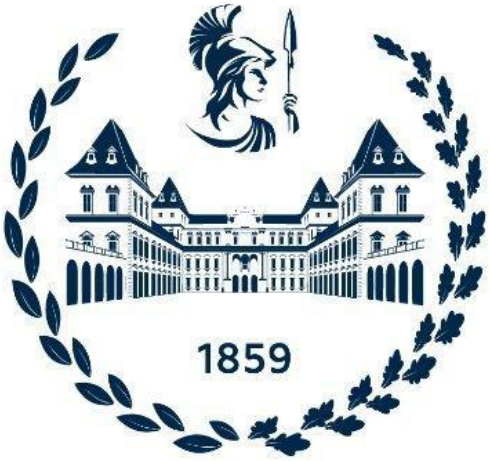
POLITECNICO DI TORINO

Collegio di Ingegneria Gestionale e della Produzione
Corso di Laurea Magistrale in Management Engineering



AI Module for Customer Preferences System: Product-Level Recommendation Classification with LP-based UTADIS and Machine Learning Benchmarks

Supervisor
Prof. Guido Perboli

Candidate
Aigerim Kabyl

A.A. 2025/2026

Table of Contents

Abstract 4

1. Introduction 5

2. Background: Customer Preferences and MCDA 7

2.1 Customer preferences in decision-making 7

2.2 From ratings to preference signals 8

2.3 Product-level recommendation classification 8

2.4 Multicriteria Decision Aid and Multicriteria Sorting 9

2.5 Preference disaggregation and UTADIS 11

3. Data Description and Preprocessing 13

4. Methodology 20

4.1 Experimental design 20

4.2 Software Tools and Model Implementation 21

4.3 LP-based UTADIS formulation 23

4.4 Benchmark models 25

4.5 Evaluation metrics 26

4.6 Additional analysis design 27

5. Results and Discussion 28

5.1 Threshold Sensitivity Analysis 32

5.2 Confusion Matrix Comparison 34

5.3 UTADIS Criteria Interpretation 39

5.4 Feature Group Contribution Analysis 42

6. Limitations and Future Work 45

6.1 Limitations.....45

6.2 Future Work.....47

7. Conclusion.....49

8. Bibliography.....52

Abstract

This thesis is dedicated to the development and evaluation of the AI module for the customer preferences system. The main task of the work is to classify products at the product level as recommended or non-recommended based on customer reviews and product metadata. The Amazon Reviews dataset for the Electronics category is used for the experiment. This dataset is suitable for this task because it contains both customer reviews and ratings, as well as product metadata (Hou et al., 2024; McAuley Lab, 2023). Ratings are converted to customer preference signals, after which the data is aggregated at the product level and combined with metadata. The input criteria are review-based indicators, metadata completeness features, price, brand, and manufacturer information.

The main reference method in the work is LP-based UTADIS, a method from the field of Multicriteria Decision Aid (MCDA) that constructs an additive utility function and a classification threshold. Its results are compared with several benchmark models: Majority baseline, Logistic Regression, Neural Network, and Weighted Neural Network. The comparison is carried out using accuracy, balanced accuracy and macro F1-score. The results show that the Weighted Neural Network achieves better predictive performance. At the same time, LP-based UTADIS remains useful as an interpretable multicriteria method that allows analysing the contribution of individual criteria to the final classification.

1. Introduction

In today's environment, companies are increasingly using customer, product, and user behaviour data to support management decisions. This is especially true for e-commerce, where customers leave ratings, reviews, helpful votes, and other traces of their interactions with products. Such data allows companies to better understand customer preferences: which products and characteristics are perceived positively by customers, and which may reduce the product's attractiveness.

Analysing customer preferences is an important task because they rarely depend on a single factor. The decision to purchase or evaluate a product may simultaneously be influenced by price, brand, manufacturer, number of reviews, verified purchases, completeness of product information, and other criteria. Therefore, to analyse such preferences, methods are useful that can take into account several features simultaneously and yield a clear result for decision-making.

This thesis concerns the development of an AI module for the customer preferences system. As part of the work, the task is considered at the product level. This means that the goal is not to build a personal recommendation system for each user, but to classify products based on aggregated customer signals and product metadata. This approach helps to determine which products in general can be considered recommended, and which are better classified as non-recommended.

Initially, such a task could be considered a rating prediction, that is, an attempt to predict an accurate product rating on a scale of 1 to 5. However, ratings are subjective, and the difference between neighbouring ratings does not always have a clear interpretation. For example, ratings 4 and 5 can both indicate a positive attitude towards a product, but they depend on the user's personal habits. Therefore, in this paper, the task was formulated as a binary product-level recommendation classification, where the product belongs to one of two classes: recommended or non-recommended.

The Amazon Reviews dataset for the Electronics category is used to build the experiment. This dataset is suitable for this task because it contains both customer reviews and ratings, as well as product metadata. Ratings are used to generate the aggregated customer preference outcome, and metadata and review-based indicators are used as input criteria for models. Thus, the original review-level data is converted into a product-level classification dataset.

The main reference method in this thesis is UTADIS. This is a method from the field of Multicriteria Decision Aid (MCDA), which is used for Multicriteria Sorting (MCS) tasks. UTADIS builds an additive utility function, where each criterion makes a certain contribution to the final utility of the product. This utility is then compared with the classification threshold, and the product belongs to one of the predefined classes. This paper uses LP-based UTADIS, that is, a formula based on linear programming.

Benchmark models are used to evaluate UTADIS results. The Majority baseline is used as the minimum comparison point. Logistic Regression is also used as a simple machine learning benchmark, a conventional Neural Network as an initial AI-based model, and a Weighted Neural Network as an improved AI-based model adapted to class imbalance. The main purpose of the thesis is to compare LP-based UTADIS with machine learning and AI-based benchmarks in the product-level recommendation classification task.

Thus, this work aims to build and evaluate an experimental framework for the customer preferences system. The thesis shows how customer reviews and product metadata can be used to form a product-level classification problem, and compares different approaches in terms of predictive performance and interpretability.

2. Background: Customer Preferences and MCDA

2.1 Customer preferences in decision-making

In modern companies, the analysis of customer preferences became one of the important parts of decision-making. Customer preferences usually refer to customers' preferences for products, services, brands, or a single element of an offer. These preferences can be explicitly expressed, for example, through ratings, reviews, purchases or repeat purchases and implicitly through customer behaviour, clicks, search history, helpful votes and other forms of interaction with the product.

Customer preferences are an important source of information for companies. They help companies understand which products are perceived positively by customers, which product characteristics make them more attractive, and which may reduce the likelihood of purchase or recommendation. This is especially important in the context of e-commerce, where the buyer faces a large number of similar products and must decide with limited information.

Customer preferences are rarely the result of a single factor. A customer can select a product not only if it has a high average rating, but also if it has a recognisable brand, a clear description, sufficient reviews, an affordable price, and thorough technical details. This suggests that preference formation is a multi-criteria process. The client evaluates the product using a set of attributes, even if he does not do it formally.

This thesis considers customers' product-level preferences rather than at the level of each individual user. This product-level approach makes practical sense: a company can pre-filter products suitable for recommendation and separate them from those with insufficiently positive customer feedback or that require additional verification.

2.2 From ratings to preference signals

One of the most common sources of customer preferences in e-commerce is rating. A rating is a simple numerical score a user gives a product after purchase or use. The rating is generally 1 to 5, where 1 is very negative and 5 is very positive.

But ratings are not the only thing. First, they are subjective: different users may use the same scale differently. Secondly, users might interpret the 3 rating differently. For some people, it's a neutral evaluation, for others it's a negative signal, and for others it's just an average result. So, treating all the ratings as strictly ordered classes can add noise to the model.

Thus, in this thesis, we use ratings to build customer preference signals instead of predicting exact ratings. Ratings 4 and 5 are positive preference signals, ratings 1 and 2 are negative preference signals, and rating 3 is neutral. This approach reduces the noise in the target variable and focuses on clearly expressed customer opinions.

The ratings are converted into preference signals and aggregated at the product level. For each product, the positive_ratio is computed, which is the proportion of positive preference signals out of all clear preference signals. Then, a recommended binary target variable is created based on positive_ratio.

2.3 Product-level recommendation classification

In classical recommendation systems, personalised recommendations are often the focus. These systems try to predict which products a particular user will like based on their past interactions, purchase history, ratings, or similarities with other users. However, personalised recommendations require quite detailed user-level data.

In this thesis, a different approach is taken: product-level recommendation classification. Here, the focus of analysis is the product itself, not a single user-product interaction. The product is described by a set of criteria, and the target indicates whether it is recommended based on aggregated customer feedback. This setting is not meant to replace the personalised recommendation system, but can be useful as part of it or as a separate decision-support tool.

Product-level classification can be used to preselect products for inclusion in the recommendation pool. Non-recommended products may be downgraded or sent for further review and if a product is classified as recommendable, it may be considered a better candidate for promotion, recommendation, or further ranking.

It should be noted that the criteria at the product level are not necessarily the direct causes of recommendability. For example, a long description does not mean that the product is good, but it can be an indirect indicator of the completeness of product information. The presence of brand or manufacturer information is no guarantee of quality, but it might indicate more transparent product metadata.

2.4 Multicriteria Decision Aid and Multicriteria Sorting

Customer preferences are formed under the influence of multiple factors; therefore, when analysing them, it is logical to use methods that can handle multiple criteria. That is where the Multicriteria Decision Aid (MCDA) comes into play. MCDA integrates methods to help to make decisions with multiple criteria and structure the decision process.

In MCDA, there are usually several types of tasks: a choice problem is to select one or more best alternatives; a ranking problem is to order alternatives from most preferred to least preferred; a sorting problem is a problem of sorting alternatives into some predefined categories. In this thesis, the task has a specific relation to the sorting problem, since the products are sorted into two predefined classes: recommended and non-recommended.

Multiple Criteria Sorting is a well-established field within MCDA that focuses on assigning alternatives to predefined, ordered categories based on multiple criteria (Belahcène et al., 2024). It is particularly useful when categories have a practical interpretation. In our product recommendation task, a recommended class means that the product has strong positive customer signals and suitable product-level indicators, warranting further consideration in the recommendation context. A not recommended class is when the product does not reach the chosen level or has characteristics that do not make it suitable for client recommendation.

