# Products Recommendation

WÜRTH ITALIA – Capstone Projects

Gyeongwon Song

September 20, 2024

## 1 Introduction

This project aims to build a recommendation system to predict customer preferences and provide relevant product suggestions using different methods: User-Based Collaborative Filtering, Clustering, and Neural Collaborative Filtering (NCF). The dataset includes detailed customer actions and product information but does not have explicit ratings, only action types and counts, which makes building a recommendation system more challenging.

To solve this, we applied multiple algorithms. The Neural Collaborative Filtering (NCF) model performed the best, using log loss to measure how well the model fits the data. As the log loss decreases over time, it shows that the model is learning effectively and may give the most accurate recommendations.

## 2 Data Sources

The dataset provides insights into customer behavior with respect to a product (level5). Actions include both online (e.g., click, search, download) and offline (e.g., purchase) interactions. Key fields in the dataset are as follows:

- **Customer Code**: customerid
- **Action Type**: actions_type
  - purchase
  - ecommerceAbandonedCart
  - search
  - action
  - download
  - no_data
- **Date**: dat_action
- **Actions Count**: actions_count
- **Market Segment**: att_marketsegment
- **Sales Channel**: att_saleschannel
- **Market Sector**: att_sector
- **Product Hierarchy Levels**: level1id to level5id

## 3 Data Preparation

### 3.1 Handling Missing Values

- check for missing values in the dataset and find that the `dat_action` column contains missing entries.
- Remove rows with missing values from the dataset to ensure data quality.

### 3.2 Feature Engineering

- **Removing Irrelevant Actions**: Actions such as `ecommerceAbandonedCart` do not directly contribute to product recommendations.
- **Converting Data Types**: The action dates (`dat_action`) are converted to a datetime format to facilitate time-based analysis. Additionally, categorical columns like `actions_type` and `att_sector` are converted to categorical types for efficiency.

- **Handling Duplicates**: Duplicate rows are removed to avoid skewing the analysis.
- **Creating New Features**: A new column, `action_month`, is created to capture the month of each customer action. This helps in analyzing seasonal trends in customer behavior.

# 4 Solution Description

## 4.1 User-Based Collaborative Filtering (CF)

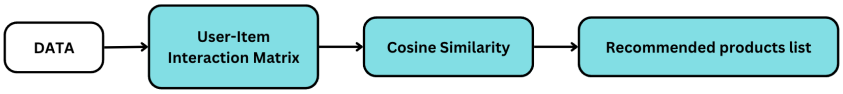


Figure 1: Architecture of User-Based CF

User-Based CF recommends products to a user similar to a target user based on their historical interactions (e.g., product purchases, views) and recommends items that similar users have interacted with.

- **User-Item Interaction Matrix Creation**: Construct a user-item interaction matrix, where each row represents a unique customer and each column corresponds to a product (level5id). The values in the matrix indicate the total actions count (e.g., purchases) for each customer-product pair.
- **Cosine Similarity**: Using the cosine similarity metric, compute similarities between users based on their interaction patterns. This score helps identify users with similar preferences.
- **Product Recommendations**: For a given user, identify similar users and aggregate their interactions with products that the target user has not yet engaged with. The top recommended products were then suggested based on the aggregated actions count from similar users.

## 4.2 Clustering Algorithm

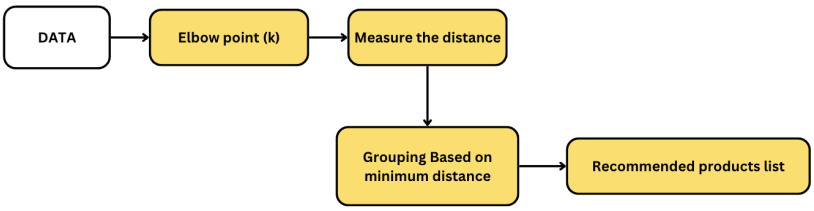


Figure 2: Architecture of Clustering Algorithm

Clustering is a technique used to group similar data points together based on their features. K-Means clustering was implemented to segment customers based on their behavior and interaction with products.

- **Feature Selection**: To perform clustering, relevant features were selected from the dataset, including:
  - **Actions Count**: The total number of actions performed by each customer.
  - **Frequency of Actions**: How often each customer engages with products over a defined time period.
- **K-Means Clustering**: Applied the K-Means algorithm to segment customers into distinct clusters:
  - **Choosing the Number of Clusters (K)**: Methods such as the Elbow Method and Silhouette Analysis were employed to determine the optimal number of clusters.

