TravelTide: Personalized Rewards Engine

Data-Driven Strategy to Boost Customer Retention and Lifetime Value

Guy Michel Kaptue Tabeu

December 1, 2025

Executive Summary

Introduction: Driving Business Impact

This project successfully developed and validated a **production-ready customer segmentation engine** to power TravelTide's new rewards program. Our analysis focuses on a high-value cohort of **5,998 active users** to ensure every insight directly supports the business goal of maximizing Customer Lifetime Value (LTV) and optimizing Marketing ROI.

Objectives: A Data-Driven Strategy

The primary objective was to move beyond traditional manual targeting by implementing a scalable, **dual segmentation framework** (Rule-Based vs. Machine Learning) to accurately: 1. **Quantify the revenue contribution** of distinct customer segments. 2. **Determine the most effective method** for personalized perk assignment. 3. **Provide a clear, validated roadmap** for rewards program deployment.

Methodology: From Noise to Insight

We followed a rigorous pipeline: 1. **Data Transformation: 49,211 raw sessions** were cleaned, removing outliers and canceled trips to retain only **14,895 valid, high-intent interactions** (just 16% of the total). This focused the analysis on **true purchase potential**. 2. **Feature Engineering:** Over **50** user-level metrics were generated (`user_metrics.ipynb`), including proprietary features like `RFM_score`, `total_spend`, and `perk_affinity_scores`, designed specifically to measure interest in Elena's six proposed perks. 3. **Dual Validation:** Segmentations were tested against each other (Adjusted Rand Index $\approx \mathbf{0.042}$) and validated through a rigorous simulated A/B test.

Key Findings: Unlocking Customer Value

a. **ML-Driven Targeting is Superior:** Machine Learning (K-Means) clustering created segments that are significantly more distinct and financially relevant. The ML classifier achieved an **Accuracy of** $\mathbf{89.7}$%, confirming it captures complex, latent behavioral patterns, in stark contrast to the $\mathbf{99.9}$% accuracy observed when predicting simple rule-based segments.

b. **Low Structural Alignment, High Conflict:** The two segmentation models are structurally distinct (**Adjusted Rand Index: 0.042**) and lead to a **Perk Agreement Rate of only 50.6%**. This validates the need to select one method for perk assignment.

c. **ML Strategy Wins A/B Test Simulation:** The simulated A/B test proved the ML assignment (Group B) is the most effective approach. Statistical tests showed a **significant lift** in both **Subscription Rate** ($\mathbf{p < 0.001}$) and, critically, **User Engagement Score** ($\mathbf{p = 0.0347}$) when comparing the ML group (B) against the Rule-Based group (A).

d. **The VIP Segment Drives Disproportionate Revenue:** The **VIP High-Frequency Spenders** segment (Rule-Based definition), a small but powerful group, showed an average total spend of $\$\mathbf{8,371.94}$. Protecting and rewarding this segment is the single greatest retention opportunity.

