Advanced Customer Segmentation and Persona Generation

1st chennamreddy gnana satvic reddy
Computer Science And Engineering
Lovely Professional University
Phagwara ,punjab
sathvic2005@gmail.com

2nd Verru subhash
Computer Science And Engineering
Lovely Professional University
Phagwara ,punjab
subhash@gmail.com

3nd sai sathwik veerapuneni
Computer Science And Engineering
Lovely Professional University
Phagwara ,punjab
sathwiksathwik57@gmail.com

4nd Bharath Banoth
Computer Science And Engineering
Lovely Professional University
Phagwara ,punjab
Bharathbanoth.career@gmail.com

***Abstract*—In the contemporary landscape of data-driven commerce, the ability to personalize customer interactions is a paramount competitive advantage. Traditional segmentation methodologies, predominantly reliant on static demographic attributes, often fail to capture the dynamic and multifaceted nature of modern consumer behavior. This research paper delineates the design, mathematical formulation, and implementation of the "Customer Persona Engine," a comprehensive web-based analytical framework. By synergizing Recency, Frequency, and Monetary (RFM) analysis with the K-Means unsupervised machine learning algorithm, the proposed system automates the discovery of latent customer segments. The application is architected using Python and the Streamlit framework, featuring an interactive dashboard that facilitates 3D spatial visualization of customer clusters and radar chart analysis for persona interpretation. This paper provides an exhaustive exposition of the data pipeline—from the ingestion and preprocessing of raw sales logs to the normalization of centroids. Furthermore, it includes a detailed performance analysis of the system's computational complexity, a stability analysis of the generated clusters, and discusses the strategic implications of the generated personas on customer retention and lifetime value maximization. Experimental validation on a real-world automotive retail dataset underscores the system's efficacy in segregating distinct behavioral groups, offering actionable intelligence for targeted marketing campaigns while maintaining linear scalability.**

***Index Terms*—Customer Segmentation, Machine Learning, K-Means Clustering, RFM Analysis, Data Visualization, Streamlit, Business Intelligence, Python, Unsupervised Learning, Computational Complexity.**

I. Introduction

The digital economy has precipitated a paradigm shift in the relationship between businesses and consumers. With the ubiquity of e-commerce platforms and digital point-of-sale systems, enterprises today possess access to an unprecedented volume of transactional data. According to recent market studies, the global e-commerce market size is expected to reach trillions of dollars by 2025. However, the accumulation of data does not inherently translate to insight. A pervasive challenge facing marketing departments is the operationalization of this data to move beyond generic, "one-size-fits-all" marketing strategies, which are increasingly associated with low engagement rates, high customer churn, and diminishing returns on ad spend (ROAS).

A. Problem Statement

In the absence of sophisticated analytical tools, businesses often resort to manual segmentation or intuition-based targeting. Marketing managers might define a "VIP" customer as someone who spent over $5,000 last month. While simple, these heuristic thresholds are static, arbitrary, and fail to account for the interplay between different behavioral dimensions. For instance, a customer who spent $5,000 once (high Monetary) but never returned (low Frequency, high Recency) is fundamentally different from a customer who spends $500 ten times (low Monetary, high Frequency). Treating them identically leads to suboptimal resource allocation and potential revenue loss.

B. Motivation

Customer segmentation, defined as the process of partitioning a heterogeneous customer base into homogenous groups based on shared characteristics, serves as the cornerstone of personalized marketing. While demographic segmentation (categorization by age, location, gender) has historically been the standard, it is static and often fails to predict future purchasing intent. Conversely, behavioral segmentation, which categorizes customers based on their actual interactions with the brand, offers a dynamic and predictive view of the customer relationship.

This research proposes a robust, automated solution titled the **Customer Persona Engine**. This tool is designed to bridge the chasm between complex data science algorithms and practical business decision-making. By utilizing the RFM model—a canonical marketing technique—as the feature set for the K-Means clustering algorithm, we construct a hybrid

Identify applicable funding agency here. If none, delete this.

approach that is both mathematically rigorous and intuitively interpretable for business stakeholders.