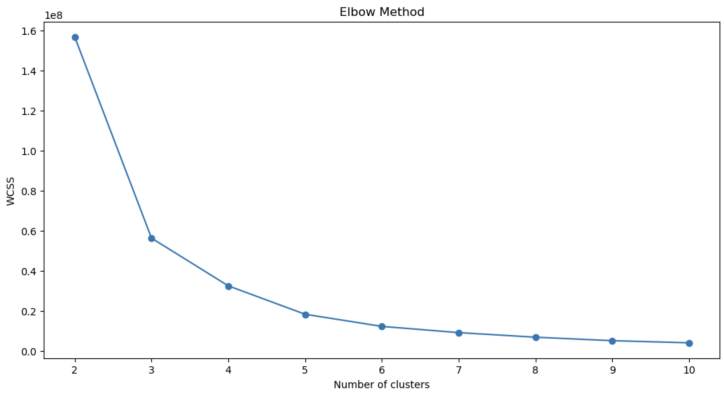


Figure 3: K-Means Clustering Elbow Method

  - **Clustering Process**: The K-Means algorithm iteratively assigned customers to the nearest cluster centroid and updated the centroids based on the mean of the assigned points until convergence was achieved.

- **Apply Clustering**: Use K-Means clustering to group users and recommend popular items in each cluster.

## 4.3 Neural Collaborative Filtering (NCF)

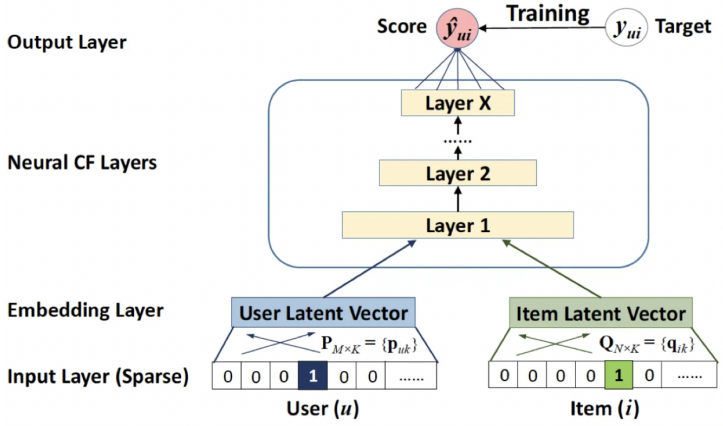


Figure 4: Architecture of NCF

Neural Collaborative Filtering (NCF) is an advanced approach to recommendation systems that leverages deep learning techniques to model complex user-item interactions.

- **Embedding Laye**

  - **User and Item Embeddings**: Embedding layers were created for both users and items to convert categorical variables (user IDs and item IDs) into dense vectors.

- **Multi-Layer Perceptron (MLP)**: After obtaining the embeddings, they were fed into a multi-layer perceptron (MLP) architecture. The MLP consists of several fully connected layers with activation functions (e.g., ReLU) to capture complex interactions between user and item embeddings.

- **Output Layer**: Use log loss and ranking metrics to evaluate the model.

- **Make Predictions and Recommendations**: The output layer used a sigmoid or softmax activation function, depending on whether binary interactions (e.g., purchase or not) were being predicted.

# 5 Evaluation Results

- **Splitting the Data**: The data is split into two parts: **past data** and **future data**. The past data is used to train the recommendation models, while the future data is reserved for validation and testing. The split is done in chronological order to mimic a real-world scenario where recommendations are based on past behavior.
  - **User-Based CF**: This approach identified only 5 out of 50 relevant items. The primary limitation here is that user-based collaborative filtering depends heavily on the presence of user interactions and similarities between users.
  - **Clustering**: The clustering algorithm identified 15 out of 50 relevant items, proving to be more effective in this context. However, it still has limitations. Clustering methods rely on predefined features and might not capture nuanced user preferences or changes in behavior over time.
- **NCF**: The performance of the NCF model was evaluated using metrics:
  - **Log Loss**: This metric measured the model's performance on the binary classification task, with lower values indicating better predictions.
  - **Training and Validation Metrics**: Training loss and evaluation log loss were tracked across epochs to assess overfitting and generalization.

Table 1: Results and Effectiveness

| **Metric** | **Value** |
|---|---|
| Initial Training Loss | 0.6229 |
| Final Training Loss (Epoch 10) | 0.4838 |
| Initial Eval Log Loss | 0.6886 |
| Final Eval Log Loss (Epoch 10) | 0.4721 |

The decrease in log loss from the first to the last epoch shows that the model improved in predicting how users interact with items. Specifically, the evaluation log loss dropped from 0.6886 to 0.4721, indicating that the model's predictions became more accurate as it trained.

In the final evaluation on the test dataset, the model achieved a test loss of 0.4704, further confirming that the NCF model effectively captures user preferences and item similarities.

# 6 Conclusion

The Products Recommendation project effectively built a recommendation system using User-Based Collaborative Filtering, Clustering, and Neural Collaborative Filtering (NCF). While NCF demonstrated the best performance, the project identified several limitations in each approach.

- **User-Based Collaborative Filtering**: This method relied too much on user interactions, identifying only 5 out of 50 relevant items. To improve this, we can add more user data and use hybrid models that combine collaborative and content-based filtering.
- **Clustering**: This approach identified 15 relevant items but depended on predefined features, which might miss changes in user behavior. Adapting clustering methods and incorporating real-time data could help address this issue.

Expanding the dataset to include user demographics and product attributes can provide better insights. Additionally, exploring advanced algorithms like reinforcement learning could further enhance recommendation accuracy.