Dynamic Customer Segmentation and Forecasting Analysis

1. Project Overview / Executive Summary

Domain: Retail / E-Commerce

Project Type: End-to-End Data Analytics & Machine Learning Application

Summary: In the highly competitive retail landscape, understanding customer behavior is paramount. Traditional segmentation methods often treat customers as static entities, failing to capture behavioral changes over time (e.g., a loyal customer gradually becoming inactive). This project implements a **Dynamic Customer Segmentation** system using the **LRFM (Length, Recency, Frequency, Monetary)** model combined with **Multivariate Time-Series Clustering**.

The goal was to transition from reactive marketing to proactive engagement by identifying behavioral *trajectories*. The solution includes a predictive forecasting module to estimate future spending for specific segments. The final output is a deployed interactive web dashboard that empowers business stakeholders to visualize customer personas and future revenue trends.

Tech Stack:

- **Language:** Python 3.10
- **Data Manipulation:** Pandas, NumPy
- **Machine Learning:** Scikit-learn, Tslearn (TimeSeries K-Means), Facebook Prophet (Forecasting)
- **Visualization:** Plotly, Matplotlib
- **Deployment/Dashboarding:** Streamlit

2. Business Problem and Objectives

The Business Scenario

A bicycle retail business has accumulated 12 months of transactional data. Currently, the marketing team treats all customers with similar *total* spend as identical. However, a customer who spent $500 in January and stopped is fundamentally different from a customer who spent $500 last week. The business is wasting budget on generic campaigns that do not resonate with where the customer is in their lifecycle.

Problem Statement

The business lacks visibility into **customer trajectories**. They cannot distinguish between

"Churning High-Value Customers" and "Growing New Customers," leading to missed revenue opportunities, inefficient ad spend, and higher churn rates.

Objectives

1. **Segment Customers Dynamically:** Group customers based on their behavior *over time* rather than static aggregates.
2. **Identify Key Personas:** Define clear segments (e.g., Loyal Champions, At-Risk Sleepers).
3. **Forecast Revenue:** Predict the future spending potential of each segment for the next 6 months.
4. **Actionable Insights:** Provide data-driven recommendations for targeted marketing strategies.

Success KPIs

- **Silhouette Score:** Measure of cluster separation quality (Target > 0.4).
- **Segment Interpretability:** Are the clusters distinct and business-relevant?
- **Forecast Uncertainty Interval:** Reliability of the 6-month revenue prediction (Target: Narrow confidence bands).

3. Data Collection

Data Sources

The data consists of three structured CSV files representing a relational database export from the year 2017.

Extraction Process

Data was ingested locally using Python’s Pandas library. The process involved reading raw CSVs, checking for encoding errors (UTF-8), and validating column headers against a data dictionary.