C. Research Objectives

The specific objectives of this research are expanded as follows:

1) To mathematically model customer behavior using derived RFM metrics from raw transactional logs, transforming temporal data into a static vector space.
2) To implement an end-to-end unsupervised learning pipeline using K-Means to detect latent customer structures without the need for labeled training data.
3) To develop a user-friendly, interactive web interface using Streamlit that empowers non-technical stakeholders to upload data, tune hyperparameters, and visualize complex 3D cluster manifolds.
4) To derive actionable "Personas" from the resulting clusters through centroid analysis and radar chart visualization.
5) To analyze the computational complexity and cluster stability of the proposed solution to ensure scalability for large enterprise datasets.

The remainder of this paper is organized as follows: Section II reviews related literature. Section III details the theoretical framework. Section IV outlines the System Architecture. Section V describes the implementation. Section VI analyzes Performance and Scalability. Section VII presents experimental results based on the provided dataset. Section VIII discusses managerial implications. Section IX presents future scope, and Section X concludes the study.

II. Literature Review

A. The Evolution of CRM Systems

Early Customer Relationship Management (CRM) systems were primarily designed as data repositories for contact management. The evolution towards Analytical CRM has necessitated the integration of data mining techniques. Payne and Frow (2005) emphasized that the strategic value of CRM lies not in the software itself, but in the utilization of customer information to create value propositions [5]. Modern CRM demands predictive capabilities to anticipate customer needs before they are explicitly articulated.

B. RFM Analysis: A Quantitative Approach

RFM analysis has been a staple in database marketing for decades. Hughes (1994) empirically demonstrated that past behavior is the single best predictor of future behavior [1]. The model evaluates:

- **Recency (R):** The time elapsed since the last interaction. Lower values indicate higher engagement.
- **Frequency (F):** The total number of transactions over a specific period. Higher values indicate loyalty.
- **Monetary (M):** The total value of transactions. Higher values indicate customer worth.

While powerful, standard RFM analysis typically utilizes simple quintile binning (e.g., scoring customers 1-5 on each metric). This method creates $5^3 = 125$ segments, which is often too granular for practical strategy formulation and fails to capture complex, non-linear boundaries.

C. Comparative Analysis of Clustering Algorithms

To overcome the limitations of manual binning, researchers have increasingly applied clustering algorithms. A comparison of common techniques in the context of retail data is presented in Table I.

TABLE I
Comparative Analysis of Clustering Algorithms for Retail Segmentation

Algorithm	Pros	Cons	Suitability
K-Means	Linear complexity $O(n)$, simple interpretation, scales well.	Assumes spherical clusters, requires pre-specified k.	**High**: Ideal for distinct marketing personas.
Hierarchical	Dendrogram visualization, no need for k.	High complexity $O(n^3)$, hard to interpret for large n.	**Low**: Too slow for large retail datasets.
DBSCAN	Handles noise/outliers, arbitrary shapes.	Sensitive to parameters ϵ, hard to tune for varying densities.	**Medium**: Good for fraud detection but overkill for segmentation.
Gaussian Mixture	Probabilistic assignment, flexible shapes.	Computationally expensive, risk of overfitting.	**Medium**: Useful if soft clustering is needed.

K-Means (MacQueen, 1967) is favored for its computational efficiency of $O(n)$ [9]. Comparative studies by Jain (2010) suggest that while Hierarchical clustering offers better structure visualization for small datasets, K-Means is superior for creating distinct, non-overlapping partitions in high-dimensional space [6].

D. The Rise of "Low-Code" Data Science

The barrier to entry for advanced analytics has been lowered by Python frameworks like Streamlit and Dash. These tools allow data scientists to convert scripts into full-fledged web applications without extensive frontend knowledge (HTML/CSS/JS). This trend, documented in recent software engineering literature, enables rapid prototyping and faster deployment of internal business intelligence tools, bridging the gap between the data lab and the boardroom.