Examples of Multicriteria Sorting. Many real-world decision-making problems can be represented as Multi-Criteria Sorting (MCS) problems, where alternatives are assigned to a set of predefined classes, usually ordered, based on multiple criteria.

- Assessment of the loan application. Before deciding, banks categorise loan applications according to their risk level. To this end, income status, debt status, credit history, business type, and loan amount are considered. Applicants can be classified as low, medium, high risk, or rejected.
- Evaluation of the supplier. The company assesses suppliers for reliability, product quality, price competitiveness, flexibility, sustainability practices and compliance. Suppliers can be categorised as a preferred supplier, an acceptable supplier, a supplier under supervision or a rejected supplier.

- Classification of products based on recommendations. * E-commerce products can be recommended or not recommended based on price, complete product information, confirmed purchase ratio, brand/manufacturer information, customer interaction and metadata-based metrics.

2.5 Preference disaggregation and UTADIS

In classical MCDA problems, the decision maker often has to specify weights, criteria, thresholds or other preference parameters a priori. Preference disaggregation addresses this problem in a different way: it uses examples of already classified alternatives rather than defining the preference model in advance. Preference disaggregation methods infer the parameters of a preference model from assignment examples rather than requiring the decision maker to define all weights and thresholds directly (Zopounidis & Doumpos, 1999; Belahcène et al., 2024).

The results presented in this thesis are based on customer preference signals. This means that the model does not receive the decision maker's manually set preferences, but instead uses aggregated customer feedback as the source for the class labels. Then, UTADIS, based on LP, tries to construct a utility function that justifies these labels by some selected criteria.

UTADIS is an MCDA methodology for multi-criteria sorting. It constructs an additive utility function for classifying alternatives. Each alternative is evaluated against several criteria, a marginal utility is computed for each criterion, and all marginal utilities are summed up into the overall utility score. The UTADIS method is based on an additive utility function and classification thresholds, and it has been widely used for multicriteria sorting and classification tasks (Zopounidis & Doumpos, 1999; Belahcène et al., 2024).

$$U(a) = \Sigma_i \, u_i(g_i(a))$$

where U(a) is the global utility alternative a, $g_i(a)$ is the value of alternative according to criterion i, and $u_i(g_i(a))$ is the marginal utility of this criterion. If global utility is higher than the classification threshold, the alternative belongs to the preferred class. If the threshold is lower, the alternative belongs to the less preferred class.

The advantage of UTADIS is that it allows not only to classify alternatives, but also to analyse the contribution of criteria. At the same time, UTADIS has limitations: its additive structure makes the model understandable, but less flexible compared to the Neural Network. Therefore, in this work, UTADIS is compared with machine learning and AI-based benchmarks to evaluate the trade-off between interpretability and predictive performance.

3. Data Description and Preprocessing

The Amazon Reviews dataset for the Electronics category from the source https://amazon-reviews-2023.github.io/ was used in the experiment. This dataset was chosen because it contains two types of information important for building a customer preferences system: on the one hand, customer reviews and ratings, and on the other hand, product metadata. As a result, it is possible to link the customer's reaction to the product's characteristics.

In this experiment, two source tables were used. The first table contained 1,000,000 reviews. Fields such as rating, title, text, asin, parent_asin, user_id, timestamp, helpful_vote, and verified_purchase were available in it. The second table contained 500,000 product metadata records. It featured product title, main_category, average_rating, rating_number, features, description, price, store, categories, details, and parent_asin. A common product identifier parent_asin is used to combine reviews and metadata.

Data source	Number of observations	Main fields
Customer reviews	1,000,000 reviews	rating, title, text, asin, parent_asin, user_id, timestamp, helpful_vote, verified_purchase
Product metadata	500,000 records	product title, main_category, average_rating, rating_number, features, description, price, store, categories, details, parent_asin

First of all, the ratings structure was analysed. In the initial sample, most of the ratings were positive: rating 5 was 649,913 observations, rating 4 - 141,739, rating 3 - 71,612, rating 2 - 47,705, rating 1 - 89,031. This structure shows that the data has a clear bias towards positive reviews, and it is typical for e-commerce reviews.

To clarify visually, the distribution of ratings is shown in Figure 1. The graph shows a strong concentration of positive ratings, especially at rating 5, further confirming the need for careful handling of class imbalance and the rejection of exact rating prediction.

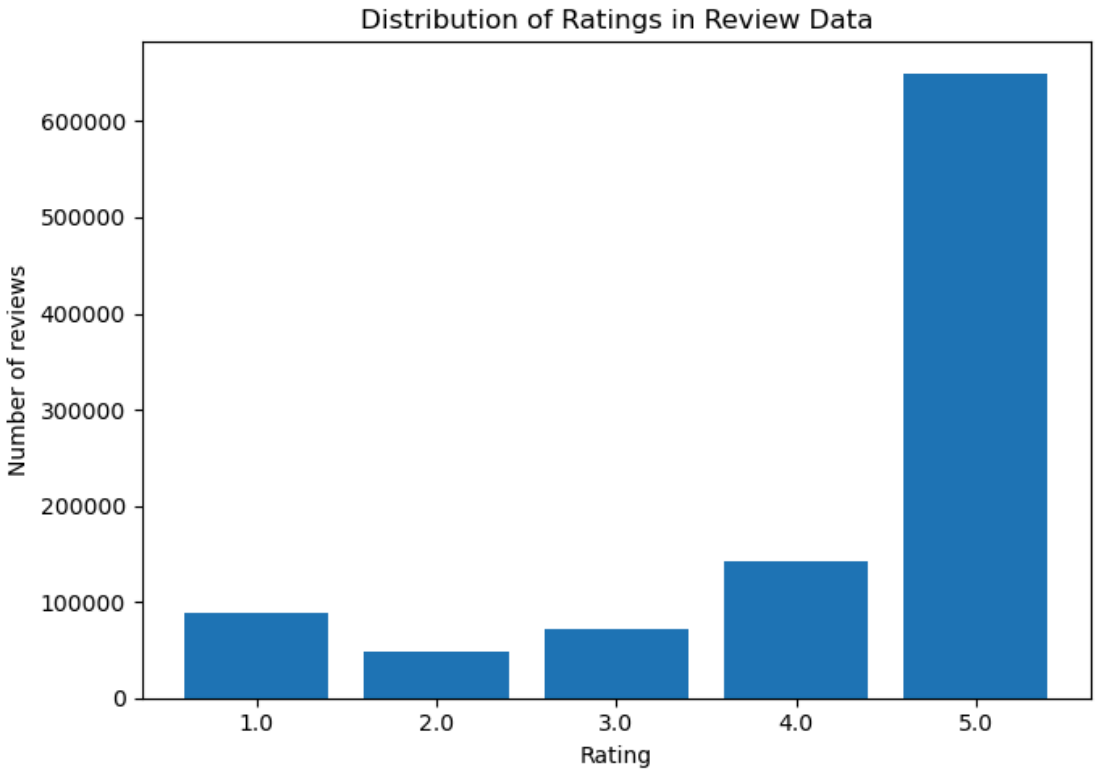


Figure 1. Distribution of ratings in the review data.

Rating	**Number of reviews**	**Interpretation in this thesis**

1	89,031	Negative preference signal
2	47,705	Negative preference signal
3	71,612	Neutral / excluded
4	141,739	Positive preference signal
5	649,913	Positive preference signal

The variable ratings were converted to customer preference signals to build the target variable. Ratings 4 and 5 were interpreted as positive preference signals, while ratings 1 and 2 were interpreted as negative preference signals. Rating 3 was excluded as neutral, as it can have an ambiguous meaning. The exclusion of the neutral class helped to make the target cleaner.

After that, the data was aggregated to the product level as discussed before. The proportion of positive signals was calculated for each product as in the formula.:

positive_ratio = number of positive preference signals / number of valid preference signals

where valid preference signals include only ratings 1, 2, 4, and 5. Also, only products with at least 5 valid preference signals were retained. This condition was used in the model in order not to classify a product based on too few reviews. After this condition, the product-level dataset contained 33,548 products.

The threshold for positive_ratio was used to create the binary target. Several threshold values were tested: 0.70, 0.75, 0.80, 0.85 and 0.90. Lower thresholds skewed too much towards the recommended class. The 0.85 level was chosen as the final threshold because it reflects a fairly strict recommendation idea and, at the same time, provides a more balanced class distribution.

$positive\_ratio \geq 0.85 \Rightarrow recommendable = 1$

positive_ratio < *0.85* ⇒ *recommendable* = *0*

After accepting the threshold value of 0.85, the result of combining into metadata was 19,731 recommended products and 13,817 non-recommended products. Consider the product-level validation data associated with the parent_asin user metadata. After the merger, the system simulation dataset contained 17,400 products. Of these, 10,497 products were classified as recommended, and 6,903 as not recommended. Thus, the distribution of targets was quite simple for binary classification: approximately 60.3% recommended and 39.7% not recommended.

The final target distribution is also shown in Figure 3. The diagram confirms that the dataset is moderately unbalanced, but it is not dominated by any one class to the same extent as in the initial rating distribution.

