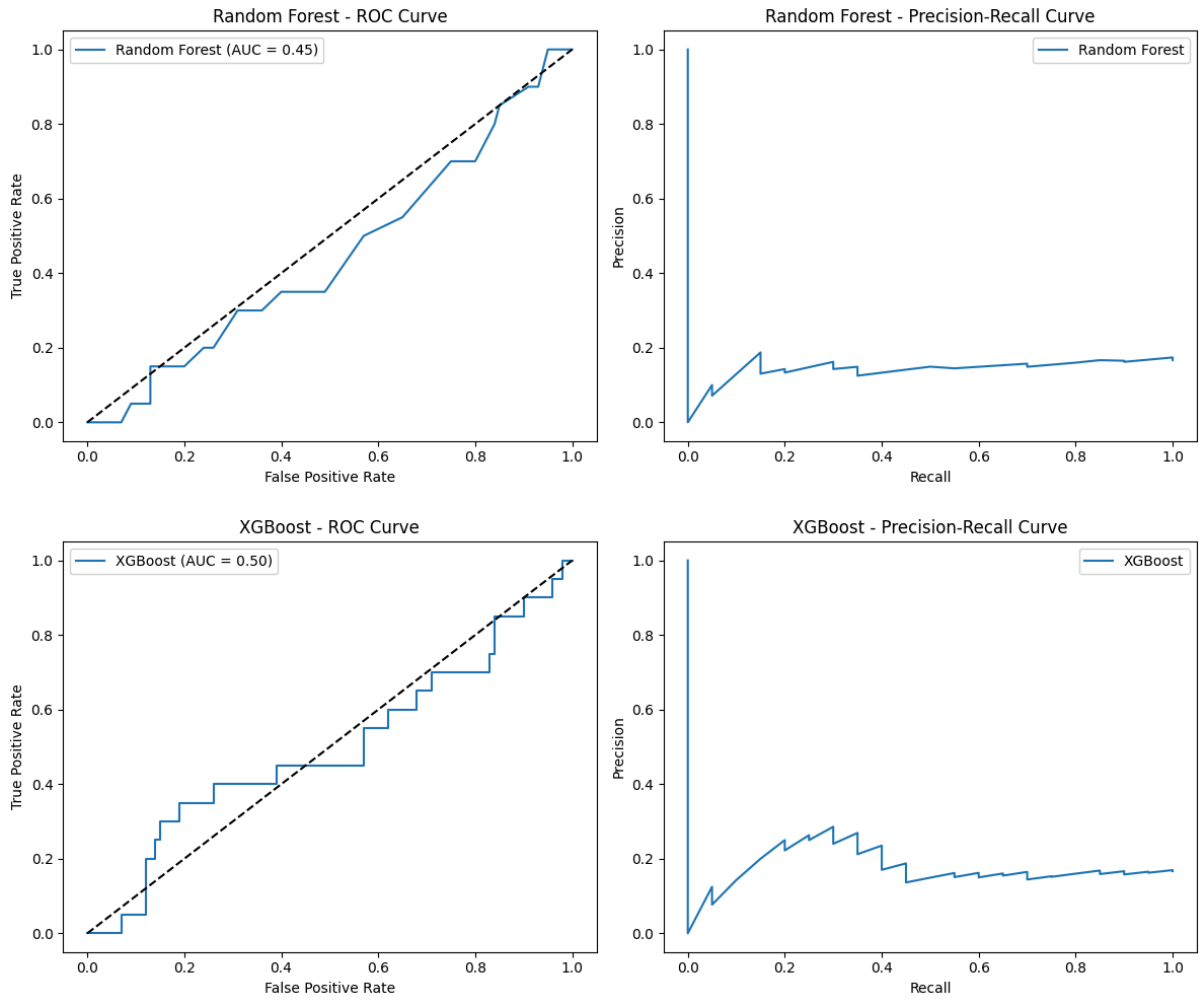
Task Brief – Section 18: Predictive Modeling for Purchase Intent

---

1. **Processed Dataset**
   – Train/test sets with features and labels.