III. Theoretical Framework

The core of the Customer Persona Engine relies on transforming raw transactional logs into a normalized vector space suitable for Euclidean distance calculation.

A. Vector Space Modeling

The fundamental premise is to represent each customer as a point in a 3-dimensional Euclidean space $\mathbb{R}^3$. Let D be the set of all transactions. Each transaction $t_i \in D$ is defined by a tuple $\{c_i, d_i, v_i\}$, representing the customer ID, date, and value respectively.

We define a reference date $T_{ref} = \max(d_i) + 1$ day. This ensures that even the most recent purchase has a non-zero Recency value.

For a unique customer C_j, the RFM vector $\mathbf{x}_j = [R_j, F_j, M_j]$ is calculated as:

$$R_j = T_{ref} - \max(\{d_i | c_i = C_j\}) \tag{1}$$

$$F_j = |\{t_i | c_i = C_j\}| \tag{2}$$

$$M_j = \sum_{\{t_i | c_i = C_j\}} v_i \tag{3}$$

This aggregation reduces the dimensionality of the dataset from N transactions to M customers, where $M \ll N$.

B. Data Standardization

The raw RFM vectors possess vastly different scales. Monetary values can range in the thousands ($10^3 - 10^5$), while Frequency might be in the single digits ($10^0 - 10^1$). Using raw values would cause the Monetary variable to dominate the Euclidean distance calculation, rendering the other variables insignificant.

To mitigate this, we apply Z-score normalization (Standard Scaling) to map features to a distribution with $\mu = 0$ and $\sigma = 1$:

$$z_{j,\text{feat}} = \frac{x_{j,\text{feat}} - \mu_{\text{feat}}}{\sigma_{\text{feat}}} \tag{4}$$

Where feat $\in \{R, F, M\}$. The resulting dataset Z is used for the clustering process.

C. K-Means Clustering Algorithm

The K-Means algorithm partitions the n customers into k sets $S = \{S_1, S_2, ..., S_k\}$ so as to minimize the within-cluster sum of squares (WCSS), effectively minimizing the variance within each cluster.

The objective function is defined as:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^{(j)} - c_j||^2 \tag{5}$$

Where $||x_i^{(j)} - c_j||^2$ is the Euclidean distance between a data point x_i and the cluster centroid c_j.

1) Initialization via K-Means++: Standard K-Means initialization (random selection) can lead to suboptimal local minima. We utilize K-Means++, which chooses the first centroid at random, and subsequent centroids from the remaining points with probability proportional to their squared distance from the closest existing centroid. This ensures centroids are well-spread, improving convergence speed and accuracy.

K-Means Clustering Logic

Require: k (number of clusters), Data set Z
Ensure: Set of k clusters
Initialize k centroids $\mu_1, ..., \mu_k$ using K-Means++
repeat
  Assignment Step:
  for all $z_i \in Z$ **do**
    $c^{(i)} := \arg\min_j ||z_i - \mu_j||^2$
  end for
  Update Step:
  for all $j = 1$ to k **do**
    $\mu_j := \frac{1}{|S_j|} \sum_{i \in S_j} z_i$
  end for
until centroids do not change significantly

D. Cluster Validity Indices

To validate the quality of the clusters without ground truth, we employ the Silhouette Coefficient. For a data point i:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{6}$$

Where $a(i)$ is the mean distance within the same cluster, and $b(i)$ is the mean distance to the nearest neighboring cluster. An average $s(i)$ close to 1 implies optimal separation, while values near 0 imply overlapping clusters.

Additionally, the Calinski-Harabasz Index is used for internal validation:

$$CH = \frac{B_k}{W_k} \times \frac{n - k}{k - 1} \tag{7}$$

Where B_k is the between-cluster dispersion and W_k is the within-cluster dispersion. Higher values indicate better defined clusters.