Data Volume

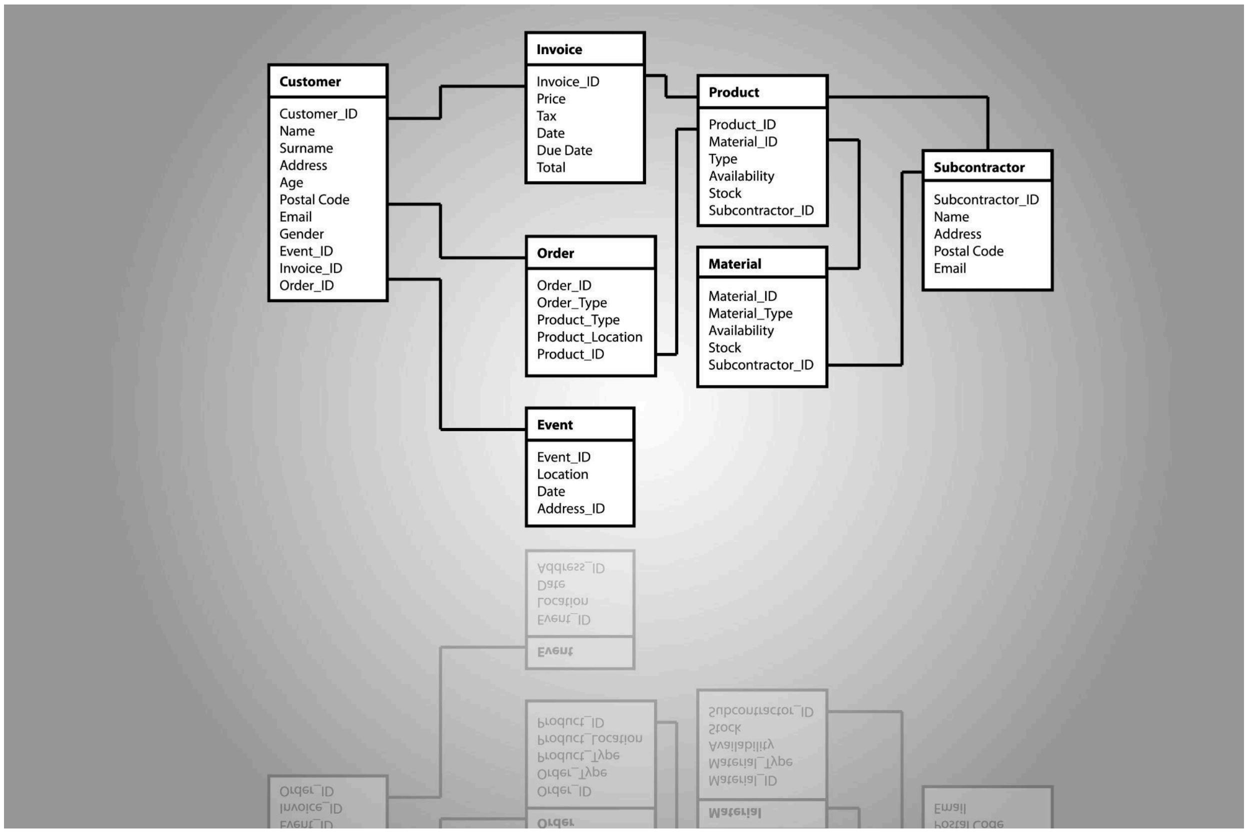
- **Transactions:** ~20,000 rows (Individual purchase records).
- **Demographics:** ~4,000 rows (Customer profile data).
- **Addresses:** ~4,000 rows (Geographic data).

Limitations

- **Time Horizon:** The dataset is limited to a single year (2017), which constrains long-term seasonality analysis (e.g., year-over-year growth).
- **Missing Features:** "Satisfaction" data (S in LRFMS) was unavailable, so the model was adapted to a practical **LRFM** implementation.

4. Data Description

The project relies on a Schema structure where customer_id acts as the primary key linking all tables.



Shutterstock

Explore

1. Transactions_Cleaned.csv

Column	Type	Description
transaction_id	Integer	Unique identifier for the transaction.
customer_id	Integer	Foreign key linking to the customer.
transaction_date	Date	Date the purchase

		occurred.
list_price	Float	The revenue generated from the item (Monetary value).
order_status	String	Status of order ('Approved', 'Cancelled').

2. CustomerDemographic_Cleaned.csv

Column	Type	Description
customer_id	Integer	Unique identifier.
gender	String	Gender of the customer.
wealth_segment	String	Wealth classification (e.g., Mass Customer, Affluent).
job_industry	String	Industry the customer works in.

3. CustomerAddress_Cleaned.csv

Column	Type	Description
customer_id	Integer	Unique identifier.
state	String	State of residence (NSW, VIC, QLD).
property_valuation	Integer	Valuation score of customer property.

5. Data Cleaning and Preprocessing

Data quality is critical for time-series modeling. The following steps were executed in the load_and_prepare_data pipeline:

Step 1: Merging Datasets

- **Action:** Left join performed on Transactions with Demographics and Address using customer_id.
- **Why:** To create a Master Analytical Base Table (ABT) that contains both behavioral (spend) and descriptive (demographic) attributes for post-analysis profiling.

Step 2: Date Conversion

- **Action:** Converted transaction_date from object/string to datetime objects.
- **Why:** Essential for time-series resampling, sorting, and period extraction.

Step 3: Filtering Invalid Orders

- **Action:** Filtered dataset to keep only rows where order_status == 'Approved'.
- **Why:** Cancelled orders do not generate actual revenue. Including them would inflate the Monetary value and skew the model.

Step 4: Handling Missing Values

- **Action:** Dropped rows where customer_id or list_price was null.
- **Why:** These are critical fields; without an ID, behavior cannot be tracked. Without price, value cannot be calculated. Imputation would introduce unacceptable bias in financial metrics.