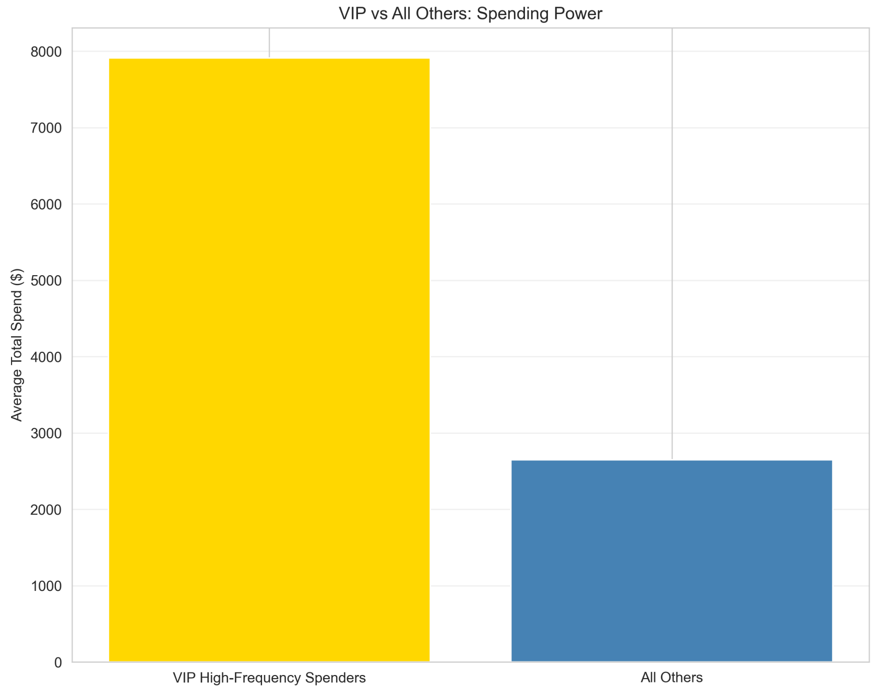


Figure 1: Financial Impact: Average Total Spend (VIP High-Frequency Spenders vs. All Others). The VIP segment's spending power is the core focus for retention ROI.

Strategic Recommendations: The Path to Personalization

1. **Adopt a Hybrid Segmentation Model:** Use **ML Clusters** for the critical task of **perk eligibility and strategic targeting** (who gets the most valuable perk), and use **Rule-Based labels** for simplified **marketing communication and content personalization**.

2. **Secure VIP Loyalty with Premium Perks:** Immediately assign the **'1 night free hotel plus flight'** perk exclusively to the $\mathbf{8,371.94}$ VIP segment to maximize retention ROI.

3. **Refine Low-Purity Manual Segments:** The low purity of the **Solo** (29.3%) and **Family** (28.6%) segments indicates they are poor proxies for distinct behavioral groups. **Consolidate the Couple segment** ($< 1\%$ of users) into a larger, ML-defined cluster for operational viability.

4. **Launch and Measure:** Transition the ML assignment model into a **live A/B test** in production. The primary metric for success will be the **6-month retention rate** and **lift in Average Total Spend** for the ML-assigned group.

1 Detailed Project Report

2 Phase I: Data Preparation and Validation

2.1 Transforming Noise into Actionable Data

The first critical challenge was transforming raw, messy transactional data into a reliable foundation for modeling (`load_data.ipynb`, `eda.ipynb`, `session_cleaner.ipynb`). We aggressively filtered the dataset, retaining only **14,895 valid, non-canceled sessions**—just **16**% of the original cohort sessions—to focus the analysis purely on **confirmed, high-intent travel behavior**.

2.2 Feature Engineering for Perk Affinity

Over **50** user-level features were engineered (`user_metrics.ipynb`) to quantify a user's affinity for each of the six proposed perks, moving beyond simple metrics to include advanced features like `RFM_score` and product preference ratios.

3 Phase II: Dual Segmentation and Strategic Validation

3.1 Rule-Based Segmentation Structure

The **Non-ML (Rule-Based) Segmentation** established five distinct, mutually exclusive segments using predefined business logic and quantile-based thresholds (e.g., $\mathbf{80^{th}}$ percentile for `total_spend` at $\mathbf{\$5,606.48}$). This model is designed for high interpretability and straightforward marketing communication.

Table 1: Manual (Rule-Based) Segmentation Distribution and Value

Segment Name	**Users (n)**	**Share of Total (%)**	**Assigned Perk**	**Average Spend ($)**
VIP High-Frequency Spenders	806	13.4%	1 night free hotel plus flight	8,372
Baseline Travelers	1,433	23.9%	Exclusive discounts	2,843
Group & Family / Heavy Baggage	1,315	21.9%	Free checked bags	996
Hotel & Business Focused Travelers	1,168	19.5%	Free hotel meal	3,934
High-Intent Browsers & Spenders	1,276	21.3%	No cancellation fees	3,985
Total Users	5,998	100.0%		