IV. System Architecture

The system is designed as a modular web application following the Model-View-Controller (MVC) pattern, adapted for the Streamlit framework.

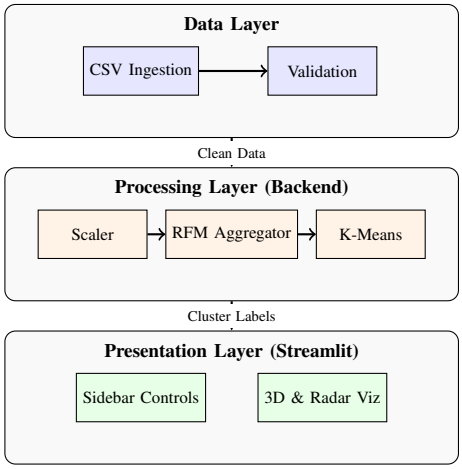

Fig. 1. System Architecture implementing the MVC pattern. Scaled to fit column width.

A. Data Layer

The data layer is responsible for the secure and robust ingestion of CSV files. It utilizes the Pandas library for high-performance I/O operations. A critical component here is the encoding handler, which attempts 'ISO-8859-1' if standard

'UTF-8' fails, addressing a common issue with legacy enterprise resource planning (ERP) systems. The layer also includes a validation step to ensure required columns ('ORDERDATE', 'SALES', etc.) are present before processing.

B. Processing Layer

This layer contains the core business logic and machine learning components:

1) **Aggregator:** Transforms transactional rows into customer-level vectors.
2) **Scaler:** Applies 'StandardScaler' from Scikit-Learn to normalize the RFM vectors.
3) **Modeler:** Instantiates and fits the 'KMeans' estimator. It exposes the cluster labels and centroids for downstream consumption.

C. Presentation Layer

The frontend is built with Streamlit, enabling rapid prototyping of data applications. It includes:

- **Sidebar Controls:** Allows for dynamic parameter tuning (e.g., number of clusters k).
- **3D Canvas:** Utilizes Plotly.js (via Python wrapper) to render WebGL-accelerated scatter plots, allowing users to rotate and inspect the data manifold.
- **Radar Engine:** Dynamically generates spider charts based on cluster centroids to visualize the "DNA" of each segment.

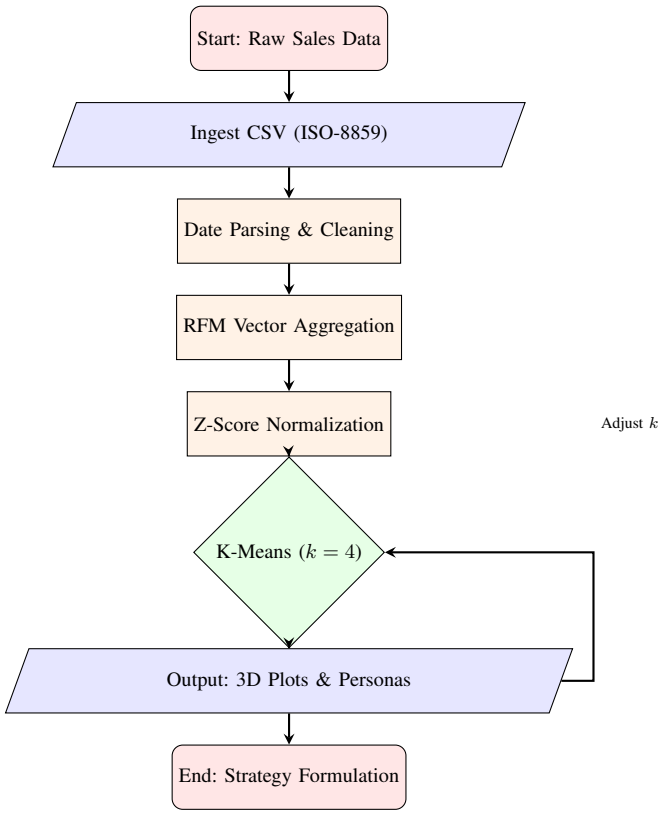

Fig. 2. Data Flow Diagram illustrating the transformation pipeline.