6. Exploratory Data Analysis (EDA)

Before modeling, an EDA was conducted to understand the data distribution:

- **Order Status Distribution:** It was observed that <1% of orders were cancelled. However, these often represented high-value items, justifying the rigorous filtering step.
- **Revenue Seasonality:** A slight dip in revenue was observed in mid-2017 (Winter in Australia), with peaks in October and December, indicating seasonal holiday shopping behavior.
- **Demographic Split:** The majority of customers fell into the "Mass Customer" wealth segment, with a roughly even split between genders.
- **Geographic Distribution:** Most customers resided in NSW (New South Wales), suggesting a targeted geographic market.

7. Feature Engineering (LRFM Modeling)

The core innovation of this project is the **Dynamic LRFM** feature engineering. Instead of one summary per customer, we generated a **Time Series** for every customer for every month.

Granularity: Monthly Resampling (.resample('M'))

Metrics Calculated Per Month:

1. **Length (L):**
   - *Formula:* (Current Month End Date - First Purchase Date).days
   - *Business Logic:* Measures loyalty. A customer who bought their first item 300 days ago has higher L than a new joiner.
2. **Recency (R):**
   - *Formula:* (Current Month End Date - Last Purchase Date in Period).days
   - *Business Logic:* Measures engagement/risk. A gap in purchasing increases this value. High R = Churn Risk.
3. **Frequency (F):**
   - *Formula:* Count(Transaction IDs)
   - *Business Logic:* Measures purchase habit/stickiness.
4. **Monetary (M):**
   - *Formula:* Sum(List Price)
   - *Business Logic:* Measures financial value to the firm.

Transformation & Scaling

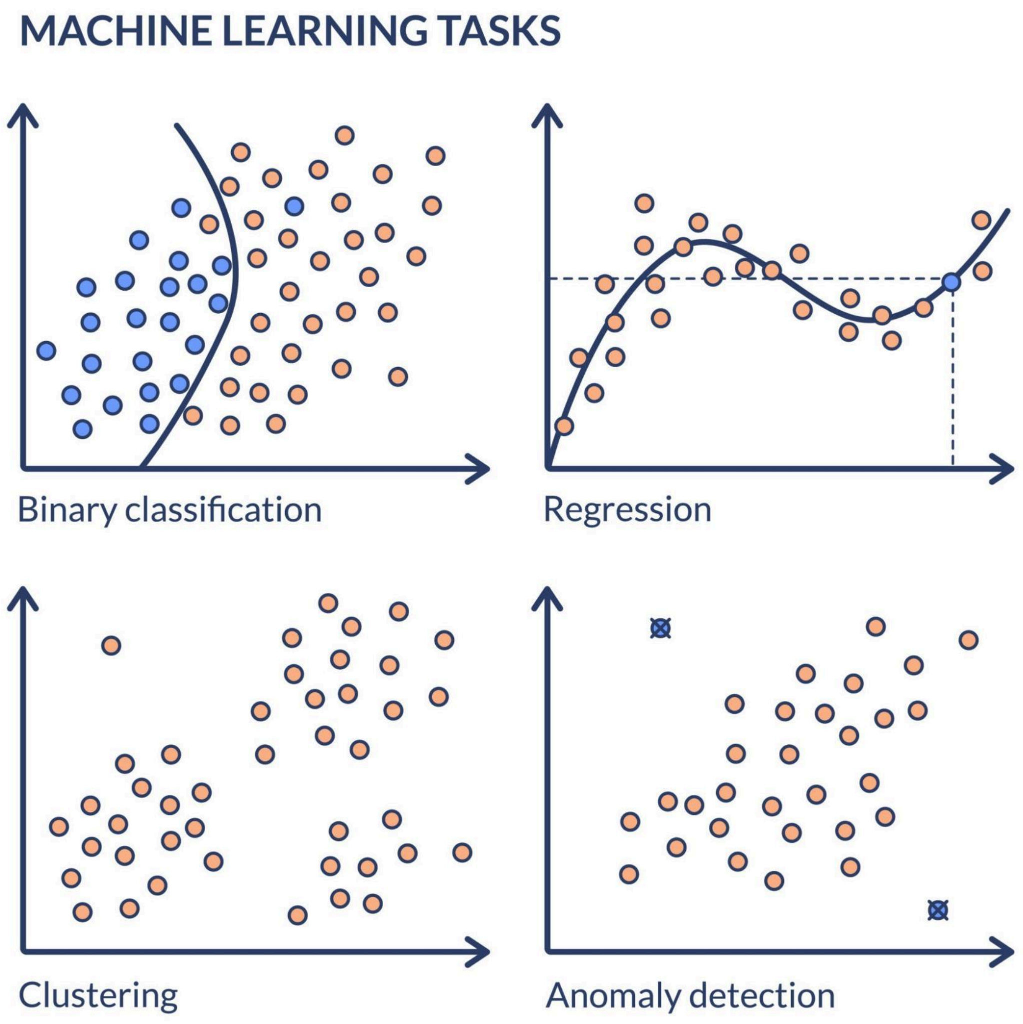
- **Log Transformation:** Not applied, as raw patterns were preferred for interpretability.
- **MinMax Scaling:** Data was scaled using TimeSeriesScalerMinMax to normalize values between 0 and 1.
  - *Why:* To prevents 'Monetary' (values in thousands) from mathematically dominating 'Frequency' (values < 10) during the Euclidean/DTW distance calculation.

