Customer Transactions Preprocessing Summary Report

Group Members and roles

Member	Role (task)
Glen Miracle	Part 1: Data Augmentation on CSV files
Peter Johnson	Part 2: Merging Datasets with Transitive properties
Nguepi Jordan	Part 3: Data consistency and Quality checks

Preprocessing Steps

The preprocessing of the `customer_transactions.csv` dataset involved several steps to clean, augment, and prepare the data for predicting customer_rating:

1. **Data Loading and Inspection**
    - Loaded the dataset (150 rows, 6 columns: customer_id_legacy, transaction_id, purchase_amount, purchase_date, product_category, customer_rating) using Pandas.
    - Inspected with head() and describe(), identifying 10 missing values in customer_rating.
2. **Data Cleaning**
    - Imputed missing customer_rating values with the mean (≈2.985) to preserve the distribution.
    - Converted `purchase_date` to datetime for temporal feature extraction.
3. **Data Augmentation**
    - Generated synthetic data by duplicating the dataset (150 → 300 rows).
    - Ensured unique transaction_id values by incrementing originals (e.g., 1150 → 1151+).

- Added noise to `purchase_amount` (±10%) and customer_rating (±0.1, clipped to [1, 5]).
- Sampled product_category from each customer's historical categories for realism.

4. **Feature Engineering**
   - Created temporal features: days_since_purchase (days from current date), purchase_month, purchase_day_of_week.
   - Normalized purchase_amount to a 0-100 scale (purchase_amount_normalized).
   - Computed moving averages (ma_3_purchases, ma_6_purchases) per customer.
   - Added behavioral features: `avg_monthly_purchases` (transactions/month), `avg_q2_spend` (Q2 average spend).
5. **Feature Selection**
   - Selected numeric features, dropped identifiers (customer_id_legacy, transaction_id).
   - Used SelectKBest with f_classif to pick the top 10 features for customer_rating: cluster, days_since_purchase, recency_weight, purchase_amount_normalized, ma_3_purchases, ma_6_purchases, purchase_month, purchase_day_of_week, avg_monthly_purchases, avg_q2_spend.
6. **Output**
   - Saved the augmented, transformed dataset (300 rows) as `final_dataset_ready.csv.`

Key Insights

- **Missing Data**: customer_rating had 6.67% missing values (10/150), addressed via mean imputation to retain all rows.
- **Spending Patterns**: purchase_amount ranged from 51 to 495 (mean ≈280.78), with synthetic data maintaining this distribution.

- **Feature Relevance**: Temporal (e.g., days_since_purchase) and behavioral (e.g., `avg_monthly_purchases`) features strongly correlated with customer_rating, suggesting predictive power.
- **Data Expansion**: Doubling the dataset improved robustness for modeling without overfitting risks.

Challenges and Solutions

- **Challenge: Limited Data Size**
  - *Solution*: Augmented with synthetic data, adding controlled noise and customer-specific category sampling to mimic real transactions.
- **Challenge: Missing Values in customer_rating**
  - *Solution*: Imputed with the mean to avoid dropping rows, given the low missing rate and numeric nature.
- **Challenge: Skewed purchase_amount**
  - *Solution*: Applied normalization (purchase_amount_normalized) to reduce skewness, aiding clustering and selection.
- **Challenge: Temporal Feature Accuracy**
  - *Solution*: Used the current date (March 16, 2025) for days_since_purchase, assuming recent data relevance; future adjustments could use a fixed reference date.
- **Challenge: Feature Overload**
  - *Solution*: Employed `SelectKBest` to focus on the top 10 predictive features, balancing complexity and utility.

Conclusion

The preprocessing transformed a small, partially incomplete dataset into a clean, augmented, and feature-rich version suitable for machine learning. The steps addressed data quality, quantity, and relevance, setting the stage for accurate customer_rating predictions.