

Summary of the Lookalike Model Logic:

The **Lookalike Model** identifies and recommends customers with similar profiles and transaction histories to a given input customer. Here's a breakdown of how the process works:

1. Input Handling:

- The model starts by accepting a **Customer ID** (e.g., C0001) as input.
- It then retrieves the profile and transaction summary of the input customer from the preprocessed dataset.

2. Data Preparation:

- **Customer Profile:**
 - Demographic details (e.g., Region) and aggregated transaction data (e.g., Total Spend, Product Diversity, Transaction Frequency) are collected.
- **Feature Normalization:**
 - Numerical features are normalized (scaled between 0 and 1) so that they contribute equally to similarity calculations.
 - Categorical features, like Region, are converted to numerical values using one-hot encoding.

3. Similarity Calculation:

- **Cosine Similarity** is used to measure how similar the input customer is to other customers based on their feature vectors.
- Cosine similarity calculates the cosine of the angle between two vectors in a multidimensional space. A score closer to 1 indicates higher similarity.
- The input customer's feature vector is compared to the feature vectors of all other customers in the dataset.

4. Recommendation:

- The model:
 - **Excludes** the input customer from being recommended to themselves.
 - **Sorts** all other customers based on their similarity scores in descending order.
 - **Selects** the top 3 customers with the highest similarity scores.

5. Output:

- The result is a list of the top 3 recommended customers based on similarity scores. The format looks like this:
 - Example: C0001: [(C0010, 0.89), (C0023, 0.86), (C0012, 0.84)]
 - This means:
 - **C0010** is the most similar customer to **C0001** with a similarity score of 0.89.
 - **C0023** and **C0012** are the next closest customers with similarity scores of 0.86 and 0.84, respectively.

6. How It Works Programmatically:

- For a given **Customer ID** input:
 1. **Extract the Feature Vector:** Retrieve the feature vector (profile and transaction data) for the customer.
 2. **Compute Similarity:** Use cosine_similarity from scikit-learn to calculate similarity scores with all other customers.
 3. **Sort and Select:** Sort customers by similarity and exclude the input customer from recommendations. Pick the top 3 most similar customers.

7. Real-World Usage:

This model can be used in various applications:

- **Personalized Marketing:** Recommend products that similar customers have purchased.
- **Customer Segmentation:** Identify groups of customers with similar profiles for targeted campaigns.
- **Customer Retention:** Suggest customers who are similar to high-value or loyal customers, which can be used to identify those at risk of churn or to match preferences.

Example Use Case:

Let's say the input customer is **C0001**, and the model finds that:

- **C0010** is the most similar customer with a high cosine similarity score of **0.89**.
- **C0023** follows with a similarity of **0.86**.
- **C0012** is the third most similar with a score of **0.84**.

Thus, the model would output:

C0001: [(C0010, 0.89), (C0023, 0.86), (C0012, 0.84)]

This is valuable for businesses that want to target customers who share similar behaviors or preferences to an existing user base.