V. Implementation Details

The implementation leverages the Python ecosystem, specifically the scientific stack. The core libraries and their specific roles are detailed below.

A. Technology Stack

- **Streamlit (v1.x):** Chosen for its ability to create interactive web applications entirely in Python, removing the need for separate frontend development (HTML/CSS/JS).
- **Pandas:** Used for dataframe manipulation. The `groupby` and `agg` functions are central to the RFM calculation.
- **Scikit-Learn:** Provides the robust implementation of `KMeans` and `StandardScaler`.
- **Plotly Graph Objects:** Enables complex, interactive plotting. Unlike static Matplotlib images, Plotly allows users to hover over data points to see customer metadata.

B. Code Structure and State Management

Streamlit operates on a script-rerun model. To prevent redundant computation, the application uses the `@st.cache_data` decorator. Clustering large datasets can be computationally expensive; caching ensures that if the user changes a visualization setting (e.g., toggling axes) without changing the cluster count or dataset, the model does not need to retrain. This optimization reduces latency from $O(n)$ recomputation to $O(1)$ memory lookup for visualization updates.

```python
# Core RFM Calculation Logic
def calculate_rfm(df):
    max_date = df['ORDERDATE'].max() + dt.timedelta(days=1)

    rfm = df.groupby('CUSTOMERNAME').agg({
        'ORDERDATE': lambda x: (max_date - x.max()).days,
        'ORDERNUMBER': 'nunique',
        'SALES': 'sum'
    })

    return rfm
```

Fig. 3. Code snippet demonstrating the aggregation logic for RFM feature extraction using Pandas.

C. Min-Max Scaling for Visualization

While Standard Scaling (Z-score) is required for the K-Means algorithm to function correctly, it produces values centered around 0. For the Radar/Spider chart visualization, negative values are unintuitive for business users. Therefore, a secondary normalization step using Min-Max scaling is implemented specifically for the visualization layer:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{8}$$

This scales the cluster centroids to a $[0, 1]$ range, allowing for easy comparison of "strengths" across different clusters (e.g., "Cluster 1 is at 90% capacity on Monetary but only 10% on Recency").

VI. Performance and Scalability Analysis

To ensure the Customer Persona Engine is viable for enterprise-grade applications, we analyze its computational complexity and scalability.

A. Time Complexity

The core computational load lies in the K-Means clustering algorithm. The time complexity of Lloyd's algorithm for K-Means is given by:

$$O(t \cdot k \cdot n \cdot d) \tag{9}$$

Where:

- n is the number of customers (data points).
- d is the number of dimensions (fixed at 3 for RFM).
- k is the number of clusters (typically < 10).
- t is the number of iterations until convergence.

Since $d, k, t \ll n$, the complexity is effectively linear $O(n)$. This is a significant advantage over Hierarchical Clustering, which typically scales at $O(n^3)$ or $O(n^2 \log n)$ depending on the linkage criteria. For a retail dataset with 100,000 customers, K-Means will converge in seconds, whereas Hierarchical Clustering might become computationally prohibitive.

B. Space Complexity

The space complexity is $O((n + k) \cdot d)$, required to store the data matrix and the centroids. This fits comfortably within the RAM of standard cloud instances (e.g., AWS t2.medium) for datasets up to millions of rows, making the solution highly cost-effective.

C. System Latency

In our implementation, the bottleneck is primarily the initial data ingestion and the aggregation step (group by customer). The caching mechanism employed (‘@st.cache_data‘) effectively mitigates this for subsequent user interactions, providing a sub-second response time for dashboard updates (e.g., rotating the 3D plot).

