FROM CLUSTERING TO CLASSIFICATION: AN END-TO-END FRAMEWORK FOR CUSTOMER PREDICTION

Group: Five Guys

Group Members:

- Aditya Nuli (2023360)
- Aditya Jaiswal (2023039)
- Amar Anand (2023079)
- Sanskar Garg (2023485)
- Abhishek (2023026)

From Transactions to Targeted Marketing: A Segmentation Approach

Problem Statement: To classify online retail customers into behavioral segments using purely transactional data and build a predictive model to classify new customers into these segments for targeted marketing.

The Two-Stage Pipeline:

- **Inference (Unsupervised):** Used Gaussian Mixture Models (GMM) to discover hidden customer groups based on RFM (Recency, Frequency, Monetary) patterns.
- **Prediction (Supervised):** Trained a classifier to predict these segments for new customers based on early behavior.

The GMM Segments:

- **Champions (5.7%):** High spend, frequent buyers.
- **Potential Loyalists (38.7%):** Recent, moderate frequency.
- **Occasional Buyers (55.5%):** Dormant, low value.

(These 3 segments became the target labels for our prediction model)

Training Model Selection & Rationale

Selected Model: XGBoost Classifier (GMM-Based)

Why XGBoost?

- **Superior Accuracy:** It achieved the highest performance (96.1%) among all tested classifiers when trained on GMM-engineered labels.
- **Robustness:** It effectively handled the class imbalance (where "Champions" represented only ~5.7% of the data) better than linear models.
- **Generalization:** The model showed minimal overfitting, with a tight gap (2.04%) between training and testing accuracy.

Training Configuration:

- **Features (m=16):** Behavioral metrics (e.g., frequency_trend, avg_days_between_orders) independent of the clustering variables to prevent data leakage.
- **Hyperparameters:** Tuned for max depth (d=3) and learning rate (0.05) to optimize the bias-variance tradeoff.

Asymptotic Running Time Analysis

Asymptotic Training Time:

- **The time complexity for XGBoost using the Exact Greedy Algorithm is:** $[O(K \cdot d \cdot ||x|| \cdot \log n)]$
- **Variables:** K (number of trees), d (max depth), $||x||$ (non-missing entries), n (number of samples).
- **Key Factor:** The $\log n$ term comes from sorting feature values at each node to find the optimal split, making it highly efficient.

Empirical Running Time:

- **Actual Training Time:** ~0.30 seconds on the training set (n=3,470).
- **Implication:** The low runtime confirms the model is lightweight enough for frequent retraining cycles in a real-world retail environment.

Prediction Time:

- $O(K \cdot d)$
- Prediction is independent of dataset size n, ensuring instant customer classification at the point of transaction.

Final Model Performance (Train vs. Test)

Performance Summary: Accuracy & Runtime Analysis

Dataset Split	Sample Count(N)	Processing Time	Accuracy
Training Data	3470	0.30	98.13%
Test Data	868	<0.05	96.08

Key Findings:

- **Asymptotic Efficiency:** The training time of ~0.30 seconds confirms the $O(K \cdot d \cdot \log n)$ complexity of XGBoost is highly efficient for this dataset size.
- **Accuracy:** The final test accuracy of 96.1% represents a massive improvement over the K-Means baseline of 74.9%.
- **Robustness:** The high F1-Score of 0.96 indicates the model is not biased towards the majority class, despite "Champions" being a small segment.

Randomized Scaling Technique – Coreset Construction

- **Technique Used:** Coreset Construction via Sensitivity Sampling.
- **Reason for Choice:**
  - Our customer data is imbalanced and clustered. Simple uniform sampling risks missing "rare" but high-value points (like outlying 'Champions').
  - Sensitivity Sampling assigns higher probability to "important" points to guarantee they are included in the small subset (coreset).
- **Methodology:**
  - We ran a lightweight K-Means (k=10) on the training data to find cluster centers for computing sensitivities.
  - Calculated the Sensitivity (s) of each point based on its distance to the nearest center. Points far from centers (informative or rare customers) get higher scores.
  - We sampled the Coreset (Q) based on these probabilities.
- **Formula Used:**

$$p(x) = \frac{s(x)}{\sum_{x' \in X} s(x')} \quad \text{where} \quad s(x) = \min_{c \in C} \|x - c\|^2 + \frac{\lambda}{n}$$

Performance Comparison (Full Data vs. Coreset)

- **Experiment:** Trained the final XGBoost model on Coresets of varying sizes (10%, 30%, 50%) constructed via Sensitivity Sampling and evaluated on the full test set.
- **Results Table:**

Training Data Size	Sample Count (N)	Average Training Time	Test Accuracy	Speedup
100% (Full Data)	3,470	0.33s	96.1%	1.0x
50% Coreset	1,735	0.24s	95.8%	1.35x
30% Coreset	1,041	0.21s	95.2%	1.58x
10% Coreset	341	0.16s	92.0%	2.02x

- **Effective Scaling:** Achieved a 1.6x speedup with negligible loss with the model trained on the 30% coreset.
- **Trade-off:** Extreme scaling (10% data) doubled the speed (2.02x) but caused a noticeable accuracy drop (~4%)

Training Time Improvement After Scaling

- **Key Idea:** Using Sensitivity-Based Coreset Sampling, we reduce the number of training points while keeping the model accuracy nearly unchanged, resulting in substantial speedup.

Asymptotic Training Time Comparison:

- **Full Dataset Training Time (XGBoost):**
  - XGBoost has training complexity $O(n \cdot \log n)$ with respect to the number of input samples n.
- **Coreset-Based Training (Sensitivity Sampling):**
  - model trains on a reduced dataset of size m, where $m < n$.
  - Training complexity becomes $O(m \cdot \log m)$, giving an asymptotic improvement factor of $O(n \cdot \log n)/O(m \cdot \log n)$.
- **Example:**
  - 30% Coreset (m = 1041): $n \cdot \log n / m \cdot \log m \approx 1.6\times$ faster

Conclusion & Future Work

Conclusion:

- **Successful Segmentation:** We utilized GMM to discover 3 statistically distinct customer groups (Champions, Loyalists, Occasional Buyers).
- **High-Accuracy Prediction:** Our optimized XGBoost model achieved 96.1% Test Accuracy, successfully turning unsupervised patterns into a reliable supervised classification system.
- **Scalability:** Demonstrated Coreset Construction (Sensitivity Sampling) as a viable scaling technique. Training on just 10% of the data retained 92% accuracy while delivering a 2x training speedup, validating the asymptotic reduction in complexity.

Future Work:

- **Big Data Implementation:** Test the Coreset technique on a massive dataset (N > 1 million rows).
- **Micro-Segmentation:** Apply the pipeline recursively to the large "Occasional Buyers" segment (55%) to identify "at-risk" vs. "lost" customers for specific re-engagement campaigns.