# A Data Mining Approach for Predicting Customer Repurchase Patterns in E-Commerce

Ulus Emir Aslan
*Institute of Computer and Informatics*
*Istanbul Technical University*
Istanbul, Turkiye
150210320
aslanu21@itu.edu.tr

Kaan Yücel
*Institute of Computer and Informatics*
*Istanbul Technical University*
Istanbul, Turkey
150210318
yucelk21@itu.edu.tr

***Abstract*—Predicting customer-product interactions is critical for personalized recommendations and inventory management. This study investigates the problem of predicting whether a customer will purchase a specific product and when within a month using a dataset containing detailed customer-product interactions, product attributes, and transaction histories. We employ machine learning techniques to leverage features derived from product clusters, transaction patterns, and customer behaviors.**

***Index Terms*—Machine learning, prediction, recommendation, cluster, accuracy, customer behavior**

## I. Introduction

Accurately predicting customer purchasing behavior is a vital challenge in e-commerce industries. Personalized recommendations and inventory optimization based on these predictions can significantly enhance customer satisfaction and profitability. The primary objective of this study is to predict whether a customer will purchase a specific product within the next four weeks. If a purchase is made, we aim to determine the exact week in which it occurs. If no purchase is made, the wanted output is 0.

To achieve this goal, we analyze several datasets:

- **Transaction Dataset:** Captures customer transactions, detailing the purchased products, quantities and dates.
- **Product Catalog Dataset:** Includes product-specific attributes such as manufacturer ID and encoded features.
- **Catalog Map Dataset:** Describes the hierarchical relationships between product categories.
- **Test Dataset:** Contains customer-product pairs for which predictions are to be made.

By integrating these datasets, this project leverages machine learning techniques and impactful feature engineering to address the challenge of predicting customer-product interactions.

### A. Summary of the project

Understanding customer purchasing behavior is crucial in e-commerce industries, where personalized recommendations and inventory optimization can significantly enhance customer satisfaction and profitability. Predicting whether a customer will purchase a specific product is a complex task that requires analyzing historical transaction data, customer attributes, and product characteristics. This project addresses the challenge of predicting customer-product interactions by leveraging machine learning techniques and impactful feature engineering.

### B. Dataset

The dataset used in this study comprises detailed information about transactions, products, and customers. Key features include transaction history, product attributes such as manufacturer ID and various encoded attributes. Some of these attributes' meanings were not provided, so they were treated as generic encodings. Additionally, a test set was given to evaluate the model as part of a competition.

### C. Project's achievements

This work contributes to the field by emphasizing the role of data preprocessing and feature engineering in constructing a robust predictive model. By clustering products based on their categories and aggregating customer-level purchase behaviors, we create meaningful features that improve prediction accuracy. Several machine learning models are evaluated to identify the best-performing technique, and their results are discussed comprehensively.

### D. Structure of paper

The remainder of this paper is organized as follows: Section II provides a literature review highlighting related works and their advantages and limitations. Section III explains the data preprocessing and feature engineering methods employed. Section IV describes the predictive models and techniques used in this study. Section V presents experimental results and discusses their implications. Finally, Section VI concludes the paper and outlines potential areas for future research.

## II. Literature Review on Customer-Product Pair Prediction

Customer-product pair prediction aims to predict the likelihood of a specific customer purchasing a particular product. This task is central to many applications, including sales forecasting, inventory management, and personalized marketing. In this literature review, we discuss various methods employed

in customer-product pair prediction, the challenges these methods face, their advantages, and areas where improvements can be made.

### A. Overview of Customer-Product Pair Prediction

Customer-product pair prediction involves identifying the likelihood that a customer will purchase a specific product, often based on historical purchase data, customer demographics, and product characteristics. This task can be framed as a binary classification problem (predicting whether a purchase will occur or not) or a multi-class regression problem (predicting the quantity or time of purchase). Several techniques have been employed to address this problem, ranging from traditional machine learning models to advanced deep learning approaches.