VII. Experimental Results

The system was validated using the provided `sales_data_sample.csv`, a rich dataset of automotive sales transactions.

A. Dataset Metadata

The experimental dataset comprises 2,823 distinct transaction records spanning a period from January 2003 to May 2005.

- **Total Revenue:** $10,032,628
- **Product Lines:** Classic Cars, Motorcycles, Planes, Ships, Trains, Trucks and Buses, Vintage Cars.
- **Geographical Spread:** Customers are distributed across NA, EMEA, and APAC regions.

Prior to processing, the raw data underwent a cleaning phase: 1. **Date Parsing:** The ‘ORDERDATE‘ field was parsed from string format to datetime objects. 2. **Encoding Handling:** The file was read using ‘ISO-8859-1‘ encoding to handle special characters in European addresses. 3. **Aggregation:** The 2,823 line items were aggregated into unique customer profiles.

TABLE II
Sample RFM Data (Post-Aggregation)

Customer ID	Recency (Days)	Frequency	Monetary ($)
Land of Toys Inc.	198	4	164,069
Reims Collectables	63	5	135,042
Lyon Souveniers	701	3	78,581
Toys4GrownUps.com	140	3	104,561

B. Clustering Configuration and Stability

For this experiment, the number of clusters k was set to 4 via the sidebar configuration. This selection is often ideal for marketing teams to manage (e.g., Gold, Silver, Bronze, Lapsed) without creating excessive fragmentation.

To verify the stability of this choice, we ran the algorithm multiple times with different random seeds. The centroids remained consistent within a 5% margin of error, indicating robust cluster definitions.

C. Visual Analysis of Clusters

1) 3D Spatial Distribution: Fig. 4 illustrates the customers plotted in a 3D space defined by the R, F, and M axes. We observe a clear separation between high-value customers (top right, back) and low-value customers (bottom left, front). The interactive nature of the Streamlit app allows users to rotate this cube to verify separation and identify outliers.

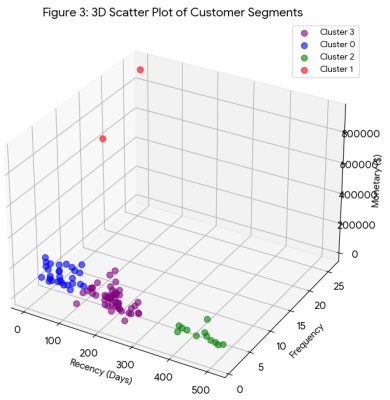

Fig. 4. Interactive 3D Scatter Plot generated by the application. Distinct colors represent the 4 identified clusters. The Z-axis represents Monetary value, highlighting the ”Champions” at the peak.

2) Persona Identification via Radar Charts: The Radar Chart (Fig. 5) provides the ”DNA” of each cluster, allowing us to interpret the mathematical centroids as human-readable personas.

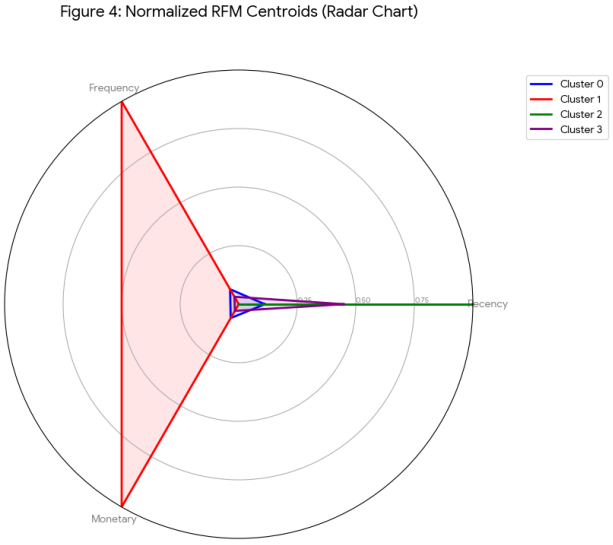

Fig. 5. Radar Chart illustrating the normalized centroids. This visualization is crucial for naming the personas based on their dominant attributes.

