Technical Report: E-commerce Recommendation Engine Project: Improving Sales Through Data Science

Introduction

The e-commerce industry is extremely competitive, with companies continuously looking for methods to improve customer experience and increase revenue. Underperforming products are a prevalent challenge that can impede overall revenue growth. This paper describes the creation and implementation of a personalised product recommendation engine intended to increase sales for an e-commerce company. This report will use the STAR (Situation, Task, Action, Result) method to define the project's scenario, goals, technical approach, and outcomes.

Scenario

The e-commerce company was having problems with certain products not performing as planned in terms of sales. Despite having a varied product catalogue, a large chunk of the inventory was underutilised. To remedy this issue, the company planned to use data science to provide consumers with personalised product recommendations based on their browsing history, purchase history, and other pertinent information.

Project Goal

The major goal was to create a recommendation engine that could offer appropriate products to customers, boosting the likelihood of purchases and overall sales performance. The engine needs to be smoothly integrated into the company's existing e-commerce platform, with real-time recommendations available on both the website and the mobile app.

Our Task

We oversaw the recommendation engine's development and deployment. This included developing the technological approach, supervising data collecting and preprocessing, selecting appropriate recommendation approaches, training, and evaluating models, and ensuring effective system implementation and monitoring. The goal was to build a strong and efficient recommendation engine that could manage massive amounts of data and provide personalised product recommendations in real time. To accommodate varied user behaviours and preferences, the engine needed to mix a variety of recommendation algorithms. Furthermore, it was critical to constantly check the engine's performance and retrain models as necessary to react to changing user behaviours and product trends.

Technical Approach

The project was carried out in numerous crucial phases, each requiring unique measures to attain the desired result.

1. Data collection and preprocessing.

- User Data Sources: The initial step was to collect information from a variety of sources, including users' purchase histories, browsing behaviour (such as seen goods and time spent on product pages), and product features (such as category, brand, and price).
- Methods: Data were gathered from user interaction logs, transaction databases, and product catalogues.

Data preprocessing:

- Cleaning: The data was cleaned to eliminate duplicates, manage missing values, and ensure format consistency. This was critical to ensuring data quality and reliability.
- Transformation: Product categories and brands were encoded, while numerical data like price were normalised. This change was necessary to get the data ready for analysis and model training.

2. Recommendation Techniques

Hybrid Recommendation Engine:

To guarantee the recommendation engine was comprehensive and accurate, a hybrid approach was taken, incorporating different recommendation techniques:

Collaborative Filtering

- Method: employed K-Nearest Neighbours (KNN) with Z-score normalisation to find commonalities in user behaviour.
- Rationale: This technique promotes products based on the assumption that customers with similar behaviours (e.g., those who purchased product X and also purchased product Y) are likely to share preferences.

Content-based Filtering:

- Method: Product attributes were used to recommend things that were similar to those that the user had previously seen or purchased.
- Rationale: By focusing on product criteria such as category, brand, and price, this technique suggests products that are similar to the user's previous preferences.

Matrix Factorization:

- Method: Non-negative matrix factorization (NMF) and singular value decomposition (SVD) were used to reduce data dimensionality while retaining user-product linkages.
- Rationale: These strategies improve suggestion accuracy by identifying latent factors influencing user-product interactions.

3. Model Training and Evaluation.

- Datasets: Historical data was separated into training and test sets to make model training and evaluation easier.
- Algorithms: Python's Surprise module was used to implement the selected algorithms (KNN with Z-Score, SVD, and NMF).

About the Dataset

This Kaggle dataset, collected from Amazon India, contains 1465 entries with product reviews. It contains detailed information such as product names, categories, pricing details, ratings, user reviews, and metadata. The dataset contains valuable insights into customer preferences, product performance, and user interactions, making it appropriate for sentiment analysis, recommendation system creation, and analysing consumer behaviour in the e-commerce sector.