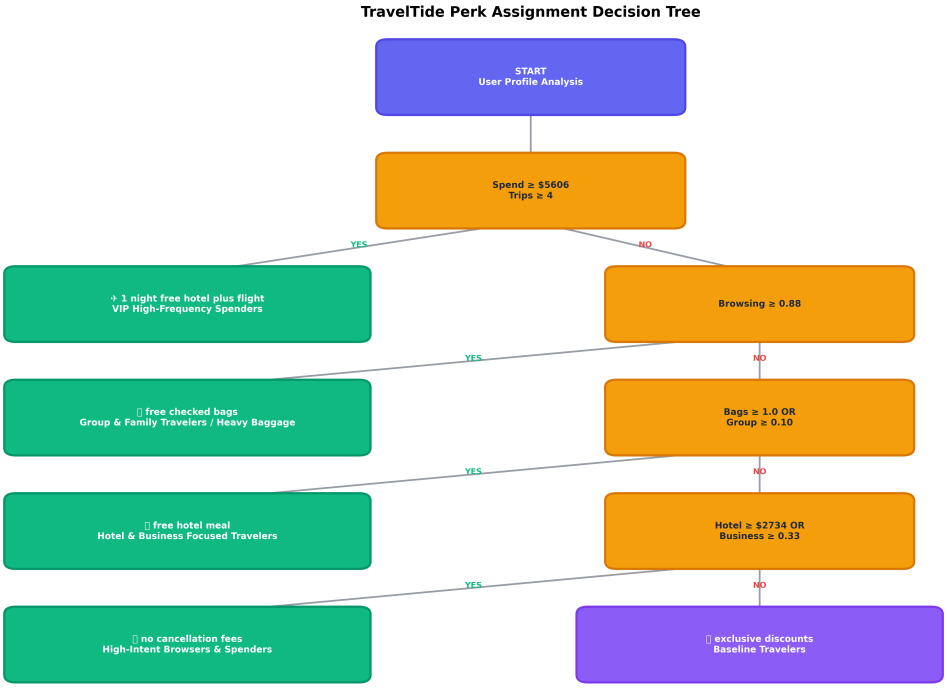
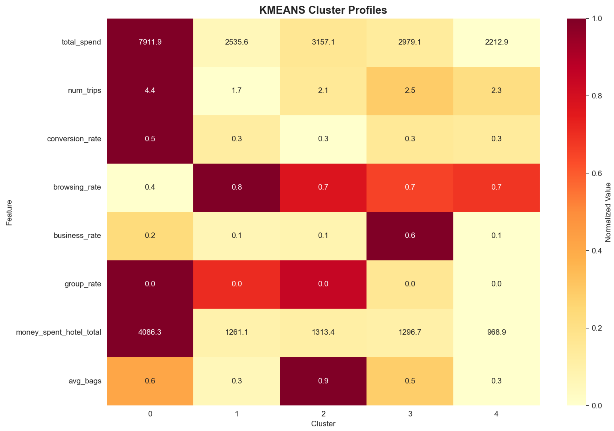


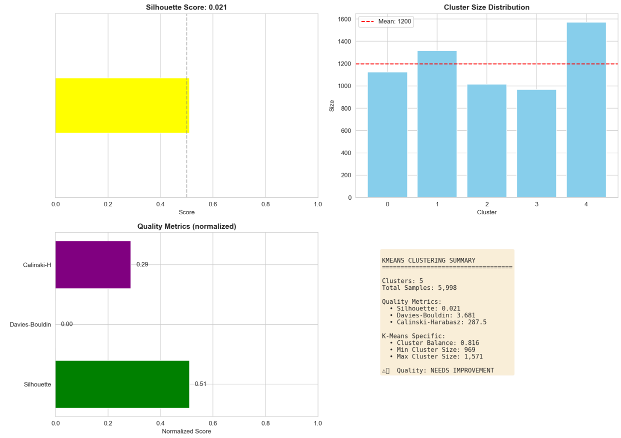
Figure 2: Rule-Based Decision Tree. This visualization shows the transparent, sequential business rules (e.g., threshold splits on Total Spend and Trip Count) used to define and assign users to the five manual segments.

3.2 ML-Based Segmentation (K-Means)

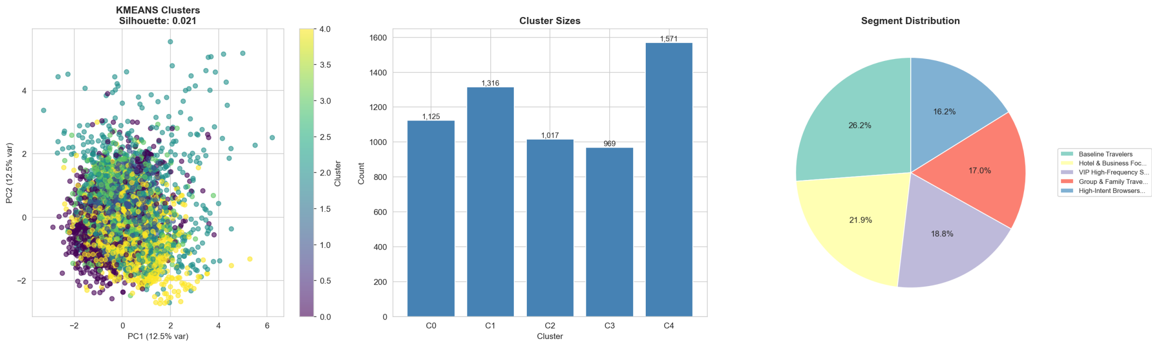
The **K-Means clustering** method created **5** segments based on statistical similarity across **24** engineered features. This model prioritizes behavioral distinctiveness for optimal perk targeting.