8. Data Modeling

Clustering Algorithm: TimeSeries K-Means

Unlike standard K-Means which clusters static points, **TimeSeries K-Means** clusters "paths" or trajectories.

- **Library:** tslearn
- **Metric: Dynamic Time Warping (DTW)**
  - *Rationale:* DTW allows the algorithm to recognize similar shapes even if they are shifted in time. For example, Customer A buying in March vs. Customer B buying in April will still be clustered together if their purchase magnitude and frequency are similar.
- **Clusters (k):** Optimized using the **Silhouette Score**, identifying **k=4** as the optimal number of segments for distinct business interpretability.



Getty Images

Forecasting Algorithm: Facebook Prophet

- **Algorithm:** Prophet (Additive Regression Model).
- **Input:** Aggregated monthly revenue per cluster (Time Series).
- **Output:** 6-month revenue forecast with confidence intervals (yhat_lower, yhat_upper).
- **Why Prophet:** It is robust to missing data and handles seasonality well without extensive parameter tuning.

9. Insights and Recommendations

Based on the clustering output, four distinct personas were identified:

Persona	Characteristics (LRFM Profile)	Business Recommendation
Loyal Champions	High Monetary, High Frequency, Low Recency. Consistent active buyers.	**Reward:** Offer VIP loyalty tiers, early access to new product launches to maintain status.
At-Risk Sleepers	High historical spend, but Recency is strictly increasing (stopped buying 3+ months ago).	**Win-Back:** Trigger aggressive "We Miss You" discounts or surveys to identify friction points.
Steady Supporters	Moderate spend, regular frequency. The stable revenue base.	**Upsell:** Bundle offers (e.g., Helmet with Bike) to increase Average Order Value (AOV).
New Potentials	Low Length, Low Frequency. Recently acquired customers.	**Nurture:** Send welcome series emails and educational content to integrate brand into their lifestyle.

10. Dashboard Documentation

The project is deployed as a **Streamlit** web application to democratize access to these insights.

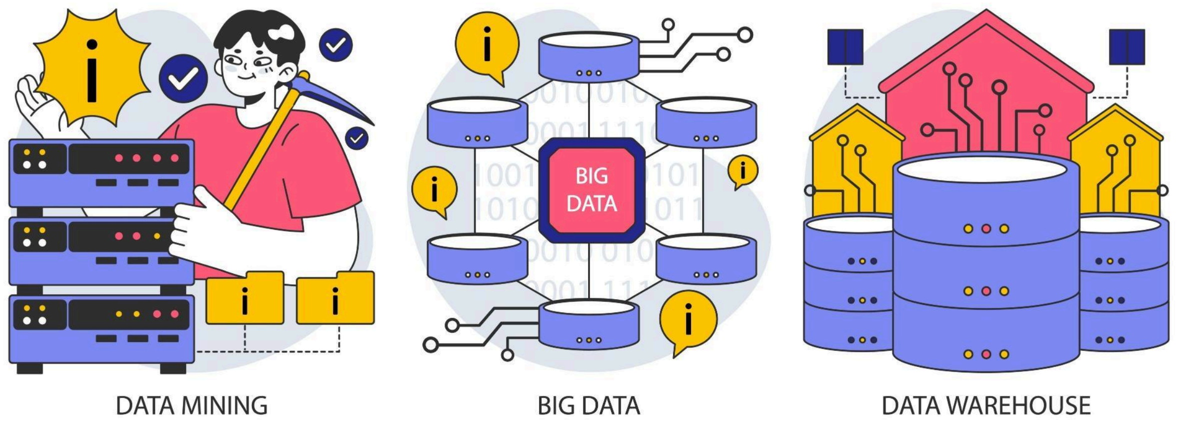
Dashboard Architecture:

1. **Sidebar Controls:**
   - k Selection: Slider to adjust number of clusters (2-8).
   - Metric Selection: Toggle between 'Euclidean' (Fast) and 'DTW' (Accurate).
2. **Top-Level KPIs:**
   - Total Revenue, Avg Revenue per Customer (ARPC), Total Customer Count.
3. **Visual 1: Behavioral Profile (Radar Chart):**
   - *Library:* Plotly.
   - *Insight:* Compares the selected segment's L, R, F, M scores against the population average to visually define the persona.
4. **Visual 2: Wealth Distribution (Donut Chart):**
   - *Insight:* Shows demographic breakdown (e.g., "Are our Champions mostly Affluent?").
5. **Visual 3: Future Spend Forecast (Line Chart):**

- *Library:* Plotly (integrated with Prophet).
- *Insight:* Shows historical dots and a predictive line with a blue shaded confidence interval for the next 6 months.

11. End-to-End Pipeline and Architecture

The system follows a modular ETL-ML pipeline architecture:



Shutterstock

Explore

Workflow:

1. **Ingestion Layer:** Python script reads local CSV files (pd.read_csv).
2. **Processing Layer (Pandas):**
   - Cleaning -> Merging -> Date Indexing.
   - Resampling (M) -> Imputation (FFill for L/R, 0-fill for F/M).
3. **Modeling Layer:**
   - tslearn processes 3D arrays (Customer × Time × Feature).
   - Prophet fits on aggregated cluster trends.
4. **Presentation Layer:** Streamlit renders the results dynamically based on user input.

12. Challenges and Solutions

Challenge 1: Handling Non-Purchasing Months

- **Problem:** When resampling time series, months with zero purchases resulted in NaN for Recency and Length. Standard 0-filling was incorrect (Recency is not 0 if you didn't buy).
- **Solution:** Implemented **Forward Fill (ffill)** logic for dates. If a customer bought in Jan and not in Feb, the "Last Purchase Date" for Feb remains Jan's date, correctly increasing

the Recency score.

Challenge 2: Computation Time of DTW

- **Problem:** Dynamic Time Warping has quadratic complexity ($O(N^2)$) and significantly slowed down the web app when running on 3,500+ customers.
- **Solution:** Implemented Streamlit's Caching mechanism (@st.cache_data). The clustering runs once per configuration and saves the result in memory, making subsequent interactions instant.

Challenge 3: Interpreting 4-Dimensional Data

- **Problem:** Explaining a 4D time series (L, R, F, M moving over time) to non-technical stakeholders is difficult.
- **Solution:** Simplified the visualization to a **Radar Chart** (aggregated average behavior) to show the "shape" of the customer, separating the technical complexity from the business insight.

13. Conclusion

This project successfully demonstrated that **Dynamic Segmentation** offers superior strategic value over static analysis. By tracking the *trajectory* of customer behavior, the business can now identify at-risk customers *before* they churn completely.

The integration of **Prophet forecasting** adds a forward-looking layer, allowing finance and inventory teams to anticipate demand fluctuations for specific customer groups. The final Streamlit application effectively bridges the gap between complex machine learning algorithms and accessible business intelligence.

Future Improvements:

- Integrate a live SQL database connection (e.g., PostgreSQL) for real-time analysis.
- Incorporate "Satisfaction" scores via Sentiment Analysis on customer support tickets.
- Implement A/B testing tracking to measure the actual ROI of the recommended marketing actions.

14. Appendix

A. Key Code Snippet: LRFM Calculation

```
# Monthly Resampling Logic
time_series_df = abt_indexed.groupby('customer_id').resample('M').agg(
    frequency=('customer_id', 'size'),
    monetary=('monetary_value', 'sum')
).reset_index()

# Recency Calculation (Days from end of month to last purchase)
# Uses forward fill to handle months with no purchases
ts_final['recency'] = (ts_final['transaction_date'].dt.to_period('M').dt.end_time -
ts_final['last_purchase_date']).dt.days
```