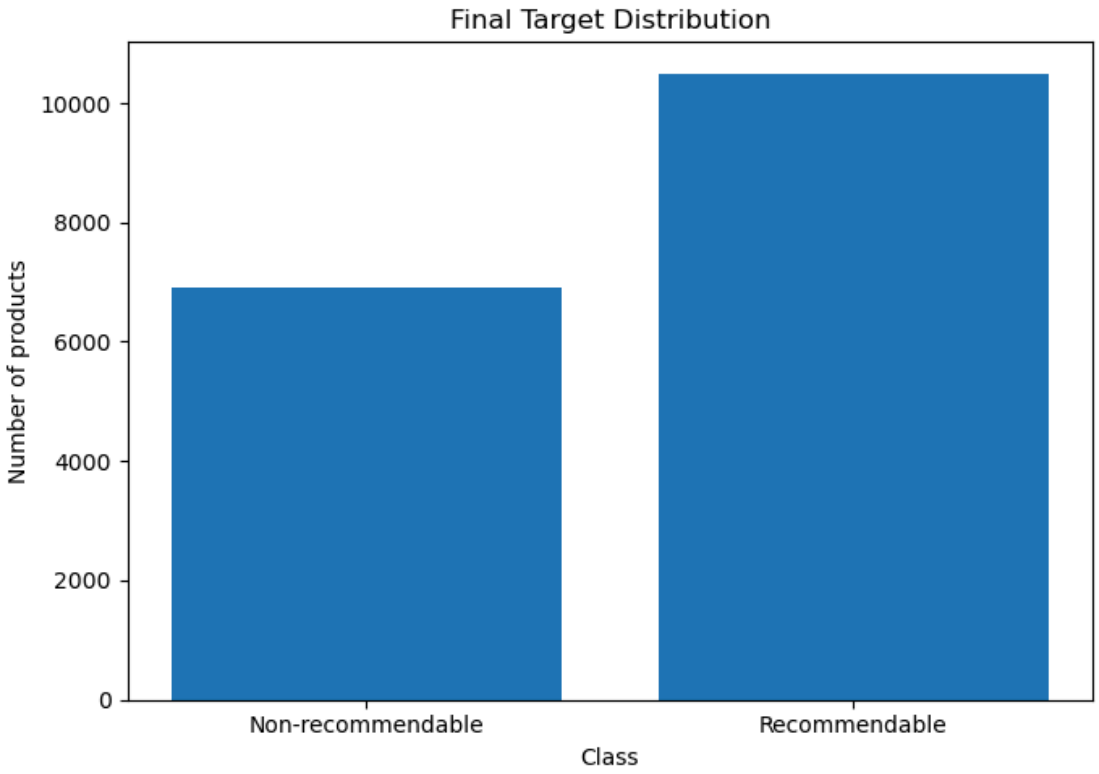


Figure 2. Final target distribution after merging reviews with metadata.

Figure 3 shows how the data volume was reduced at key stages of preprocessing. This visualisation helps to separate the raw data volume from the final modelling dataset, which was actually used for training and testing models.

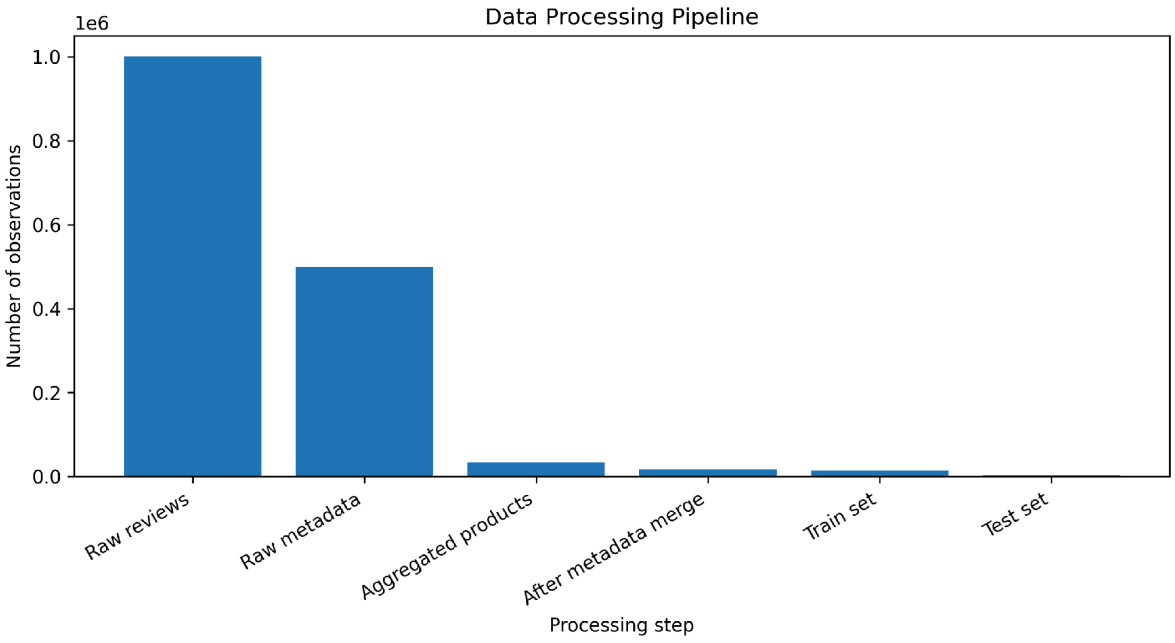


Figure 3. Data processing pipeline and number of observations at each stage.

Preprocessing step	**Number of observations**
Raw reviews	1,000,000
Raw metadata records	500,000
Products after aggregation and filtering	33,548
Products after metadata merge	17,400
Recommendable products after merge	10,497
Non-recommendable products after merge	6,903
Train set	13,920
Test set	3,480

Before modelling, signs of potential data leakage were excluded. In particular, positive_ratio was not used as an input feature because it was directly used to create a target variable. avg_rating, average_rating, and rating_number were also not used, as they are too closely related to the rating-based outcome and could artificially improve model performance.

Criteria were selected to build the models, which can be divided into several groups. The first group includes aggregated review-based indicators; the second group concerns the completeness of product metadata; and the third group concerns price and brand/manufacturer information. The categorical features main_category and store were also used in the machine learning models. To avoid too many categories, rare values were combined into the category Other, and then processed through One-Hot Encoding. Numeric features were processed by filling in missing values with the median. It was especially important to process the price, since this feature had a large number of missing values in the metadata: about 62.2% of the price values were missing. After preprocessing, there are no missing values left in the final feature matrix.

Feature group	**Features**	**Meaning**
Review-based indicators	valid_rating_count, helpful_vote_avg, verified_ratio	Customer feedback activity and reliability
Metadata completeness	title_length, features_count, description_length,	Completeness of product information and product presentation

	categories_count, details_count	
Price	price	Basic product cost indicator
Brand/manufacturer information	has_brand, has_manufacturer, log_brand_popularity, log_manufacturer_popularity	Product identity, brand/manufacturer availability and frequency-based signals
Categorical metadata	main_category, store	General product category and seller/store-related information

Thus, the data at the overview level has been transformed into a set of classification data at the product level. This requires the product to be recommended or not recommended, based on aggregated review metrics, metadata completeness, price, brand/manufacturer information, and categorical metadata.

4. Methodology

4.1 Experimental design

The methodological part of the thesis is built around a single experimental framework. The main purpose of this framework is to prepare a product-level dataset, train several classification models, compare them in terms of predictive performance and interpretability. This approach makes it possible to evaluate LP-based UTADIS not in isolation, but in comparison with simpler and more flexible benchmark models and neural network models.

The overall experiment pipeline includes the following steps:

1. Load customer reviews and product metadata from the Amazon Reviews Electronics dataset.
2. Transform raw ratings into customer preference signals.
3. Aggregate reviews at the product level using parent_asin.
4. Construct positive_ratio and binary target recommendable using a threshold of 0.85.
5. Merge product-level review data with product metadata.
6. Create review-based, metadata-based, price and brand/manufacturer criteria.
7. Remove leakage variables such as positive_ratio and rating-based aggregate outcomes.
8. Split the final dataset into train and test sets using stratification.
9. Train Majority baseline, Logistic Regression, Neural Network, Weighted Neural Network and LP-based UTADIS.
10. Evaluate models using accuracy, balanced accuracy, macro F1-score and confusion matrices.

Such a pipeline makes the experiment reproducible and logically connected: first, a target is defined based on customer feedback, then product-level criteria are created, and finally different classification methods solve the same problem.

4.2 Software Tools and Model Implementation

The experimental part was implemented in a Python Jupyter Notebook, as Python is one of the most widely used programming languages for data analysis, data science, machine learning, and artificial intelligence. It contains an enormous ecosystem of different libraries for all possible purposes: working with tabular data, numerical computations, visualisation, model training and evaluation.

Several Python libraries were used during the implementation. Here is the list of them:

- `pandas` library - was used for loading, cleaning, transforming and aggregating review and metadata tables. Besides that, it was used to group customer reviews at the product level and to create product-level indicators.
- `numpy` library - was used for numerical operations, array manipulation and mathematical transformations.
- `matplotlib` and `seaborn` - data visualisation was carried out using these libraries. Also, they were used to create rating distribution plots, target distribution plots, model comparison charts, threshold sensitivity analysis and confusion matrices.
- `scikit-learn` - was used for machine learning preprocessing and model evaluation. It was applied to train-test splitting, feature scaling, one-hot encoding of categorical variables, imputation of missing values, model training, and the calculation of evaluation metrics. In particular, accuracy, balanced accuracy, macro F1-score and confusion matrices were calculated using scikit-learn tools.

- SciPy - was used for optimisation tasks. In particular, the `linprog` function from `scipy.optimize` was used to solve the linear programming problem underlying the LP-based UTADIS model.

The machine learning benchmark models were also implemented in Python. The Majority Baseline was implemented as a simple reference model that always predicts the most frequent class. Logistic Regression was implemented using scikit-learn as a linear benchmark model, *LogisticRegression*. The Neural Network and Weighted Neural Network were implemented using scikit-learn's MLPClassifier, which represents a feed-forward multilayer perceptron neural network. The standard Neural Network was trained as a flexible nonlinear classifier, and the Weighted Neural Network used sample weights to reduce the effect of class imbalance and improve recognition of both classes.

The LP-based UTADIS model was implemented using Python's linear programming tools. The main idea of the implementation is to construct the additive utility function from normalised criteria and to learn the classification threshold by minimising weighted classification deviations. The model included monotonicity constraints on marginal utilities, a normalisation constraint on total utility, slack variables for imperfect separability, and an additional upper bound on the maximum contribution of each criterion. This upper bound was introduced to avoid a situation in which the classification result depends almost entirely on one criterion. The optimisation problem was solved using the following Python tool:

from scipy.optimize import linprog

from sklearn.utils.class_weight import compute_sample_weight

The linprog function was used to solve the linear programming formulation of UTADIS, while compute_sample_weight was used to compute class-balanced sample weights for the objective function.