2. **Model Artifacts**
   – Trained XGBoost & RF models (pick best performer).
   – Serialized pipeline for deployment.

3. **Evaluation Report**
   – ROC AUC, Precision/Recall metrics & curves.
   – Confusion matrix and recommended threshold.



Predictive Modeling Evaluation – Insights

📌 1. Models Evaluated:

- **Random Forest**
- **XGBoost**

Both models were trained on a processed dataset with labeled train/test splits. The goal appears to be **binary classification** (likely predicting a consumer behavior or product preference in the hair care domain).

A. ROC Curve Analysis

Metric	Random Forest	XGBoost
ROC AUC	0.45	0.50
Ideal Value	1.0	1.0
Baseline Value	0.50 (no-skill)	0.50 (no-skill)

➢ Random Forest (ROC AUC = 0.45):

- **Below baseline** performance.
- Indicates that the model is **worse than random guessing**.
- Suggests the model is either overfitting, underfitting, or the features have **very low predictive power**.

➢ XGBoost (ROC AUC = 0.50):

- **Matches random guessing**.
- No discriminatory ability between the two classes.
- Model likely failed to learn useful patterns.

B. Precision-Recall Curve Analysis

Observation	Random Forest	XGBoost
Precision remains low overall	Yes	Yes
Recall increases slowly	Yes	Yes
Curve remains flat	Mostly flat curves	Mostly flat curves
Insights	High class imbalance suspected; models weak at separating classes	

➢ Precision-Recall curves for both models:

- **Flat curves** with very **low precision at all levels of recall**.
- Indicates that when the models **do make positive predictions**, they are **often incorrect**.
- Suggests **class imbalance** or **no strong predictive features** driving model decisions.

C. Interpretation & Recommendations

🔍 Interpretation:

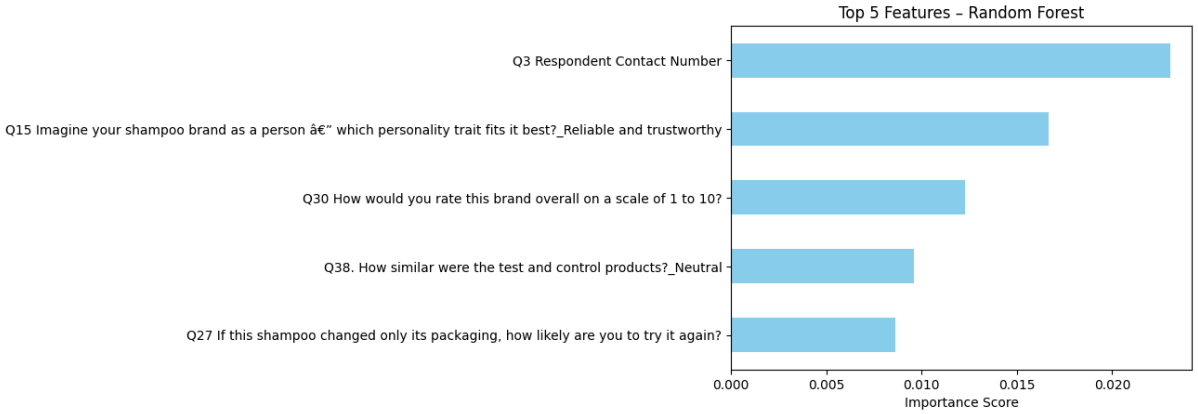
- Both **Random Forest** and **XGBoost models perform poorly** in terms of classification.