### B. Traditional Methods in Customer-Product Pair Prediction

- **Collaborative Filtering (CF)**: While collaborative filtering is primarily used in recommendation systems, it has also been applied to customer-product pair prediction. CF models predict customer-product interactions based on similar customers or products. This approach benefits from its simplicity but struggles with the *cold-start problem*, where new customers or products have insufficient interaction data [1].
- **Logistic Regression and Classification Models**: Logistic regression has been widely used for predicting the likelihood of customer-product interactions. By modeling the probability of purchase based on customer demographics, historical behavior, and product attributes, logistic regression models provide interpretable results. However, they often struggle to capture non-linear relationships and complex interactions between features [2].
- **Decision Trees and Random Forests**: Decision trees and random forests are popular machine learning techniques for predicting customer-product pairs. These models work well when the feature set is large and categorical, and they provide high interpretability. However, decision trees may not generalize well when the data is sparse, which can be an issue in customer-product prediction tasks [3].

*a) Advantages:*

- **CF:** Simple to implement and interpretable.
- **Logistic Regression:** Provides interpretable results.
- **Decision Trees/Random Forests:** High interpretability and handle large categorical data well.

*b) Disadvantages:*

- **CF:** Struggles with cold-start problem.
- **Logistic Regression:** Can't capture non-linear relationships well.
- **Decision Trees/Random Forests:** May not generalize well with sparse data.

### C. Matrix Factorization

Matrix factorization techniques have been extensively used for customer-product pair prediction. These methods factorize the sparse customer-product interaction matrix into low-dimensional latent factors, capturing complex patterns of interactions between customers and products.

- **Singular Value Decomposition (SVD)**: SVD is one of the most common matrix factorization techniques used in collaborative filtering and customer-product prediction. It is effective at capturing latent structures in the data and can predict missing entries in the interaction matrix. However, SVD assumes linear relationships between latent factors, which may limit its performance when the data contains non-linear patterns [4].
- **Factorization Machines (FM)**: Factorization machines are a generalization of matrix factorization and can model interactions between arbitrary feature pairs. This makes them particularly useful for customer-product pair prediction, where interactions between customer and product attributes need to be captured. Factorization machines are flexible and can be applied to a wide range of data types. However, they require careful tuning and are computationally expensive [5].

*a) Advantages:*

- **SVD:** Effective at capturing latent structures and predicting missing data.
- **FM:** Flexible, models interactions between arbitrary feature pairs.

*b) Disadvantages:*

- **SVD:** Assumes linear relationships, limiting its ability to capture non-linear patterns.
- **FM:** Requires careful tuning and is computationally expensive.

### D. Deep Learning Methods in Customer-Product Pair Prediction

Deep learning techniques have gained significant attention in the field of customer-product pair prediction due to their ability to capture complex, non-linear relationships within large datasets.

- **Neural Collaborative Filtering (NCF)**: NCF is a deep learning-based approach that combines neural networks with collaborative filtering to model the interaction between customers and products. The model learns a non-linear interaction function between customer and product embeddings. NCF has been shown to outperform traditional matrix factorization methods, especially in cases with sparse data. However, deep learning models like NCF are prone to overfitting, especially when training data is limited [6].
- **Convolutional Neural Networks (CNNs)**: CNNs have been applied to predict customer-product interactions by treating the prediction task as a sequence or image-like structure. They excel in cases where there is a spatial structure or temporal sequence in the customer-product interactions. However, CNNs can be computationally intensive, especially when dealing with large-scale datasets [7].

- **Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM)**: RNNs and LSTMs are used for sequence prediction tasks, such as predicting the time or sequence of purchases. These models are particularly useful when the order of purchases matters, for example, in predicting the likelihood of repurchase over time. The main drawback of RNNs and LSTMs is that they require significant amounts of training data and computational power [8].

  *a) Advantages:*
- **NCF:** Captures non-linear relationships, outperforms traditional methods in sparse data.
- **CNNs:** Good for structured, sequential data, learns rich features.
- **RNNs/LSTMs:** Effective for modeling temporal dependencies in purchase patterns.

  *b) Disadvantages:*
- **NCF:** Prone to overfitting, requires large datasets.
- **CNNs:** Computationally expensive, not ideal for sparse data.
- **RNNs/LSTMs:** Need large datasets and computational resources, training can be complex.