Overall, Python enabled the full experimental pipeline to be built reproducibly. The same environment was used for data preprocessing, feature engineering, target construction, model training, model comparison, visualisation, and interpretation of UTADIS criteria utilities.

4.3 LP-based UTADIS formulation

The main method of the current experiment is LP-based UTADIS - a multicriteria classification model based on the construction of an additive utility function and evaluation of the classification threshold through linear programming. In contrast to conventional machine learning methods, UTADIS is interesting not only for its predictive performance but also for its interpretability.

Let there be a set of products A = {A_1, a_2, ..., a□}. Each product is described by m criteria a: $g_1(a)$, $g_2(a)$, ..., g□(a). UTADIS is building a global utility function:

$$U(a) = \Sigma_i \, u_i(g_i(a))$$

where U(a) is the global utility of the product, $g_i(a)$ is the value of the product according to criterion i, and $u_i(g_i(a))$ is the marginal utility of this criterion. The binary classification uses a single classification threshold u. A product is classified as recommended if its utility is not lower than the threshold, and as non-recommended if its utility is lower than the threshold.

$$U(a) \geq u \Rightarrow a \in recommendable$$

$U(a) < u \Rightarrow a \in$ *non-recommendable*

In the practical implementation criteria, they are first normalised. A piecewise marginal utility function is built for each criterion. In order for utility functions to be logical, a monotonicity condition is introduced: after determining the direction, all criteria are reduced to the form larger is better. And the marginal utility should be non-decreasing.

$u_i(x_{i1}) \leq u_i(x_{i2}) \leq \ldots \leq u_i(x_{i□})$

The normalisation condition is also used so that the total utility is comparable between products:

$\Sigma_i u_i(1) = 1$

Since the real data is not always perfectly separable, slack variables $\sigma□ \geq 0$ are added to the model. For recommended products, the utility must exceed the threshold by a margin δ, and for non-recommended products, it must fall below the threshold. If the condition is violated, the slack variable shows the magnitude of deviation.

$U(a□) \geq u + \delta - \sigma□,$ *for* $a□ \in$ *recommendable*

$U(a□) \leq u - \delta + \sigma□,$ *for* $a□ \in$ *non-recommendable*

The objective function minimises the weighted sum of slack variables. Sample weights are used to account for class imbalance, ensuring the model does not focus solely on the majority class.

$min \Sigma□ w□\sigma□$

This paper also uses an additional restriction on the maximum criterion utility to avoid a situation when the entire utility function depends almost entirely on a single criterion. This limitation makes

the model more consistent with multicriteria logic, although it may slightly reduce predictive performance.

$u_i(1) \leq 0.30$

4.4 Benchmark models

Several benchmark models are used to evaluate the main model, LP-based UTADIS. This is necessary to understand how competitive the chosen MCDA method is compared to simpler, more flexible machine learning methods.

Model	Role in experiment	Expected behavior
Majority baseline	Minimum comparison level	High accuracy if the majority class dominates, but low balanced accuracy
Logistic Regression	Simple machine learning benchmark	Stable performance if the relationship between features and target is relatively linear
Neural Network	Initial AI-based benchmark	Flexible model, but may shift toward the majority class
Weighted Neural Network	AI-based benchmark adjusted for class imbalance	More balanced recognition of recommendable and non-recommendable classes
LP-based UTADIS	Interpretable MCDA reference method	Lower flexibility, but better interpretability through a utility-based structure

The Majority baseline always predicts the most frequent class and achieves the lowest performance. Logistic Regression is a simple benchmark that estimates the probability of class membership based on a linear combination of features. The Neural Network was used as an AI-based benchmark capable of capturing nonlinear relationships between input features and the target variable. After analysing its behaviour, a Weighted Neural Network was added, as the ordinary Neural Network tended to predict the majority class more often. The weighted version uses sample weights to better account for both classes. Neural network models are widely used in recommendation-related tasks because they can capture nonlinear relationships among users, products, and product features (He et al., 2017; Hossain et al., 2026).

The basic multicriteria method is represented by UTADIS based on LP. It is valuable not only for its predictive performance, but also for its ability to explain the classification by the utility function and the contributions of the criteria.

4.5 Evaluation metrics

The models were evaluated using accuracy, macro F1 score, balanced accuracy and confusion matrix. Multiple metrics are important to use because target classes are not perfectly balanced. In such a case, ordinary accuracy may be misleading.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

Accuracy shows the total proportion of correct predictions. However, if one class is more common, the model can achieve relatively high accuracy by simply predicting the majority class.

$$Balanced\ Accuracy = (Recall_0 + Recall_1) / 2$$

Balanced accuracy considers the recognition quality of each class separately and averages the results. So it is more advantageous for moderately balanced classification tasks.

$F1 = 2 \cdot (Precision \cdot Recall) / (Precision + Recall)$

Macro F1-score calculates F1-score for each class individually and then averages the results. This makes the metric useful when it is important to consider both recommendable and not recommendable products. To get a more detailed analysis of the errors, we use a confusion matrix. In this task, a false positive occurs when the non-recommendable product is classified as recommendable. A false negative means that a recommendable product was classified as non-recommendable. Both types of errors are important for the customer preference system, but false positives are especially critical if the system can recommend inappropriate products.

4.6 Additional analysis design

Besides the main model comparison, several additional analyses were included, which made the experimental framework more transparent. First, the sensitivity of the threshold analysis was assessed to justify the selected positive_ratio threshold for constructing the recommended target. Secondly, confusion matrices were analysed to understand the model's behaviour at the class level rather than relying solely on aggregate metrics. Finally, the multicriteria model was evaluated for its explanatory power by interpreting the UTADIS criteria utilities.

These additional analyses are important because the thesis is not only interested in obtaining the highest numerical score. Also, it is intended to understand how the target was constructed, how models behave with respect to each class, and how the UTADIS utility structure can help in decision-making. So the results section combines predictive comparison and threshold analysis, as well as the class-level error analysis and criteria interpretation.

5. Results and Discussion

At the final product level, five models were evaluated: Majority baseline, Logistic Regression, Neural Network, Weighted Neural Network and LP-based UTADIS. The comparison was designed to assess the task from two perspectives: predictive performance and interpretability. The Majority baseline was used as a minimum reference point, Logistic Regression as a simple machine learning benchmark, Neural Network as a nonlinear AI-based benchmark, Weighted Neural Network as a model adjusted for class imbalance, and LP-based UTADIS as the main interpretable MCDA method.

The test set contained 2,099 recommendable products and 1,381 non-recommendable products. Since the recommendable class is more frequent, a model biased toward this class can obtain relatively high accuracy while still performing poorly on non-recommendable products. Therefore, balanced accuracy and macro F1-score were used together with accuracy to evaluate model performance more correctly.

Figure 4 compares the main assessment indicators across all models. The figure clearly shows the difference between the usual accuracy metric and the balanced indicators: models with the same accuracy can behave very differently when both classes are taken into account.

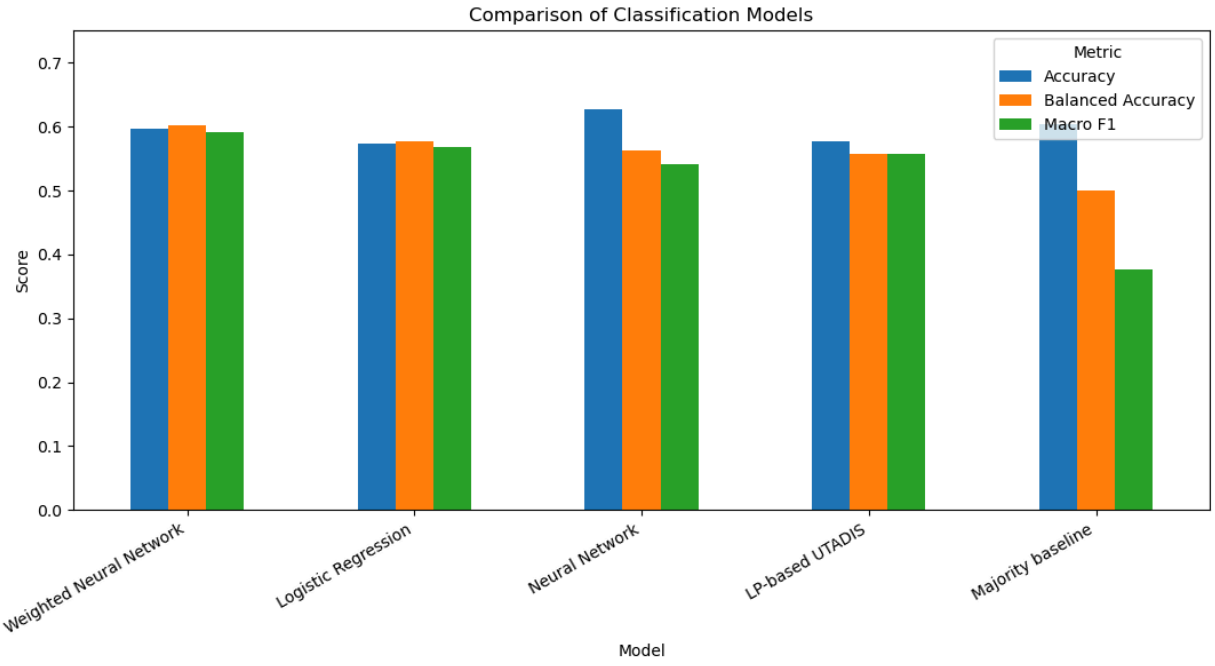


Figure 4. Comparison of classification models by accuracy, balanced accuracy and macro F1-score.

Model	**Accuracy**	**Balanced accuracy**	**Macro F1**
Majority baseline	0.603	0.500	0.376
Logistic Regression	0.573	0.576	0.569
Neural Network	0.628	0.562	0.539
Weighted Neural Network	0.596	0.601	0.590
LP-based UTADIS	0.577	0.557	0.556

The Majority baseline was calculated first. This model always predicts the most frequent class, i.e., the recommended one. It achieved an accuracy of 0.603. At first glance, this result may seem good because the model correctly classifies most objects, given the predominance of one class.

However, the balanced accuracy was only 0.500, and the macro F1-score was 0.376. The Confusion matrix confirms this: all 1,381 non-recommended products were mistakenly assigned to the recommended class. Therefore, the Majority baseline is not a useful model, but it is important as a minimum reference point.