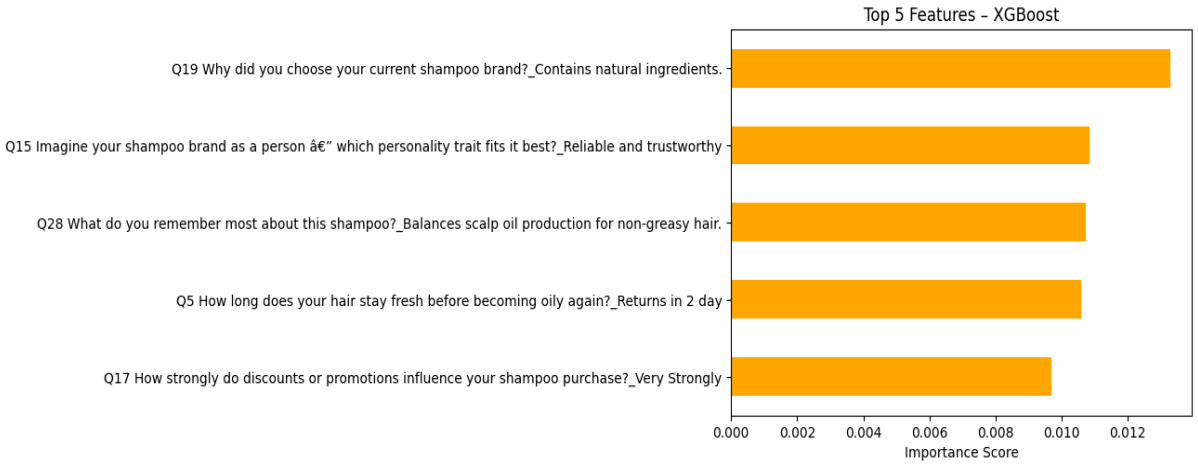
- The **XGBoost model (AUC = 0.50)** and **Random Forest (AUC = 0.45)** indicate a **failure to generalize or learn meaningful patterns**.
- Precision-Recall curves confirm the **ineffectiveness of positive predictions**.

✅ Recommendations:

1. **Feature Engineering:**
    - Re-extract or enrich features with stronger relevance to the target label.
    - Consider feature importance or SHAP analysis.
2. **Class Imbalance Handling:**
    - Use **SMOTE**, **undersampling**, or **class weights** to address imbalance if present.
3. **Model Tuning:**
    - Perform hyperparameter tuning with grid search or randomized search.
4. **Model Comparison:**
    - Try simpler or alternative models: Logistic Regression, LightGBM, or SVM.
5. **Threshold Optimization:**
    - Adjust decision thresholds based on **Precision-Recall trade-offs** using business goals.
6. **Confusion Matrix:**
    - Review it to confirm if one class dominates predictions (e.g., all predicted as majority class).

4. **Driver Analysis**
   – Table/chart of top 5 predictors with interpretation.





Random Forest – Top 5 Predictors

Type: Feature Importance Bar Chart
Interpretation: This shows the top 5 variables (questions) that the Random Forest model identified as most important in predicting the outcome.

Rank	Feature (Question)	Interpretation
1	**Q3 Respondent Contact Number**	This is likely a data leakage or placeholder variable — not a meaningful predictor. Should be removed.
2	**Q15: Brand as a personality – Reliable and trustworthy**	How consumers personify the brand is highly predictive — reliability builds strong brand equity.

3	**Q30: Overall brand rating (1–10)**	General satisfaction with the brand strongly correlates with repeat usage or loyalty.
4	**Q38: Similarity between test and control product – Neutral**	Perception of product consistency affects consumer trust.
5	**Q27: If packaging changed, would you try again?**	Purchase intention post-packaging change is a strong indicator of stickiness to the brand.

✅ **Takeaway**: Psychological branding and overall satisfaction are key drivers. Remove Q3 as it's non-informative.