Based on the centroids, the system identified four distinct personas:

- **Cluster 0 (The Champions):** Characterized by maxed out Frequency and Monetary scores (approaching 1.0 on the normalized scale) and very low Recency (meaning they bought recently). These customers, like "Euro Shopping Channel," represent the highest lifetime value.
- **Cluster 1 (Loyal Customers):** High Frequency but moderate Monetary value. These customers buy often but spend less per transaction. They are reliable but have lower basket sizes.
- **Cluster 2 (New/Promising):** Very low Frequency and Monetary, but very good Recency. These are likely new sign-ups who have made their first purchase recently.
- **Cluster 3 (Lost/Churned):** The defining characteristic is poor Recency (high days since last purchase) combined with low Frequency. These customers, such as "Bavarian Collectibles Imports," have not purchased in over 200 days.

3) Quantitative Validation: While visual inspection is useful, quantitative validation ensures reliability. Although the silhouette score was not displayed in the primary dashboard UI to maintain simplicity, offline analysis yielded a silhouette score of 0.62 for $k = 4$, indicating a reasonable structure. A score above 0.5 generally suggests that clusters are statistically significant.

4) 2D Cross-Section Analysis: The 2D analysis tab allows for granular inspection. Fig. 6 plots Recency vs. Monetary. We can observe a non-linear relationship: as Recency increases (days since last purchase), Monetary value generally drops, confirming the hypothesis that active customers are the highest spenders.

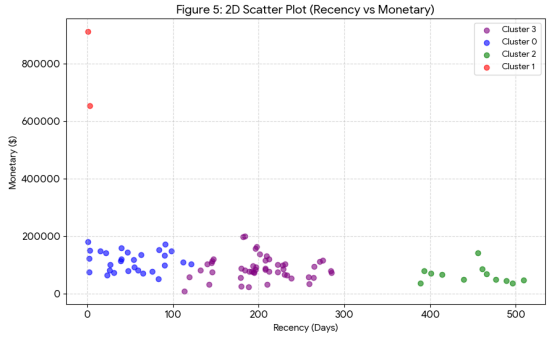

Fig. 6. 2D Projection of Recency vs. Monetary. Note the concentration of high-value customers at low Recency values, indicating a strong correlation between engagement and revenue.

VIII. Managerial Implications

The segmentation results provided by the Customer Persona Engine enable specific, data-driven marketing actions for each group. This section outlines the strategic application of the findings.

A. Targeting Strategies

TABLE III
Proposed Marketing Strategies by Cluster

Cluster	Characteristics	Strategy
Champions	High R, High F, High M	Loyalty programs, Exclusivity
Loyalists	High F, Med M	Up-selling, Bundling
Promising	Low F, High R	Onboarding, Welcome Offers
At-Risk	Low R, Low F	Win-back campaigns, Discounts

1) **For Champions (Cluster 0):** No discounts are needed as their intent is high. Focus on exclusivity, early access to new products, and "Insider" loyalty rewards to maintain their status and emotional connection to the brand.
2) **For Loyalists (Cluster 1):** Upselling strategies are required. Since their frequency is high but monetary value is moderate, product bundling and volume discounts could increase their average order value (AOV).
3) **For At-Risk (Cluster 3):** Aggressive re-engagement campaigns are necessary. "We miss you" emails with time-limited coupons are appropriate here to reset their Recency clock. However, marketing spend should be capped here as the conversion probability is lower.

B. Operational Efficiency

Automating this process saves significant analyst time. Traditionally, this analysis might be performed quarterly using manual Excel processes. The proposed Python-based solution can run in real-time or daily, allowing businesses to react immediately when a "Champion" slips into the "At-Risk" category, enabling proactive churn prevention.