Logistic Regression achieved an accuracy of 0.573, a balanced accuracy of 0.576, and a macro F1-score of 0.569. Compared to the Majority baseline, ordinary accuracy has decreased, but this does not indicate a deterioration in quality. On the contrary, Logistic Regression began to better recognise both classes. The model correctly classified 804 out of 1,381 non-recommended products and 1,198 out of 2,099 recommended products. This shows that the selected product-level criteria do contain information related to the recommendation outcome.

The usual Neural Network received an accuracy of 0.628, a balanced accuracy of 0.562 and a macro F1-score of 0.539. Its accuracy was high, but a more detailed analysis showed a bias towards the recommended class. Neural Network correctly classified 1,819 out of 2,099 recommended products, but only 365 out of 1,381 non-recommended products. This means that the model often referred non-recommended products to the recommended class. This is a problem for the customer preferences system, because non-recommended products are also important.

For this reason, a Weighted Neural Network was additionally built. This version used sample weights to reduce the impact of class imbalance. The Weighted Neural Network showed the best result among all tested models: accuracy 0.599, balanced accuracy 0.605 and macro F1-score 0.595. Although its ordinary accuracy is lower than that of a simple Neural Network, it better balances the recognition of the two classes. The model correctly classified 882 out of 1,381 non-recommended products and 1,201 out of 2,099 recommended products.

LP-based UTADIS achieved an accuracy of 0.577, a balanced accuracy of 0.557, and a macro F1-score of 0.556. Compared to the Majority baseline, UTADIS achieves higher balanced accuracy and macro F1-score. This means that the model does not just repeat the majority class, but tries to separate recommended and non-recommended products. The confusion matrix shows that LP-based UTADIS correctly classified 640 out of 1,381 non-recommended products and 1,353 out of 2,099 recommended products.

When compared with machine learning benchmarks, LP-based UTADIS is inferior to Weighted Neural Network and Logistic Regression in predictive performance. However, this does not mean that UTADIS is useless for this task. Its role in the thesis is different from that of purely predictive models. UTADIS builds an additional utility function and allows us to analyse which criteria have contributed more to the final utility score. This makes the model more understandable for decision-making.

Comparing all the models, an important trade-off can be noticed. The majority baseline has relatively high accuracy but little classification value. The usual Neural Network shows high accuracy, but does not recognise the non-recommended class well enough. Logistic Regression shows a stable and fairly balanced result. The Weighted Neural Network shows the best result for balanced accuracy and macro F1-score. LP-based UTADIS is not the best predictive metrics model, but it remains valuable because of its interpretability and utility-based representation.

In general, the results of the experiment show that building a customer preferences system requires a balance between predictive performance and interpretability. If the main goal is only to maximise classification quality, the Weighted Neural Network turns out to be the most suitable model among the tested approaches. If transparency of the solution and the ability to understand the

contribution of individual criteria are important, LP-based UTADIS remains a useful interpretable reference method.

5.1 Threshold Sensitivity Analysis

One additional analysis tested the sensitivity of the target variable to the chosen threshold. In this thesis, the final threshold was 0.85, but it is important to show that this decision was not completely random. Since the recommended class label is derived from positive_ratio, threshold variation directly affects the class distribution and thus the difficulty of the classification problem.

In this case, different threshold values were tested: 0.70, 0.75, 0.80, 0.85 and 0.90. Lowering the threshold values makes the definition of a recommended product less strict. In this case, a large fraction of the products belong to the recommended class, making the target highly imbalanced. On the contrary, a higher threshold makes the recommendation rule stricter and decreases the proportion of recommended products. Therefore, the choice of threshold is an important part of the experimental design.

The results of the threshold sensitivity analysis are illustrated in Table 6. After aggregation and filtering, the product-level dataset contained 33,548 products prior to combining with metadata. For each threshold, the proportion of recommended and non-recommended products was calculated.

Threshold	**Recommendable products**	**Non-recommendable products**	**Recommendable share**
0.70	28,818	4,730	85.9%
0.75	27,433	6,115	81.8%
0.80	25,569	7,979	76.2%
0.85	19,731	13,817	58.8%

0.90	15,403	18,145	45.9%

Table 6 shows that thresholds 0.70, 0.75 and 0.80 produce a highly imbalanced target dominated by the recommendable class. For example, with a threshold of 0.70, 85.9% of products are classified as recommendable. In such a setting, a simple majority-based model could achieve relatively high accuracy without truly distinguishing between recommendable and non-recommendable products. Therefore, these lower thresholds were considered less suitable for the purpose of this thesis.

A similar picture is shown graphically in Figure 5. The proportion of recommended products decreases as the threshold value is tightened. The chosen value of 0.85 represents a compromise between a strict recommendation rule and an acceptable class distribution.

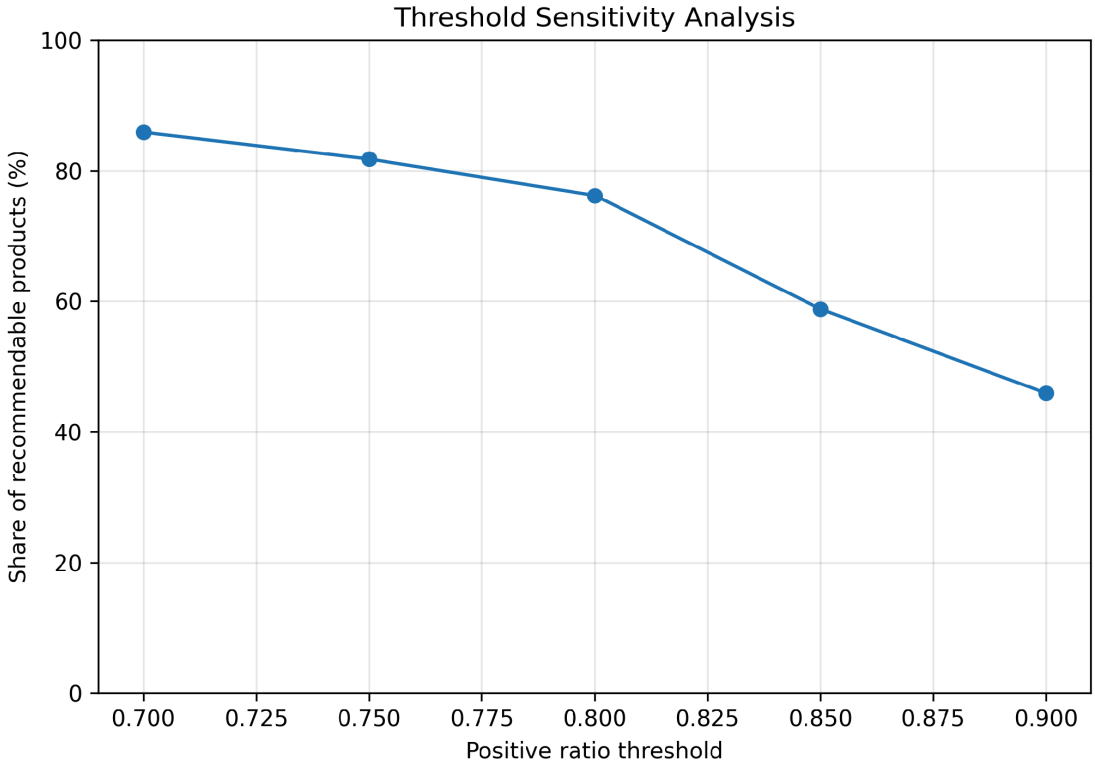


Figure 5. Threshold sensitivity analysis based on the share of recommendable products.

Threshold 0.90, by contrast, is stricter and results in the recommended class being smaller than the non-recommended class. Such a statement can be useful in a very conservative recommendation setting, but can exclude some products with sufficiently strong positive customer feedback. In this paper, we chose a threshold of 0.85 as a compromise. It maintains the strict notion of recommendation while, at the same time, creating a more balanced target distribution than lower thresholds.

The final modelling dataset included 17,400 products after merging with the metadata at a threshold of 0.85, including 10,497 products classified as recommended and 6,903 as non-recommended. This distribution is not perfectly balanced, but it is acceptable for binary classification and for evaluating models using balanced accuracy and macro F1-score.

5.2 Confusion Matrix Comparison

In addition to aggregate metrics such as accuracy, balanced accuracy and macro F1-score, confusion matrices were considered to interpret the results. This is particularly important in this task because the two errors have different practical meanings. A false positive means a product not recommended is classified as recommended. This may be a problem for the recommendation system, as it may promote a product with insufficient customer preference signals. A false negative means that the product to be recommended has been classified as non-recommended. It may result in the loss of a potentially good product from the recommendation pool.

The confusion matrix describes which models are really good at recognising both classes and which models are mostly just looking at the majority class. Table 7 shows the confusion matrices of all models used in the experiment. The rows in the table are actual classes, and the columns are predicted classes.

For ease of reading, the normalised confusion matrices are shown below. By normalising by the true class, we can compare models regardless of the absolute number of observations in each class.

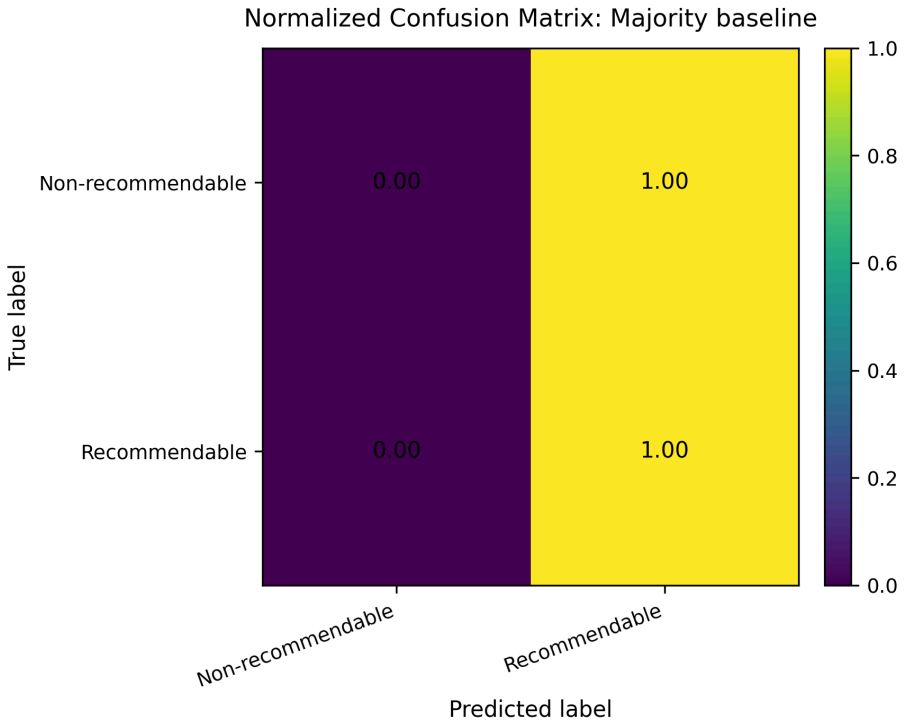


Figure 6. Normalised confusion matrix for Majority baseline.

