99 Little Orange DataSet

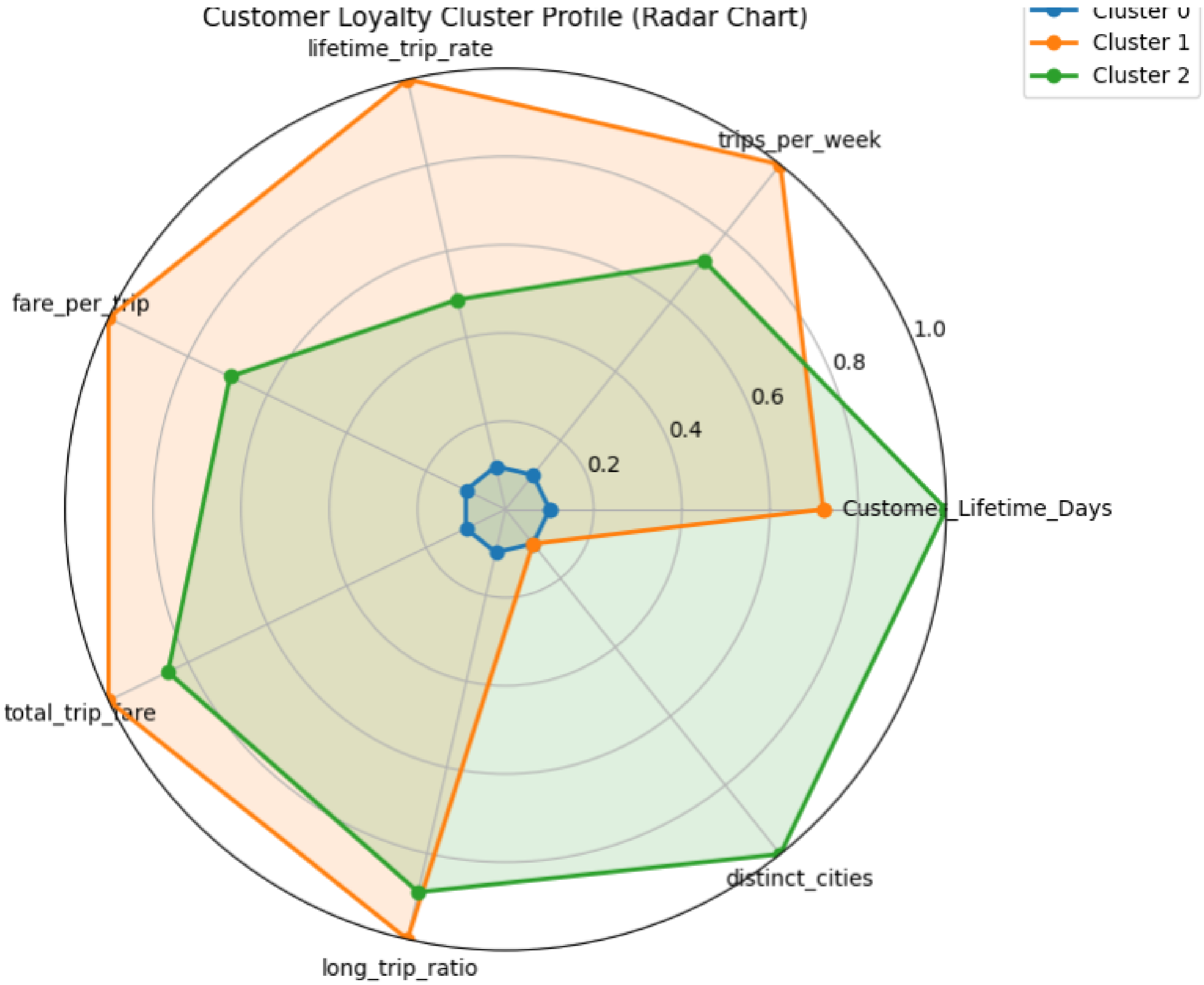
WHICH CUSTOMERS ARE MORE LIKELY TO CHURN

Business Overview

99 Little Orange operates a **two-sided marketplace** connecting **drivers (supply)** and **passengers (demand)**.

The main goal of the business is to keep the marketplace balanced:

- **High demand, low supply →** surge pricing, longer wait times and passenger dissatisfaction.
- **High supply, low demand →** driver idle time and supply drop-off.



Our Clustering Findings (For Last Time)

Casual Customers (≈194k users)

- Very low trip frequency
- Lowest spending and shortest lifetime
- These users naturally churn and are not critical for long-term retention.

Moderate-Usage Customers (≈891 users)

- Highest lifetime (~306 days)
- Travel across more cities on average
- Healthy weekly activity and strong overall engagement
- This group gives us the most stability.

High-Usage Customers (≈39k users)

- High spending and strong trip frequency
- Lifetime around ~282 days
- Important revenue drivers and very valuable to retain.

Key Parameters That Drive The Dashboard

- Number of clusters
- Churn inactivity threshold (days)
- Cluster selection
- Date range

Our Visualization Core Concepts

Flexibility

Different types of users (analysts, managers, decision-makers) can interact with the dashboard according to their needs.

Filters, sliders, and multi-select options allow anyone to explore behavior at different levels of detail.

Dynamic Reflection

Every parameter change (clusters, churn threshold, date range) updates all visuals instantly.

Users immediately see how behavior shifts without rerunning code or refreshing the environment.

Interpretability

Visuals such as radar charts, scatter plots, and KPIs simplify complex ML outputs.

Clusters are labeled (A, B, C…) based on business importance to make results easy to understand.

Action-Driven Insights

Who is churning?

Which clusters are most valuable?

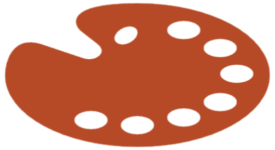
How do behavior patterns change across segments?

The Implementation Steps

Libraries Used:

Marimo for dynamic modeling

matplotlib and seaborn for visualization



Visualization Layer:

Built reusable plotting functions (radar, bar/countplot, scatter)

Used custom palettes + cluster coloring to improve interpretation

KPI cards computed and displayed (churn rate, active users, cutoff date)

Visuals reflect user-selected filters (date range, clusters, features)

How Marimo Powers Our Interactive Dashboard

Fully Reactive Execution

Marimo automatically re-executes only the cells affected by a user's action.

When a slider, dropdown, or date filter changes, all relevant visuals, KPIs, and tables update instantly.

Enables real-time exploration without rerunning the notebook or refreshing the interface.

Composable UI Components & Live Data Interaction

Marimo provides a rich set of widgets—sliders, multiselects, dropdowns, date pickers, buttons—allowing flexible user input.

These controls make the dashboard accessible to non-technical users while still driving complex ML operations in the background.

Every interaction (changing clusters, churn threshold, selected features, etc.) triggers immediate recalculation of metrics and charts.

Seamless Integration With Matplotlib

Although Matplotlib is a static visualization library, Marimo renders each figure cleanly and responsively inside the dashboard.

Custom Matplotlib charts—such as our radar plots, scatter plots, and countplots—are displayed through mo.ui.matplotlib(fig).

The integration allows us to keep familiar Python plotting workflows while benefiting from Marimo's interactive layer.

The result: traditional Matplotlib visuals behave as reactive dashboard components, updating whenever filters or parameters change.

So, Let’s see the dasboard

Any Questions

Thank you