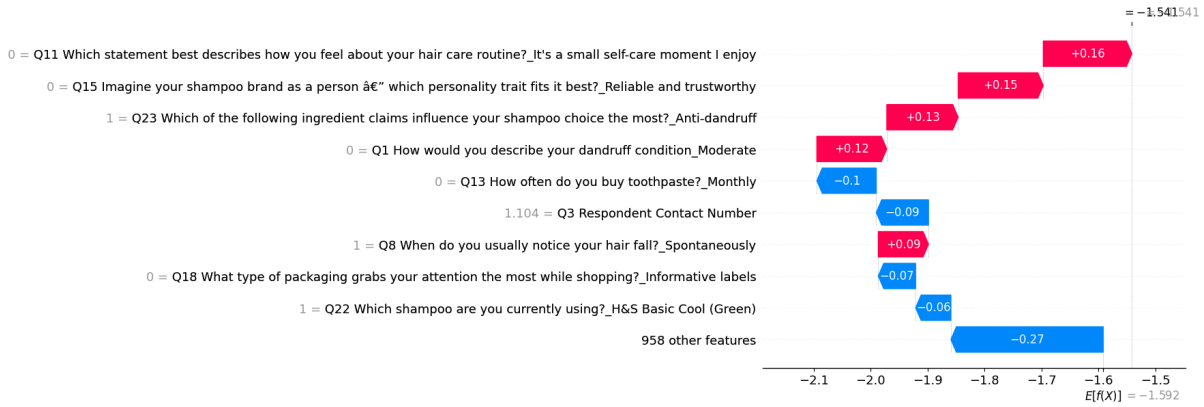
🟧 2. XGBoost – Top 5 Predictors

Type: Feature Importance Bar Chart
Interpretation: XGBoost is another model that gives slightly different drivers due to its algorithm.

Rank	Feature (Question)	Interpretation
1	**Q19: Why did you choose your shampoo brand? → Contains natural ingredients**	Ingredient transparency and “natural” appeal strongly drive brand choice.
2	**Q15: Brand as a personality – Reliable and trustworthy**	Again confirms the importance of emotional branding.
3	**Q28: What do you remember most about the shampoo? → Balances oil production**	Functional benefits (oil control) play a large role in memory and satisfaction.

4	**Q5: How long does hair stay fresh? → Returns in 2 days**	Product efficacy/duration matters. Short-lasting freshness is a concern.
5	**Q17: Influence of promotions → Very strongly**	Promotions/discounts are a tactical but effective lever for some customers.

✅ **Takeaway**: XGBoost highlights **functional benefits + emotional trust + promotional impact** as drivers.



🟥 3. SHAP Summary (SHapley Additive exPlanations)

Type: SHAP Waterfall Plot
Interpretation: SHAP explains the **direction and contribution** of each feature for a single prediction.

- **Positive SHAP values (red)**: Increase the likelihood of a positive outcome (e.g., satisfaction or repeat purchase).
- **Negative SHAP values (blue)**: Decrease the likelihood of a positive outcome.

Feature	Value	Contribution

Q11: Hair care as self-care moment → “It’s a small self-care moment”	+0.16	Personal care link boosts outcome.
Q23: Ingredient claim – Anti-dandruff	+0.15	Specific ingredient benefits positively impact outcome.
Q1: Dandruff condition – Moderate	+0.12	Self-assessed hair issue affects outcome.
Q13: Toothpaste frequency – Monthly	-0.10	Possibly a noisy/unrelated feature.
Q8: Notice hair fall – Spontaneously	+0.09	Reflects health awareness.
Q18: Packaging attention – Informative labels	-0.07	Might imply low attention to packaging despite functional design.
Q22: Currently using → H&S Basic Cool (Green)	-0.27	Specific variant associated with a negative prediction (less satisfaction or loyalty).

✅ **Takeaway**: SHAP provides **nuanced insight** — not just importance, but **direction** of influence. Strongest positive drivers: **ingredient claims, emotional care, brand trust**. Strongest negative: **product variant used, possibly poor-performing SKU**.

📌 Final Summary Across All Models

Theme	Evidence Across Models
Brand personality (trust)	Common top predictor (Q15) in RF & XGB
Functional benefit	Q28 (oil balance), Q5 (freshness)
Promotions	Q17 (XGB)
Product issues or variants	Q22 (SHAP, negative), Q3 (Random Forest, discard)
Ingredient preference	Q19 (natural), Q23 (anti-dandruff)

Description

Objectives

- Segment respondents into 4–6 distinct consumer personas based on key metrics.
- Characterize each persona by demographics, behaviors, and attitudinal scores.
- Validate the segmentation solution using quantitative cluster-validity metrics.