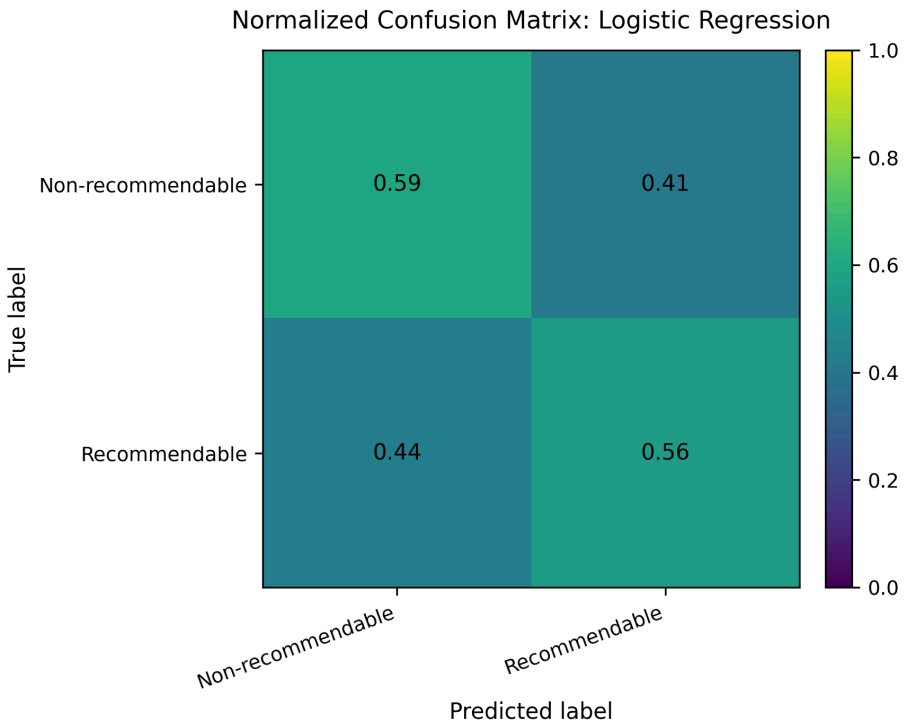


Figure 7. Normalised confusion matrix for Logistic Regression.

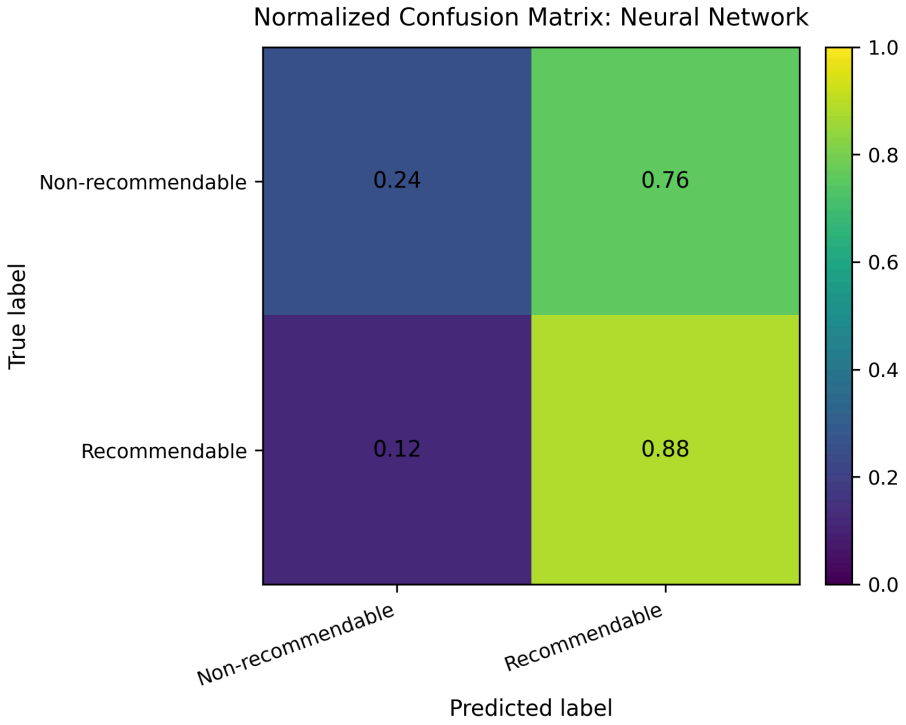


Figure 8. Normalised confusion matrix for Neural Network.

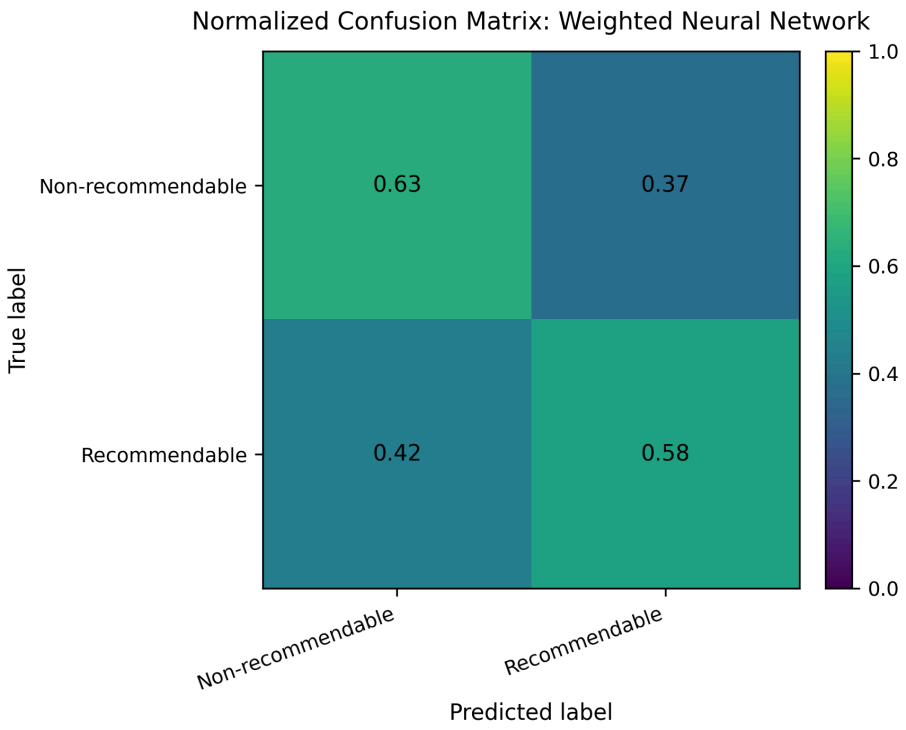


Figure 9. Normalised confusion matrix for Weighted Neural Network.

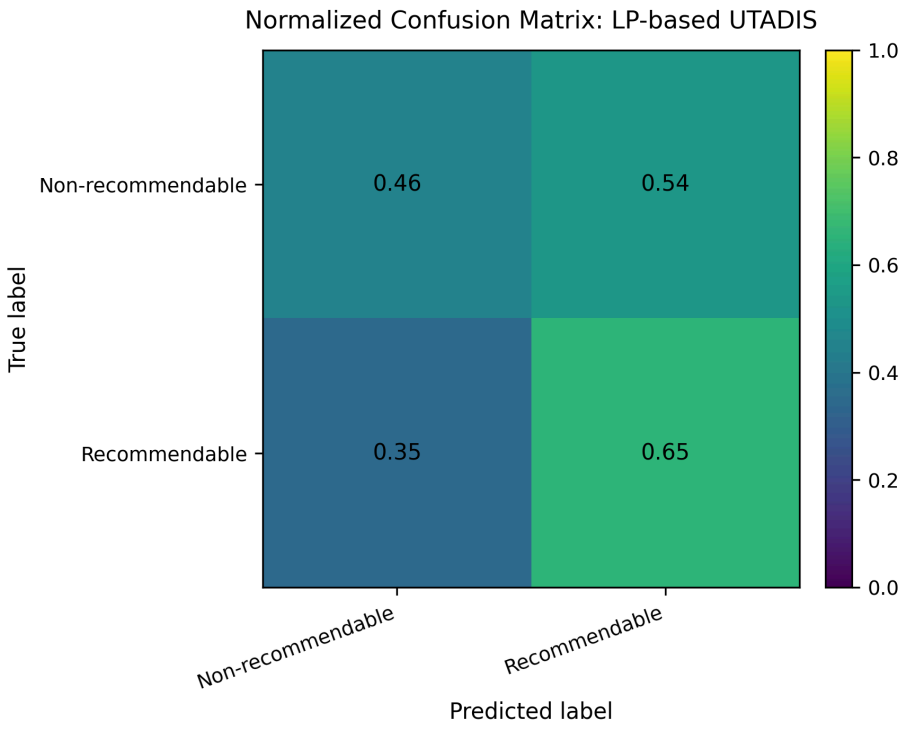


Figure 10. Normalized confusion matrix for LP-based UTADIS.