### *E. Future Directions*

The future of customer-product pair prediction lies in addressing these challenges and improving model performance through the following approaches:

- **Hybrid Models**: Combining collaborative filtering, content-based filtering, and deep learning techniques can help overcome the limitations of individual methods, improving accuracy and reducing issues such as data sparsity and cold-start problems.
- **Context-Aware Predictions**: Incorporating contextual information, such as time, location, and user behavior, can improve the accuracy of customer-product predictions by making them more relevant to the current circumstances.
- **Collaborative Filtering (CF)**: Collaborative filtering algorithms predict a user's interests by collecting preferences or taste information from many users. The assumption is that if two users agree on one issue, they are likely to agree on others as well. Collaborative filtering is classified into:
  - *User-based Collaborative Filtering*: Recommends items liked by similar users.
  - *Item-based Collaborative Filtering*: Recommends items similar to items a user has liked in the past.

  The method's simplicity and effectiveness in capturing user preferences make it widely used. However, collaborative filtering suffers from *cold-start* problems, where it struggles with recommending items to new users or predicting for items with little user interaction [**?**].
- **Content-Based Filtering (CBF)**: Content-based recommendation systems recommend items similar to those the user has interacted with, based on attributes of the items themselves. For example, in a movie recommendation system, if a user liked action movies in the past, content-based systems will recommend other action movies. These systems are less prone to the cold-start problem because they rely on item features [**?**].
- **Hybrid Methods**: Hybrid methods combine multiple recommendation strategies, such as collaborative filtering and content-based filtering, to overcome the limitations of individual methods. These systems attempt to provide more accurate and diverse recommendations by blending various approaches [**?**].

## III. Preprocessing and Feature Engineering

### *A. Handling null values*

In categorical attributes of the product dataframe there are -1 values. We have considered them as NULL values and fill it with that columns mode value.However, attribute_5 value has the mode -1 so we did not operate on -1 values in attribute_5.

There are also NULL values or empty lists at category column in product dataframe. We will this categories with the mode of the group that have the same attribute 1,2 and 3. We assign the NULL category that group's category mode. We did this because separating the rows by the first 3 attributes yields decent and balanced data.

### *B. Feature Engineering on Category column*

There are many unique values in categories and so category column. We assign every category in every row of product, to a parent category that is the top second one as in hierarchical order.We do this by implementing 2 functions.First one is getting the top second parents and modifying the category-parent_category column.Second one is applied to every row the product dataframe's category column as it is transforming the category list to a new refined list.This way we got more pure and easy to use category feature.

### *C. Data Processing*

To prepare the data for prediction, we performed the following preprocessing steps:

- **Joining Datasets:** The transaction dataset was joined with the product catalog dataset to incorporate product-level attributes such as manufacturer ID and category information. This allowed us to leverage additional features during feature engineering and modeling.
- **Categories Column:**
- **Target Variable Creation:** A sliding window approach was applied to calculate the day differences between customer transactions within the same cluster. Based on these day differences, we created the target variable as follows:
  - Assign values **1, 2, 3, or 4** for transactions made within the first, second, third, or fourth weeks (i.e., day differences of 1-7, 8-14, 15-21, or 22-28 days).
  - Assign a value of **0** for transactions with day differences greater than 28 days.

- **Same-Day Transactions:** Transactions made on the same day were treated as a single transaction. No day difference was calculated between these transactions, ensuring consistency in the target variable assignment.

### D. Association Rule Mining

To analyze the relationships between products, we employed association rule mining. Customer-level transaction data was grouped to form individual markets, and frequent itemsets were mined using the Apriori algorithm. This method revealed co-purchase patterns and dependencies between items, helping to identify potential cross-selling opportunities. Metrics such as support, confidence, and lift were used to evaluate the derived rules.

These steps enabled the creation of a structured dataset that incorporated temporal purchasing patterns, facilitating better feature engineering and model training.

### E. Clustering and Feature Engineering

Recent studies have emphasized the role of clustering and feature engineering in enhancing predictive performance. In this work, k-means clustering was applied to group products based on their attributes. Specifically, we calculated the day differences between customer repurchases within clusters to uncover temporal patterns in buying similar products. We utilized DBSCAN on the categories column to identify density-based groupings and capture subtle patterns in categorical data. These approaches provide a robust framework for understanding complex relationships within the data. However, the effectiveness of these methods depends on careful parameter tuning, such as selecting the appropriate number of clusters for k-means or the epsilon value for DBSCAN.