The 16 Columns

- product_id: represents a special identification code for every product. Usually, a series of alphanumeric characters is used to differentiate one product from another.
- product_name: includes the product's name or title. This aids in recognising and characterising the product.
- category: shows the category that the product is in. Among the categories could be apparel, appliances for the house, electronics, etc.
- discount_price: represents the product's price after any applicable discounts. This column of numbers shows the selling price as of right now.
- actual_price: represents the product's initial cost, before any discounts. This numerical column also displays the price in its entirety.
- discount_percentage: displays the product's % discount. The difference between the reduced price and the actual price is used to compute this.
- rating_count: shows the total amount of reviews or ratings that the product has gotten. This shows the number of users who have left reviews for the product.
- about_product: includes information on the product, such as a description. Features, specifications, and other pertinent data may be included in this.
- user_id: is a special number that corresponds to every person who has given the product a rating or review. This facilitates tracking actions specific to a user.
- user_name: includes the user's name or username who submitted the review or rating. This can be applied to user behaviour analysis or interaction personalisation.
- review_id: serves as a special identification for every review. This makes reviews easier to discern from one another, even when they are about the same product.
- review_title: includes the review's title that the user submitted. This is frequently a succinct synopsis of the review's contents.
- review_content: includes the entire text of the user-provided review. This can offer comprehensive insights into the views and experiences of users.
- img_link: includes a URL that leads to a picture of the product. This aids in the product's visual identification.
- product_link: includes the URL that leads to the product page on the internet. This enables customers to see additional information or make a purchase.

Evaluation metrics

To assess the performance of the recommendation models, the following metrics were utilised:

- Root Mean Squared Error (RMSE): This metric computes the average magnitude of errors between expected and actual ratings.
- Mean Absolute Error (MAE): This metric determines the average absolute difference between projected and actual ratings.

- Performance: To ensure robustness and reliability, models were cross-validated fivefold.

Evaluation Report

We selected four different algorithms: KNNWithZScore, SVD (Singular Value Decomposition), NMF (Non-Negative Matrix Factorization), and a second iteration of SVD. These algorithms were chosen for their ability to capture user-item interactions and produce accurate suggestions. After picking the models, we trained them on the dataset and assessed their performance using cross-validation techniques.

The evaluation process offered information about each algorithm's prediction ability. The following are the evaluation metrics for each algorithm:

KNNWithZScore:

- Average RMSE: 0.2495
- Average MAE: 0.1816
- Top 10 recommendations for a specific user: [('B07JW9H4J1', 4.1098), ('B098NS6PVG', 1098), ..., ('B08CF3D7QR', 4.1098)]

SVD (First iteration):

- Average RMSE: 0.2520
- Average MAE: 0.1917
- Top 10 recommendations for a specific user: [('B07XLCFSSN', 4.1772), ('B0B9BXKBC7', 1756), ..., ('B09MT84WV5', 4.1571)]

NMF:

- Average RMSE: 0.2699
- Average MAE: 0.2071
- Top 10 recommendations for a specific user: [('B07JW9H4J1', 4.1075), ('B098NS6PVG', 1075), ..., ('B08CF3D7QR', 4.1075)]

SVD (Second iteration):

- Average RMSE: 0.2529
- Average MAE: 0.1920
- Top 10 recommendations for a specific user: [('B09C6HXFC1', 4.1840), ('B0B9BXKBC7', 1659), ..., ('B0B23LW7NV', 4.1537)]

Report on Recommendation System Development.

In order to increase sales across a whole product catalogue, we created a personalised product suggestion engine using data science approaches. Our trip began with data collection, which involved retrieving essential information from a CSV file named 'data.csv'. This dataset included important elements such user interactions, product attributes, and ratings.

Following data collecting, we focused on data preprocessing to verify data integrity and homogeneity. This stage was expedited because of the dataset's cleanliness, which eliminated the need for expensive cleaning procedures. We then proceeded to model selection, using

collaborative filtering with KNNWithZScore, SVD, and NMF algorithms. These algorithms were chosen for their ability to capture user-item interactions and produce accurate suggestions.

After picking the models, we began model development, dividing the data into training and testing sets. We next trained the models using the training data to discover the underlying patterns in user-item interactions. This step established the groundwork for making predictions and assessing model performance. The next phase was model evaluation, in which we used cross-validation techniques to examine the models' predictive ability. Metrics like RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) were employed to assess the models' accuracy. In addition, we developed top recommendations for a single user to evaluate the models' ability to suggest appropriate products. To improve model performance, hyperparameter tuning was performed on the top-performing algorithms, KNNWithZScore and SVD. This technique entailed systematically varying algorithm settings to determine which combinations produced the greatest results. The best-performing combinations were chosen based on their RMSE values. After attaining sufficient performance metrics, we ran the models across the full dataset to generate complete recommendations for all users. This marked the end of our recommendation system development journey, giving stakeholders a powerful tool to increase sales and improve user experience.

In summary, our efforts included data collecting, preprocessing, model selection, construction, evaluation, and optimisation. By combining cutting-edge algorithms and rigorous evaluation procedures, we created a powerful recommendation engine capable of providing personalised product recommendations, allowing the e-commerce company to prosper in a competitive market scenario.