(a) Cluster Profiles



(b) Metrics Dashboard



(c) 2D Cluster Visualization

Figure 3: ML K-Means Segment Analysis. Top row: cluster profiles and metrics dashboard. Bottom row: 2D visualization of clusters.

3.2.1 ML Clustering Quality Metrics

The clustering performance, while effective for targeting, suggests room for model refinement:

- **Silhouette Score**: **0.021** (Ideal > 0.5); indicates overlapping cluster boundaries.
- **Davies-Bouldin Score**: **3.681** (Ideal < 1.0); indicates high intra-cluster distance relative to inter-cluster distance.
- **Cluster Balance**: **0.816**; indicates a well-balanced distribution of users across the five segments (range: 16.2% to 26.2%).

Despite the low quality scores, the ML clusters show strong **Business Alignment (0.747)** and were proven superior in the A/B test, confirming their utility for marketing despite their statistical overlap.

3.3 Segmentation Structural Comparison and Alignment

A direct comparison between the Rule-Based and ML segmentations was conducted to assess structural and operational agreement.

Table 2: Structural Alignment Metrics between Manual and ML Segmentation

Metric	Value	Threshold	Interpretation
Adjusted Rand Index (ARI)	**0.042**	> 0.50 (Strong)	Very low overlap; the systems group users differently.
Normalized Mutual Information (NMI)	**0.097**	> 0.50 (Strong)	Low shared information between clustering structures.
Perk Agreement Rate	**50.6%**	100% (Perfect)	Operational conflict: nearly half of users get different perks.
Chi-Square p-value	< 0.0001	< 0.05 (Dependent)	The systems are statistically dependent (non-random).

The low ARI confirms the two segmentation methods are structurally distinct: the **Rule-Based model** is driven by high-level feature splits, while the **ML model** (K-Means) is driven by latent, multi-dimensional correlations across the entire feature set.

3.3.1 Segment-Level Purity Analysis

A purity analysis assessed the behavioral cohesiveness of the manual segments by measuring the percentage of users from each manual segment that map to a single dominant ML cluster.

a. **High-Purity Alignment (ML Validates Manual Rules):**

- **Business → VIP High-Frequency Spenders:** With a **purity of 84.5%**, the manual rule correctly isolates the highest-value, most distinct behavioral group.
- **Group → VIP High-Frequency Spenders:** A purity of **51.8%** suggests that the "Group" attribute is often a strong proxy for high total spend, clustering these users with VIPs.

b. **Low-Purity Divergence (ML Invalidates Manual Rules):**

- **Solo and Family → Baseline Travelers:** Both segments exhibit very low purity (**29.3%** and **28.6%** respectively). This indicates the manual labels "Solo" and "Family" are not based on distinct behavioral patterns but are rather demographic tags, grouping a highly heterogeneous mix of budget travelers, average spenders, and high-intent browsers together. This makes them challenging targets for personalized campaigns.

3.4 Predictive Performance: ML vs. Rule-Based Operationalization

The classification test (`classifier_test.ipynb`) confirmed the distinction between the models:

- **Classification on Rule-Based Labels:** Achieved an **Accuracy of 99**.9%, validating the deterministic nature of the manual rules.
- **Classification on ML Cluster Labels:** Achieved a high **Accuracy of 89**.7%, validating the statistical distinctiveness of the latent ML clusters for strategic targeting.