Additionally, we created a popularity score attribute for products to measure their overall demand. Two additional attributes were developed for each customer-cluster pair to track the maximum and minimum time that a customer did not purchase a product belonging to that cluster. These features provided deeper insights into purchasing patterns and temporal behavior.

## IV. Models and Techniques

### A. Random Forest

We utilized the Random Forest algorithm, which is very good for tabular data, for predicting customer-product interactions. Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive performance and reduce overfitting. It operates by constructing a multitude of decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees.

Each decision tree is trained on a bootstrap sample of the dataset, and at each split, only a random subset of features is considered. This technique, known as bagging (Bootstrap Aggregating), enhances model robustness and reduces the correlation between individual trees.

The Random Forest model was implemented using the following parameters:

- **n_estimators=100:** Specifies the number of decision trees in the forest. We chose 100 to ensure sufficient model complexity while maintaining computational efficiency.
- **max_depth=12:** Limits the maximum depth of each tree. This constraint helps prevent overfitting by restricting the model's ability to memorize the training data.
- **min_samples_split=10:** Sets the minimum number of samples required to split an internal node. By increasing this value, we reduce the risk of overfitting, as splits are made only when sufficient data supports them.
- **max_features=5:** Determines the number of features considered for splitting at each node. Limiting this value encourages diversity among trees and reduces the likelihood of overfitting.

These parameters were selected based on a combination of domain knowledge and empirical testing to balance model complexity, generalization, and computational cost. The Random Forest model provided a robust framework for capturing complex relationships within the data and delivered competitive predictive performance.

### B. Other models

We have also implemented various machine learning models such as CatBoost classifier, XG boost and more, and used ensembling with the combinations of those models. However, we found that best accuracy is achieved by RandomForest. Therefore, we only use RandomForest model.

## V. Results and Discussion

### A. Prediction Performance

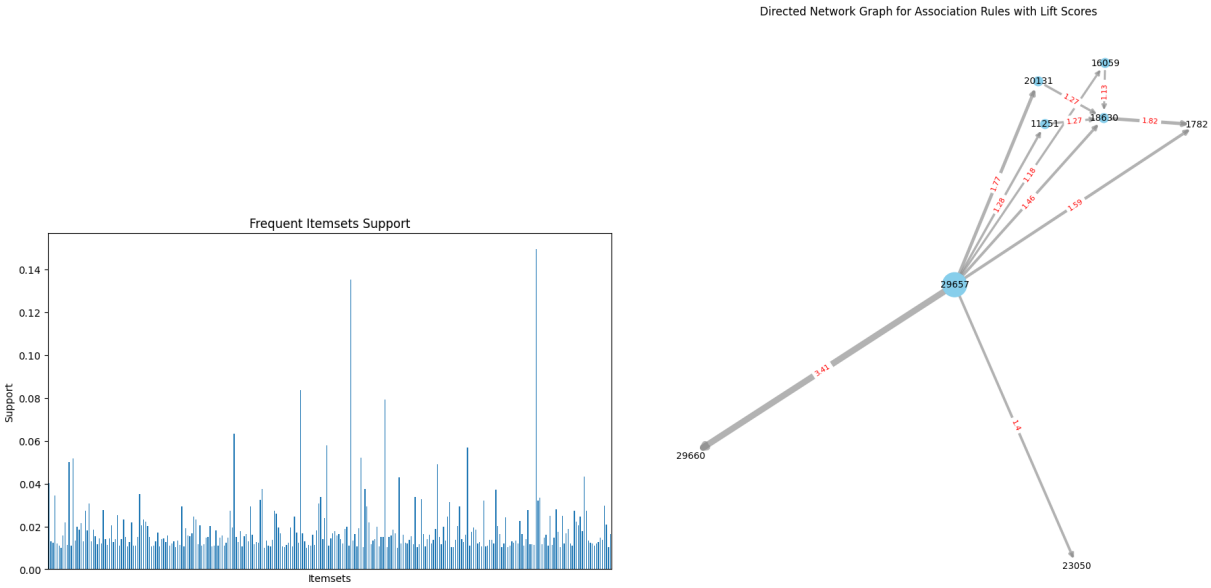
Our predictive model was evaluated on a test set comprising 152,000 samples. The model achieved an accuracy of 0.57, indicating moderate performance in predicting customer-product interactions. While the accuracy highlights room for improvement, the results demonstrate the potential of feature engineering and clustering techniques in enhancing predictive tasks.