Model	**TN: actual 0, predicted 0**	**FP: actual 0, predicted 1**	**FN: actual 1, predicted 0**	**TP: actual 1, predicted 1**
Majority baseline	0	1,381	0	2,099
Logistic Regression	815	566	919	1,180
Neural Network	336	1,045	250	1,849
Weighted Neural Network	865	516	890	1,209
LP-based UTADIS	636	745	726	1,373

The most obvious problem is visible in the majority baseline: the model does not predict class 0 at all. All non-recommended products were coded as recommended. Thus, with an accuracy of about 0.603, this model has no practical value for the customer preferences system. It just shows what kind of result could be achieved in case of totally ignoring the signs and always choosing the majority class.

Logistic Regression behaviour is more fair. It correctly identifies 815 products it does not recommend and 1,180 it does. That means the model is really trying to discriminate between the two classes. At the same time, the number of false negatives is quite high, which indicates the complexity of the task: the profile of features of the recommended products part is similar to that of the non-recommended products.

The common Neural Network has a different problem. It correctly classifies 1,849 recommended products but only finds 336 non-recommended products. Therefore, it is the best in terms of accuracy among some models, but its balanced accuracy and macro F1 score are lower than those of the Weighted Neural Network. The model is overfitting to the recommended class.

The Weighted Neural Network has the most balanced confusion matrix among the predictive models. It finds 882 non-recommended products, which is much better than a regular Neural Network. At the same time, it keeps the acceptable recommended class recognition. This result confirms that the inclusion of sample weights was an appropriate response to class imbalance.

UTADIS, based on LP, is somewhere in the middle. Correctly classified 636 products as not recommended and 1,373 products as recommended. This is better than the Majority baseline but weaker than Logistic Regression and Weighted Neural Network on balanced metrics. The confusion matrix, however, shows that UTADIS does not merely reproduce the majority class but also establishes a genuine multicriteria classification boundary.

5.3 UTADIS Criteria Interpretation

As UTADIS is the main reference method in this thesis, it is important to analyse its results not only through predictive metrics but also through the structure of the resulting utility function. Unlike the Neural Network, UTADIS allows the decision maker to see which criteria have received a greater contribution to the final utility score.

The last LP-based UTADIS model used the numerical criteria that were available after the preprocessing stage. All criteria were normalised, and the direction of each criterion was empirically determined from the training data. This means that directions should not be interpreted as universal causal relations, but as a data-driven orientation within a given experiment.

Criterion	**Direction**	**Maximum marginal utility**
valid_rating_count	Benefit	0.300
helpful_vote_avg	Cost	0.300
log_brand_popularity	Benefit	0.170
has_brand	Benefit	0.167
title_length	Cost	0.014
categories_count	Benefit	0.012
verified_ratio	Benefit	0.011
description_length	Cost	0.010
details_count	Cost	0.006
log_manufacturer_popularity	Cost	0.004
features_count	Cost	0.004

price	Cost	0.001
has_manufacturer	Benefit	0.000

Table 8 and Figure 11 show that two criteria, valid_rating_count and helpful_vote_avg, reached the maximum marginal utility constraint of 0.300. This suggests that the model relied most on review activity and helpfulness-related information. At the same time, the constraint did not allow one criterion to completely dominate the entire utility function.

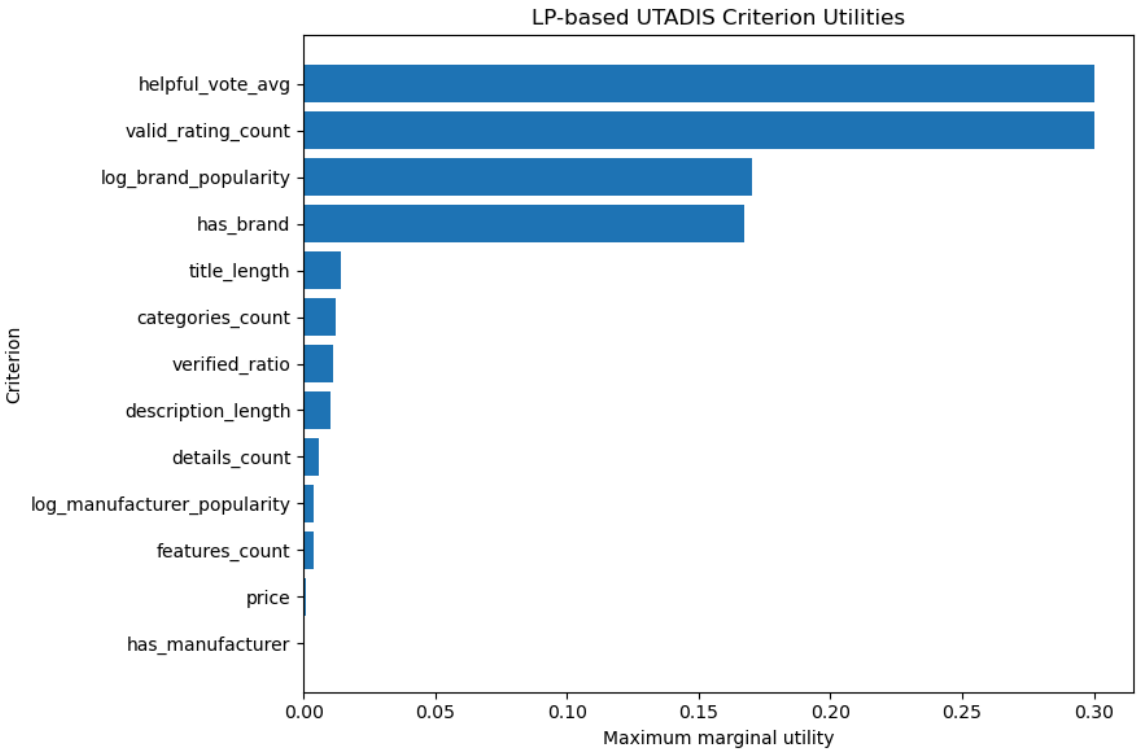


Figure 11. LP-based UTADIS criterion utilities.

valid_rating_count received a high contribution as a benefit criterion. This can be interpreted so that products with a large number of clear preference signals provide more stable information about customer perception. For product-level classification, such a criterion is important because the recommendation decision becomes more reliable when it is based on more customer signals.

helpful_vote_avg also received a high maximum marginal utility, but was classified as a cost criterion in the empirical direction procedure. This does not mean that helpful votes generally degrade the product. A more correct interpretation is that in this dataset, a high helpful_vote_avg may be associated with more controversial or problematic products, where users are more likely to rate reviews as helpful.

The next most important were log_brand_popularity and has_brand. Their utility values show that brand-related information plays a significant role in classification. This supports the idea that product identity and the availability of brand information can be linked to customer trust and the outcome of recommendations.

The other criteria received smaller marginal utility values. This does not mean that they are completely useless, but their contribution to the additive utility function turned out to be more limited. For example, title_length, categories_count, verified_ratio, description_length and details_count contributed to the model, but not as strongly as the leading review-based and brand-related criteria.

In general, the interpretation of the UTADIS criteria shows that the model relied mostly on review activity, helpfulness patterns, and brand-related metadata. This is useful for the current thesis, as UTADIS provides not only a class prediction, but also an interpretable explanation of which product-level criteria are more strongly connected with the final classification.

5.4 Feature Group Contribution Analysis

To provide additional interpretation of the criteria, a feature-group analysis was conducted. Its goal is not to add another competing model, but to explain and show which types of information are used in the experiment and what role they could play in the classification task. This is especially

important for this thesis, since UTADIS and MCDA logic assume a conscious selection of criteria rather than the random use of all available columns.

All selected features can be divided into several groups: review-based indicators, metadata completeness indicators, price, brand/manufacturer information, and categorical metadata. This grouping helps to link technical feature engineering with decision-making interpretation. In the customer preferences system, it is important to understand not only that the model received a certain score, but also which dimensions of product representation were included in the decision process.

Feature group	**Main variables**	**Interpretation**
Review-based indicators	valid_rating_count, helpful_vote_avg, verified_ratio	Reflect activity, reliability and structure of customer feedback
Metadata completeness	title_length, features_count, description_length, categories_count, details_count	Reflect how completely the product is represented in metadata
Price information	price	Represents product cost, although its effect may differ across subcategories
Brand/manufacturer signals	has_brand, has_manufacturer, log_brand_popularity, log_manufacturer_popularity	Capture product identity, metadata availability and frequency-based patterns

Categorical metadata	main_category, store	Capture broader category and seller/store-related context for machine learning models

Review-based indicators are closest to customer feedback, but they were selected carefully to avoid direct data leakage. Variables such as positive_ratio, avg_rating, average_rating, and rating_number were excluded because they are directly related to the target construction. Instead, variables such as valid_rating_count, helpful_vote_avg and verified_ratio were used to describe the amount and reliability of feedback without directly revealing the target label.

Metadata completeness indicators were included because product presentation can affect customer understanding and trust. For example, a product with a more complete description, clearer title and structured details may be easier for customers to evaluate. At the same time, these variables are only indirect proxies. They do not prove product quality, but they describe how much information is available about the product.

Brand and manufacturer variables were included to capture product identity. The binary variables has_brand and has_manufacturer separate the presence of this information from frequency-based popularity variables. This distinction is important because missing brand/manufacturer information may itself be informative. Meanwhile, log_brand_popularity and log_manufacturer_popularity measure how frequently a given category appears in metadata, reducing the dominance of very frequent values through logarithmic transformation.

A full numerical ablation experiment by feature group would be a natural extension of this analysis. However, in the current notebook, the final additional outputs available for interpretation are threshold sensitivity, confusion matrices and UTADIS criterion utilities. Therefore, this section

uses feature group analysis as an interpretive complement to the model results rather than as a separate performance table with invented values.

6. Limitations and Future Work

6.1 Limitations

The thesis has developed a comprehensive experimental framework for product-level recommendation classification, but the work still has several limitations. These limitations are related to the data, the construction of the target variable, the choice of criteria and the models used. These are important to consider when interpreting the results.

The first limitation concerns the nature of customer ratings. In this paper, we use ratings as the primary source of customer preference signals. That is a logical approach, but ratings are still subjective. Different users may use the same scale differently. The rating may be determined not only by the product quality, but also by delivery experience, expectations, price perception or customer service.

