Methodology

Data Source and Scope

The analysis uses the Instacart Online Grocery Shopping Dataset (2017), a publicly available dataset that captures anonymized online grocery purchase behavior. The dataset contains information on customer orders, purchased products, and product metadata. For the purposes of this project, three core files were used: orders, prior order–product interactions, and product descriptions. These files were sufficient to model user–product interactions required for recommendation system development.

Data Preparation and Integration

Data preparation focused on constructing a clean user–product interaction dataset. Order–product records were merged with order-level data to associate each purchased product with a unique user identifier. Product metadata were then joined to enable interpretation of recommendation outputs.
To improve computational efficiency and recommendation quality, the dataset was filtered to retain active users with a minimum number of purchases and products with sufficient interaction frequency. This step reduces sparsity and noise while preserving realistic purchase behavior patterns. All preprocessing steps were implemented using Python and standard data manipulation libraries.

Modeling Approach: Item-Based Collaborative Filtering

An item-based collaborative filtering approach was selected to build the recommendation system. This method recommends products based on similarity between items rather than direct similarity between users. The approach is well suited to retail purchase data, as it captures patterns of products that are frequently purchased together across users.
User–product interactions were modeled as implicit feedback, where a value of one indicates that a user has purchased a product and zero indicates no observed interaction. Instead of constructing a dense user–item matrix, interactions were represented using a sparse matrix structure to efficiently handle the high dimensionality of users and products. This design choice avoids unnecessary memory usage and reflects common industry practice when working with large-scale transactional datasets.

Similarity Computation and Recommendation Logic

Product–product similarity was computed using cosine similarity applied to the transposed sparse interaction matrix. This resulted in a similarity score between pairs of products based on shared purchasing patterns across users.
Recommendations for a given user were generated by aggregating similarity scores of products the user had previously purchased and ranking candidate products accordingly. Products already purchased by the user were excluded from the recommendation list. The final output consists of the top-N products with the highest aggregated similarity scores, along with their product names for interpretability.

Evaluation Strategy

Model performance was evaluated using a simple train–test split of user–product interactions. A portion of interactions was held out as test data, and recommendations were generated based on the remaining interactions. Precision at K was used as an evaluation metric, measuring the proportion of recommended products that appeared in the user's held-out purchase set. This evaluation approach provides an intuitive and widely used measure of recommendation relevance in implicit feedback settings.

Tools and Implementation

All data preprocessing, modeling, and evaluation were conducted in Python using common data science libraries, including pandas, NumPy, scikit-learn, and SciPy. The methodology emphasizes clarity, efficiency, and reproducibility, making the project suitable for an entry-level analytics portfolio while remaining grounded in established recommendation system techniques.