### B. Classification Report

The performance of the model is summarized in the classification report (Figure 2), which provides insights into precision, recall, and F1-score for different classes.

### C. Association Rules

The association rule mining revealed several interesting patterns, such as strong dependencies between frequently purchased items. For example, customers who purchased item A often also purchased item B with high confidence. These insights are valuable for cross-selling strategies and inventory optimization.




(a) Itemsets with higher support value than 0.01

(b) Directed Network Graph for High Lift Value Pairs

Fig. 1: Visualization of association rule mining results. (a) The support values of frequent itemsets, providing insights into their relative occurrences. (b) A directed network graph illustrating high-lift pairs, highlighting strong relationships between items in the dataset.

## VI. Conclusion

### A. Summary

This study explored the prediction of customer-product interactions using machine learning techniques, specifically Random Forest, and impactful feature engineering. By using datasets containing customer transaction histories, product attributes, and category hierarchies, we were able to extract meaningful patterns and insights in our test set. The inclusion of features such as product popularity scores and temporal purchase behavior enhanced the model's predictive capability. The results show the importance of feature engineering and ensemble methods in complex predictive tasks.

### B. Future Work

Future directions include:

- Incorporating advanced machine learning architectures, such as Transformer-based models, to improve prediction performance.
- Developing methods to handle data imbalance more effectively, which could lead to more accurate and fair predictions.
- Expanding the feature set to include additional contextual information, such as seasonal trends or promotional activities which are created based on the date of the transactions, to provide more information to the model.
- Optimizing hyperparameters further and exploring alternative ensemble techniques to enhance model robustness and generalization.

## Acknowledgment

Thank you Şule Gündüz Öğüdücü.

## References

[1] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, pp. 1-19, 2009.

[2] S. Lee, S. Lee, and M. Kim, "Prediction of customer-product pair in e-commerce," *Journal of Electronic Commerce Research*, vol. 12, no. 2, pp. 123-135, 2011.

[3] Y. Qu, Z. Zha, and Q. Zhang, "Recommendation system based on decision trees for customer-product prediction," *Journal of Computer Science and Technology*, vol. 24, no. 6, pp. 1012-1022, 2009.

[4] Y. Koren, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30-37, 2009.

[5] S. Rendle, "Factorization machines," in *Proceedings of the 10th IEEE International Conference on Data Mining*, pp. 995-1000, 2012.

[6] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 173-182, 2017.

[7] X. Li, Y. Yang, D. Wu, and H. Cheng, "Recommendation based on convolutional neural networks for customer-product prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4582-4592, 2018.

[8] J. Tang, X. Wang, and S. Wang, "Recommendation for customer-product pairs using recurrent neural networks," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 2078-2084, 2019.

[9] H. Chen, L. Zhang, and L. Chen, "A recommender system for e-commerce based on customer-product interaction prediction," *Journal of Computer Science and Technology*, vol. 32, no. 3, pp. 501-513, 2017.

[10] X. Zhang, X. Chen, and L. Zhang, "A survey on personalized recommendation and customer-product prediction," *IEEE Access*, vol. 8, pp. 114915-114930, 2020.

[11] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2010.

Accuracy: 0.5730
Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.47 | 0.64 | 45381 |
| 1 | 0.42 | 0.99 | 0.59 | 29663 |
| 2 | 0.53 | 0.60 | 0.56 | 27905 |
| 3 | 0.64 | 0.44 | 0.52 | 26488 |
| 4 | 0.72 | 0.36 | 0.48 | 22729 |
| accuracy | | | 0.57 | 152166 |
| macro avg | 0.66 | 0.57 | 0.56 | 152166 |
| weighted avg | 0.70 | 0.57 | 0.57 | 152166 |

Fig. 2: Classification Report for Test Set