5. Persona Profiling	- Aggregate means for each cluster - Breakdown by demographics (age, gender, NCCS) - Label segments with persona narratives	`python df["Cluster"] = kmeans_labels profile = df.groupby("Cluster").agg({ "Dandruff_Score": "mean", "HairFall_Freq": "mean", "Purchase_Cadence": "mean", "Sentiment_Score": "mean", "Gender": lambda x: x.value_counts().index[0], "Age_Group": lambda x: x.value_counts().index[0], "NCCS": lambda x: x.value_counts().index[0] }).reset_index() print(profile)`
6. Validation & Refinement	- Cross-tab against known segments (e.g., Test vs Control) - Bootstrap sampling to assess cluster stability	`python from sklearn.utils import resample stability_scores = [] for i in range(10): X_sample = resample(X_scaled, random_state=i) km = KMeans(n_clusters=5, random_state=42) labels =`

```
km.fit_predict(X_sample) score =
silhouette_score(X_sample, labels)
stability_scores.append(score)
print("Mean Silhouette Stability:",
np.mean(stability_scores))
```

Deliverables

Feature Matrix

- Final preprocessed dataset used for clustering:
  ```
  feature_matrix_scaled.csv
  python
  pd.DataFrame(X_scaled).to_csv("feature_matrix_scaled.csv",
  index=False)
  ```

Clustering Results

- Cluster assignments:
  ```
  python df[["Respondent_ID",
  "Cluster"]].to_csv("cluster_assignments.csv", index=False)
  ```
- Model selection plots (Silhouette, BIC/AIC)
- KMeans parameters: `n_clusters=5`, `init='k-means++'`

Persona Profiles

- 4–6 consumer personas with titles such as:
  - “Loyal High-Frequency Users”
  - “Skeptical Low-Engagement Trialists”
- Profile document includes:
  - Mean scores
  - Demographic dominance
  - Key behaviors and sentiment indicators

Visualization Deck

- Plots:
  - Silhouette and BIC/AIC curves
  - Heatmaps or radar charts of cluster averages
- One slide per persona:
  - Name
  - Demographic composition
  - Key usage/sentiment behavior

Analysis Notebook

- Fully commented Python code:
  - Feature engineering
  - Clustering
  - Evaluation
  - Persona generation
- Reproducible and modular for future updates