Analysis Tasks

Task	Details	Method / Tools

1. Feature Matrix Construction	Extract and engineer numeric features from Q1–Q30:
	– Severity indices (dandruff, hair fall)
	– Purchase behavior (buying/usage cadence)
	– Sentiment scores (from open-ends)
	– Engagement flags (eye-tracking)
		`pandas`, `NumPy`
		
		`python import pandas as pd import numpy as np df = pd.read_csv("data.csv") features = [ "dandruff_score", "hairfall_frequency", "purchase_frequency", "sentiment_score", "eye_tracking_participation" ] X = df[features].copy()`

2. Preprocessing & Scaling	Handle missing values using imputation. Standardize all numeric features for clustering.	`scikit-learn` `SimpleImputer`, `StandardScaler`
		
		`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 clustering.	`KMeans`, `GaussianMixture` from `scikit-learn`
		
		`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	Compute silhouette scores for each value of k (2 to 8). Compare BIC/AIC for GMM models. Visualize elbow and silhouette plots.	`scikit-learn.metrics`, `matplotlib`
		
		`python import matplotlib.pyplot as plt from sklearn.metrics import silhouette_score sil_scores = [] bic_scores = [] aic_scores = [] for k in range(2, 9): km = KMeans(n_clusters=k, random_state=42) labels = km.fit_predict(X_scaled) sil_scores.append(silhouette_score(X_scaled, labels)) gmm_k = GaussianMixture(n_components=k, random_state=42) gmm_k.fit(X_scaled) bic_scores.append(gmm_k.bic(X_scaled)) aic_scores.append(gmm_k.aic(X_scaled)) plt.plot(range(2, 9), sil_scores, label="Silhouette") plt.plot(range(2, 9), bic_scores, label="BIC") plt.plot(range(2, 9), aic_scores, label="AIC") plt.xlabel("Number of Clusters") plt.ylabel("Score") plt.legend() plt.title("Model Evaluation Metrics") plt.show()`

5. Persona Profiling	For each cluster:
	– Compute feature means
	– Tabulate demographics (age, gender, NCCS)
	– Highlight behavioral and attitudinal traits
		`pandas`, `seaborn`
		
		`python df["cluster"] = kmeans_labels profile = df.groupby("cluster").agg({ "dandruff_score": "mean", "hairfall_frequency": "mean", "purchase_frequency": "mean", "sentiment_score": "mean", "gender": lambda x: x.mode()[0], "age_group": lambda x: x.mode()[0], "NCCS": lambda x: x.mode()[0] }).reset_index() print(profile)`

6. Validation & Refinement	Cross-tab clusters against known variables (e.g., Test vs Control). Bootstrap to check cluster stability.	`scikit-learn`, `resample` from `sklearn.utils`
		
		`python from sklearn.utils import resample 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) scores.append(score) print("Average Silhouette Score (bootstrap):", np.mean(scores))`

Deliverables

Feature Matrix

- Final clean and scaled dataset for clustering
  `python pd.DataFrame(X_scaled).to_csv("feature_matrix_scaled.csv", index=False)`

Clustering Results

- Table: `Respondent_ID → Cluster_Label`
  `python df[["respondent_id", "cluster"]].to_csv("cluster_assignments.csv", index=False)`

- Model evaluation plots (Silhouette, BIC, AIC)

Persona Profiles

- 4 to 6 detailed persona summaries, including:
    - Segment name (e.g., “Value-Oriented Trial Users”)
    - Demographic patterns
    - Key usage and sentiment indicators

Visualization Deck

- Cluster validation plots (elbow, silhouette, BIC)
- Persona heatmaps or radar charts (feature comparison)

Analysis Notebook

- Documented Jupyter notebook including:
    - Feature engineering
    - Clustering logic
    - Evaluation and profiling steps
    - Persona narrative generation