The second limitation is that the task was designed at the product level rather than the individual user level. This means the model does not recommend specific items for a particular user. With this approach, products can be classified based on aggregated signals. This approach is useful for pre-evaluation of products, product screening or decision support, but it does not replace a full-fledged personalised recommendation system.

The third limitation concerns the threshold selection used to build the target variable. In this paper, we recommend product belongs to class if the positive_ratio is no less than 0.85. This was a selected threshold, and any solution based on a threshold is still a modelling assumption. If the threshold is selected as 0.80 or 0.90, the target distribution and the results may vary.

The fourth limitation is the feature engineering. The work used review-based indicators, metadata completeness features, price, brand and manufacturer information. These features make it possible to describe a product from different perspectives, but they are not always direct characteristics of product quality. For example, title_length or description_length reflects the completeness of the product information, not the product's actual quality.

The fifth limitation relates to brand and manufacturer variables. has_brand and has_manufacturer indicate whether the metadata contains the relevant information. The variables log_brand_popularity and log_manufacturer_popularity use frequency-based encoding to capture brand or manufacturer categories. But these variables may capture not only real popularity, but also patterns of information availability in meta data.

The sixth limitation is that the features used in the work were mostly structured and engineered, but the text reviews were not analysed in depth. Textual feedback may contain a lot of information about the causes of satisfaction or dissatisfaction, but its use requires additional NLP methods and may increase the risk of data leakage.

The seventh restriction concerns LP-based UTADIS. The thesis implemented the basic linear programming formula. More complex variants, such as UTADIS II or UTADIS III were not implemented due to the need for more complex optimisation, such as mixed integer programming. Moreover, the extra utility structure makes UTADIS interpretable, but less flexible than the Neural Network. More advanced MCDA sorting models, including ELECTRE TRI-C and robust ordinal regression with sets of additive value functions, could be considered in future extensions of this work (Almeida-Dias et al., 2010; Greco et al., 2010).

Finally, the results are in the same category of Amazon Reviews- Electronics. This category is large and diverse. However, it does not necessarily reflect customer behaviour in other product domains.

6.2 Future Work

With these limitations in mind, we can see several future work directions. The first direction is related to the enlargement of datasets. In the future, the framework can be further tested on other Amazon Reviews categories such as Home, Beauty, Clothing, Sports or Books. This would allow us to see the stability of the results across different product domains.

The second direction involves using more detailed product attributes. For Electronics: technical specifications, warranty information, product dimensions, compatibility, battery life, material, energy consumption, delivery options, or return policy may be useful. These features could make classification more meaningful and less dependent on indirect indicators like description_length or details_count.

The third area concerns the application of Natural Language Processing (NLP). Reviews include not only ratings but also textual feedback. Future studies could extract even more preference indicators with sentiment analysis, topic modelling, or text embeddings. But these features should be used carefully to prevent data leakage.

The fourth direction concerns the transition from product-level classification to the personalised recommendation setting. In the future, user-level features such as user rating history, preferred product categories, or previous interactions can be added. This allows us to go from the question "is a product recommended in general?" to the question "is a product recommended for a specific user?". Future work could extend the current product-level classification framework toward personalised recommendation models, including neural collaborative filtering approaches (He et al., 2017).

The fifth direction concerns extending the set of models. In future work, LP-based UTADIS can be compared not only with Logistic Regression and Weighted Neural Network, but also with Random Forest, Gradient Boosting, XGBoost, LightGBM or special recommendation system models.

The sixth direction is connected to the further development of UTADIS implementation. UTADIS II or UTADIS III can be implemented, different numbers of breakpoints for marginal utility functions, different margin parameters and different constraints for maximum criterion utility can be tested.

In general, future work can extend this thesis to wider data, complex models, deeper personalisation, and stronger explainability. The next step is to check the robustness of the proposed framework with respect to category, features, target definition and modelling approach.

7. Conclusion

The purpose of this thesis was to develop and evaluate an AI module for the customer preferences system. The main idea of the work was to use customer reviews and product meta data for product-level classification. In the final formulation, each product was classified as either recommended or not recommended. We chose this formulation because it better aligns with the practical logic of the recommendation decision than predicting the exact rating on a scale of 1 to 5.

In this work, the source data was transformed from a review-level classification dataset to a product-level classification dataset. Customer preference signals were derived from ratings, and then the positive_ratio was computed for every product. A threshold of 0.85 was used to build a recommended binary target. This allowed us to move from individual subjective reviews to a more stable aggregate product-level assessment.

The main reference method in the thesis was UTADIS based on LP. This method was chosen as it belongs to the MCDA field and is appropriate for Multicriteria Sorting tasks. In this paper, UTADIS was used to construct an additive utility function, in which each criterion contributes a certain amount to the final utility of the product. Then the utility score was compared with the classification threshold, and the product was classified as a recommended or non-recommended class.

To evaluate UTADIS, it was compared against benchmark models, such as the Majority baseline, Logistic Regression, Neural Network and Weighted Neural Network. The majority baseline showed that the class imbalance can be misleading in terms of ordinary accuracy. The machine learning benchmark result for Logistic Regression was consistent. The standard Neural Network achieved high accuracy but suffered from majority-class bias. Therefore, the Weighted

Neural Network was added, which used sample weights and balanced the recognition of both classes better.

The Weighted Neural Network showed the best results in terms of balanced accuracy and macro F1-score. It means it was the best at coping with the task of simultaneously recognising recommended and non-recommended products. This result is reasonable because the Neural Network is a more flexible model and can explain more complex dependencies between features and the target variable.

LP-based UTADIS did not outperform Weighted Neural Network and Logistic Regression in predictive metrics. However, its result was better than Majority baseline in balanced metrics. This means that UTADIS potentially can generate an interpretable classification structure instead of just mimicking the most popular class. Meanwhile, the primary strength of UTADIS remained its ability to explain classification with a utility-based structure.

The overall model comparison reveals an important trade-off between predictive performance and interpretability. The Weighted Neural Network was the best in terms of prediction. Logistic Regression showed a stable benchmark result. The UTADIS method, based on LP, was not as strong in numerical performance, but it offered a multicriteria structure which was more understandable. This means different methods can be useful for different purposes within the customer preference system.

The main contribution of the thesis is to demonstrate how customer reviews and product metadata can be combined to yield a meaningful product-level classification task. It introduced how to derive customer preference signals from raw ratings, how to aggregate data at the product level, how to build a recommended target, and how to compare different classification methods within the

same experimental framework. This allows us to formulate recommendations not just as a machine learning problem but also as a multicriteria decision problem.

Thus, the thesis confirms that LP-based UTADIS can be used as an interpretable method for analysing customer preferences, although it is inferior in predictive performance compared to more flexible machine learning models. Meanwhile, the Weighted Neural Network produced the best-balanced results and can be considered the most effective predictive benchmark. The final conclusion is not that one method is the best for every case, but that every method has its place: machine learning models are more appropriate for prediction and UTADIS for transparent multicriteria explanation.

Overall, the work shows that a balance between quality classification and explainability is needed when building a customer preferences system. For practical decision support, it is important not only to know if a product is recommended, but also which product-level criteria are related to this decision. Therefore, a combination of AI-based methods and the MCDA approach can be a promising direction for further development of systems of customer preferences.

8. Bibliography

1. Almeida-Dias, J., Figueira, J. R., & Roy, B. (2010). ELECTRE TRI-C: A multiple criteria sorting method based on characteristic reference actions. *European Journal of Operational Research, 204*(3), 565–580. https://doi.org/10.1016/j.ejor.2009.10.018
2. Belahcène, K., Mousseau, V., Ouerdane, W., Pirlot, M., & Sobrie, O. (2024). A guided tour of multiple criteria sorting models and methods. *Annals of Operations Research, 343*, 785–845. https://doi.org/10.1007/s10479-024-06278-w
3. Greco, S., Mousseau, V., & Słowiński, R. (2010). Multiple criteria sorting with a set of additive value functions. *European Journal of Operational Research, 207*(3), 1455–1470. https://doi.org/10.1016/j.ejor.2010.05.021
4. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 173–182). https://doi.org/10.1145/3038912.3052569
5. Hossain, M. J., Sarker, M. R., Tazim, M. M. H., Talha, A., Chowdhury, R. I., & Miah, M. A. (2026). E-commerce conversion rates and customer satisfaction through neural collaborative filtering recommendations. In S. Dutta et al. (Eds.), *Proceedings of International Conference on Computational Intelligence and Information Retrieval* (pp. 401–411). *Lecture Notes in Networks and Systems, 1617*. Springer. https://doi.org/10.1007/978-3-032-04539-3_29
6. Hou, Y., Li, J., He, Z., Yan, A., Chen, X., & McAuley, J. (2024). Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*. https://arxiv.org/abs/2403.03952

7. Jacquet-Lagrèze, E., & Siskos, Y. (1982). Assessing a set of additive utility functions for multicriteria decision-making, the UTA method. *European Journal of Operational Research, 10*(2), 151–164. https://doi.org/10.1016/0377-2217(82)90155-2
8. Kadziński, M., Wójcik, M., & Ghaderi, M. (2025). From investigation of expressiveness and robustness to a comprehensive value-based framework for multiple criteria sorting problems. *Omega, 131*, 103203. https://doi.org/10.1016/j.omega.2024.103203
9. Liu, J., Kadziński, M., Liao, X., & Mao, X. (2021). Data-driven preference learning methods for value-driven multiple criteria sorting with interacting criteria. *INFORMS Journal on Computing, 33*(2), 586–606. https://doi.org/10.1287/ijoc.2020.0977
10. McAuley Lab. (2023). *Amazon Reviews 2023* [Data set]. University of California San Diego. https://amazon-reviews-2023.github.io/
11. Sobrie, O., Mousseau, V., & Pirlot, M. (2019). Learning monotone preferences using a majority rule sorting model. *International Transactions in Operational Research, 26*(5), 1786–1809. https://doi.org/10.1111/itor.12512
12. Zopounidis, C., & Doumpos, M. (1999). Business failure prediction using the UTADIS multicriteria analysis method. *Journal of the Operational Research Society, 50*(11), 1138–1148. https://doi.org/10.1057/palgrave.jors.2600817