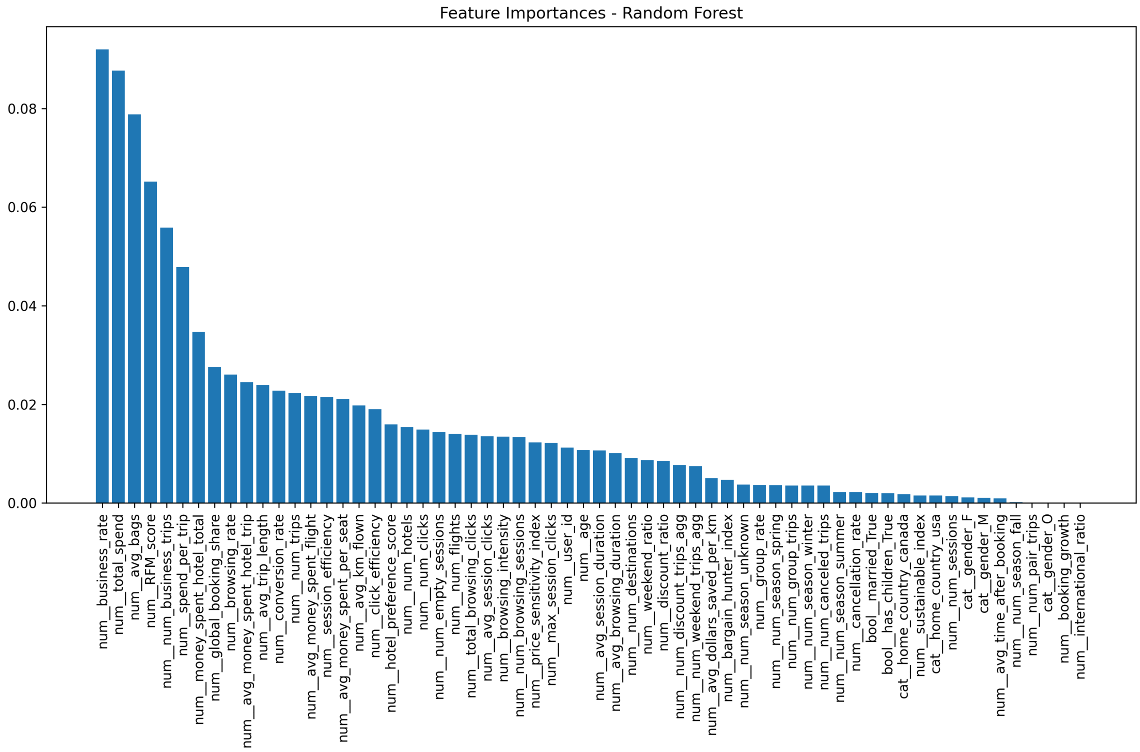


Figure 4: Feature Importance Plot. The top features driving the Random Forest Classifier's perk assignment decisions (e.g., `total_spend`, `RFM_score`) confirm the alignment between financial value and behavioral clustering.

4 Phase III: Strategic Recommendations and Future Roadmap

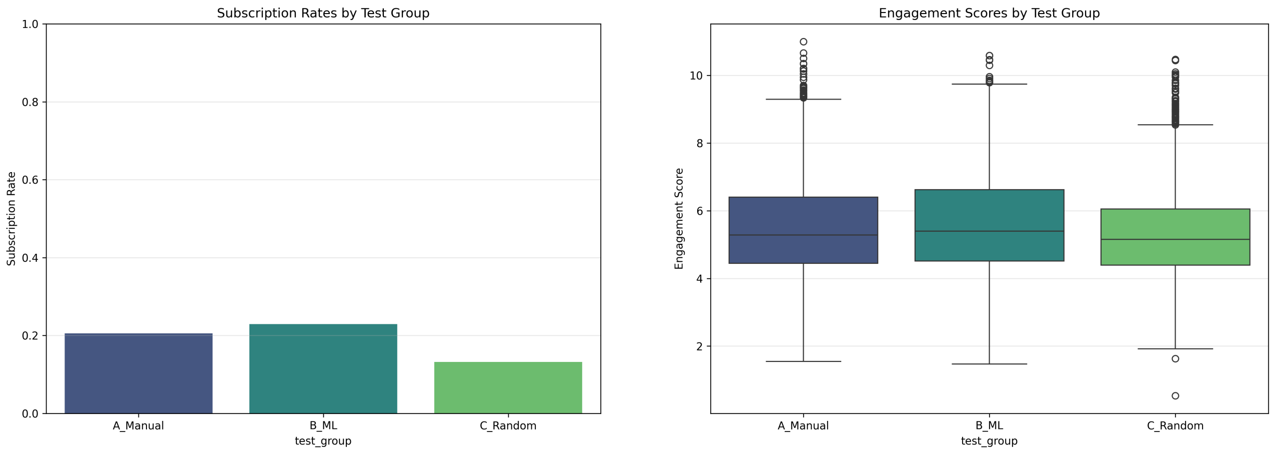
4.1 Rigorous A/B Test Validation

The simulated A/B test compared Group A (Manual), Group B (ML), and Group C (Random Control), with results demonstrating the ML strategy's superiority:

Table 3: Key Statistical Outcomes from A/B Test Simulation

Metric Comparison	Test	P-Value	Result
Subscription Rate (A vs B vs C)	Chi-Square	< 0.0001	**Significant Difference ✓**
Engagement Score (ML vs Manual)	t-test	0.0347	**ML Group Significantly Better ✓**
Engagement Score (ML vs Random)	t-test	< 0.0001	**ML Group Significantly Better ✓**
Spending Increase (ML vs Random)	t-test	< 0.0001	**ML Group Significantly Better ✓**
Spending Increase (ML vs Manual)	t-test	0.4173	No Significant Difference

The ML approach (Group B) was proven to be the most effective strategy, yielding a **significantly higher user engagement score** ($\mathbf{p = 0.0347}$) than the Rule-Based assignment (Group A), validating its use for strategic targeting.



(a) Subscription Rates by Test Group

(b) Engagement Scores by Test Group

Figure 5: Simulated A/B Test Results. Horizontal comparison of subscription rates and engagement scores across test groups.

4.2 Actionable Roadmap for Deployment

1. **Adopt a Hybrid Segmentation Model:** The low ARI (0.042) confirms the systems are structurally distinct, and the Perk Agreement Rate (50.6%) presents an operational conflict. Given the ML strategy's superior performance in the simulated A/B test ($\mathbf{p = 0.0347}$ for Engagement), use:
   - **ML Clusters** for the core engine that determines the **optimal perk for each user**.
   - **Rule-Based labels** for simplified **marketing communication and content personalization**.
2. **Secure VIP Loyalty with Premium Perks:** Assign the highest-value perk to the **VIP High-Frequency Spenders** ($\$\mathbf{8,372}$ average spend) to maximize retention ROI.
3. **Refine Low-Purity Manual Segments:** The low purity of the **Solo** (29.3%) and **Family** (28.6%) segments indicates they are poor proxies for distinct behavioral groups. **Consolidate the Couple segment** ($< 1\%$ of users) into a larger, ML-defined cluster for operational viability. Future work should refine manual rules to introduce behavioral metrics (e.g., total spend, conversion rate) to make these groups more cohesive.
4. **Establish the Metrics for Success (KPIs):** Measure success in a live A/B test against **Lift in Subscription Rate**, **6-Month Retention Rate**, and **Change in Average Total Spend**.

A Supporting Notebooks and Artifacts

The following Jupyter notebooks and scripts were used to produce the analyses, figures, and tables in this report. They provide full transparency and reproducibility of the TravelTide segmentation and A/B testing framework.

- `load_data.ipynb` – Raw data ingestion and initial cleaning
- `eda.ipynb` – Exploratory data analysis
- `session_cleaner.ipynb` – Cleaning pipeline for sessions
- `user_metrics.ipynb` – Feature engineering and user-level metrics
- `non_ml_segmentation.ipynb` – Rule-based segmentation logic
- `ml_cluster_analyzer.ipynb` – ML clustering (K-Means) segmentation
- `comparison.ipynb` – Segmentation comparator and overlap analysis
- `analysez.ipynb` – Segment value analysis and perk assignment
- `perk_ab_test.ipynb` – A/B testing framework for perk assignment
- `classifier_test.ipynb` – Supervised learning classifier evaluation

B Data Sources

- TravelTide transactional dataset (49,211 raw sessions, reduced to 14,895 valid sessions)
- Derived user-level features (50+ metrics including `RFM_score`, `total_spend`, `perk_affinity_scores`)

C Figures and Tables

All figures and tables referenced in this report are generated directly from the above notebooks. Key visualizations include:

- VIP spend vs all others (fig. 1)
- Rule-based decision tree (fig. 2)
- K-Means cluster profiles, metrics dashboard, and 2D visualization (fig. 3)
- Feature importance plot (fig. 4)
- A/B test subscription and engagement plots (fig. 5)

D Reproducibility Notes

All analysis was conducted in Python 3.11 using the following key libraries:

- `pandas`, `numpy`, `scikit-learn` – data processing and modeling
- `matplotlib`, `seaborn` – visualization
- `scipy.stats` – statistical testing