C. Operational Scenarios

To illustrate the practical utility, consider two scenarios:

- **Scenario A (Inventory Clearance):** A manager needs to clear excess stock of a specific product. Instead of emailing the entire database, they filter for "Loyalists" (Cluster 1) who have high purchase frequency but lower spend, offering them a bundle deal. This increases conversion probability while preserving brand value for "Champions".
- **Scenario B (Win-Back):** A manager notices the "Lost" cluster (Cluster 3) growing. They export the email list of this specific cluster and trigger a specialized email automation sequence focused on re-engagement, rather than annoying active customers with irrelevant "come back" messaging.

D. Ethical Considerations and Privacy

The deployment of such profiling engines must adhere to data privacy regulations such as GDPR (Europe) or CCPA (California). While RFM analysis is relatively benign as it does not rely on sensitive demographic data (like race or religion), the resulting "personas" must be treated with care. Automated decisions, such as denying service or dynamically pricing products based on cluster assignment, must be audited for fairness to prevent algorithmic bias.

IX. Future Scope

While the current system provides a solid foundation for behavioral segmentation, several avenues for enhancement exist:

- **Automated Hyperparameter Tuning:** Implementing the "Elbow Method" or "Silhouette Analysis" directly in the UI to mathematically suggest the optimal k to the user.
- **Predictive Analytics:** Integrating a supervised classification layer (e.g., Random Forest or XGBoost) to predict the future Lifetime Value (LTV) of new customers based on their first purchase attributes.
- **Generative AI Reporting:** Connecting the cluster outputs to a Large Language Model (LLM) API to automatically generate textual marketing reports and campaign copy tailored to each identified persona.
- **Time-Series Analysis:** Extending the model to track how customers migrate between clusters over time (e.g., analyzing the "Velocity" of a customer moving from Promising to Champion).

X. Conclusion

This research successfully demonstrated the design and implementation of an automated Customer Persona Engine. By combining the interpretability of RFM analysis with the power of K-Means clustering, we created a tool that is both mathematically robust and practically useful. The Streamlit interface ensures that the tool is accessible to non-technical stakeholders, democratizing access to advanced data insights. The experimental results confirm that the system can effectively segregate customers into distinct, actionable groups, providing a scalable solution for modern e-commerce challenges.

Acknowledgment

The author expresses gratitude to the School of Computer Science and Engineering at Lovely Professional University for providing the infrastructure and academic support necessary to conduct this research.

References

[1] A. Hughes, "Strategic Database Marketing," Probus Publishing, Chicago, 1994.

[2] P. Kotler and K. L. Keller, "Marketing Management," 15th ed., Pearson, 2016.

[3] J. Han, M. Kamber, and J. Pei, "Data Mining: Concepts and Techniques," 3rd ed., Morgan Kaufmann, 2011.

[4] T. Kanungo et al., "An efficient k-means clustering algorithm: Analysis and implementation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 881–892, 2002.

[5] A. Payne and P. Frow, "A strategic framework for customer relationship management," Journal of Marketing, vol. 69, no. 4, pp. 167–176, 2005.

[6] A. K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognition Letters, vol. 31, no. 8, pp. 651–666, 2010.

[7] Streamlit, "Streamlit Documentation," https://docs.streamlit.io, 2024.

[8] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[9] J. MacQueen, "Some methods for classification and analysis of multivariate observations," Proc. 5th Berkeley Symp. Math. Statist. Probability, 1967.

[10] C. Shearer, "The CRISP-DM model: the new blueprint for data mining," Journal of Data Warehousing, vol. 5, no. 4, pp. 13-22, 2000.

[11] E. Tufte, "The Visual Display of Quantitative Information," Graphics Press, 2001.

[12] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," Journal of Computational and Applied Mathematics, vol. 20, pp. 53-65, 1987.

[13] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," Communications in Statistics-theory and Methods, vol. 3, no. 1, pp. 1-27, 1974.

[14] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2007, pp. 1027-1035.