Radboud University Nijmegen

Faculty of Science

Optimizing Local Product Performance of Fast-Moving Consumer Goods

a Case Study

Thesis MSc Computing Science
Specialization: Data Science

Author:
Gijs Dekkers BSc

Supervisor:
Prof. dr. ir. Arjen de Vries

External supervisor:
Jeroen van Dijk MSc

Second reader:
Prof. dr. ir. Djoerd Hiemstra

July 28, 2021

Abstract

Retailers in the Fast-Moving Consumer Good (FMCG) industry often experience sub-optimal product performance, which results in reduced sales volumes. This can be partly attributed to incorrect planogram compliance or an sub-optimal assortment selection. For incorrect planogram compliance, we propose a logically consistent market share model that detects what product categories are underperforming on a store level. Also, we propose a recommendation system that recommends products to a specific store that may potentially sell well in that store. In both methods, we use the notion that similar stores sell similar products and have a comparable market share. To determine these similar store clusters, we use a K-nearest neighbor algorithm that selects the nearest neighbors in terms of geographical location, total sales volume, assortment selection and number of unique products. For planogram compliance, we find that we can accurately predict market shares for a majority of stores. However, for some stores we observe high deviations between actual and expected (predicted) market shares. Such deviations are strong indications for product underperformance or overperformance in a specific store. Zooming in on our recommendation system, we observe that we are able to achieve high prediction performance as well. At the same time, we also find high coverage and diversity scores. This indicates that our recommendation model recommends products from the complete assortment set and recommendations are tailored to the characteristics of the target store. Combining the results of these two methods, we provide a complete monitoring tool to optimize product performance.

Contents

1 Introduction

On a daily basis, people spend a significant amount of money on items purchased for consumption. These items are typically consumed rapidly, sold in large quantities, and have a relatively low price. Examples of these item categories include food, beverages, toiletries, cosmetics, and cleaning products and are better known under the title Fast-Moving Consumer Goods (FMCGs) [33]. The FMCG industry had a market size of 10.020 billion dollars worldwide in 2017 and is expected to reach a market size of 15.361,8 billion dollars in 2025, which indicates the importance and influence of the industry [7]. With many companies involved and the rise of digital competition [16], retailers and manufacturers in the FMCG industry need to ensure that their products are sold effectively and with the largest sales volume possible to be successful. For large retailers, store plans to achieve this effectivity are usually created by corporate headquarters and executed in-store. Store plans include a wide variety of complex decisions, for example assortment selection, in-store location of products (planogram), inventory levels and product pricing [24]. Perfect store execution is therefore a complex task and in practice a sub-optimal store execution is often observed. Wal-Mart, for example, estimated in 2014 a 3 billion dollar opportunity only on Out-Of-Shelf issues [22], which indicates the magnitude of possible execution errors. A sub-optimal assortment selection may be another cause of sub-optimal sales, where both the absence or presence of a specific product in a store can influence the total sales volume. Especially with the use of advanced analytics and machine learning techniques, available data about sales volumes in these stores can provide insights and solutions in this matter.

In our research, we focus on two possible causes of sub-optimal sales volumes and more specifically methods for automated identification of situations in which these causes occur. With the use of a data set that contains all products of one specific product category of FMCGs in the Netherlands sold at a specific retailer, we distinguish between *(i)* incorrect planogram compliance and *(ii)* sub-optimal assortments.

Planogram compliance is a term to describe in-store execution, based on the planogram of a store. A planogram is a diagram that directs how shelvings and store displays should look and how products should be placed [39]. Planogram compliance involves factors as merchandising, shelving, and product displays. Both brands and retailers invest a lot of time and money to improve their retail execution and product placement strategies in an effort to maximize their total sales volume. However, in practice still approximately 40% of manufacturers and retailers do not have a solution to measure planogram compliance rates effectively [64]. Only 3 out of 10 retailers measure their store-level compliance frequently enough to catch in-store execution problems and make corrections early [55]. This is undesirable since the identification of planogram compliance rates is fundamental to make increased planogram compliance possible. Research has indicated that 100% planogram compliance could on average result in an uplift in sales volume of 7.8% [64].

A sub-optimal assortment selection is another factor that could decrease total sales volumes. Retailers have to consider a variety of options concerning their assortment strategy, which makes it a complex task. For example, we can distinguish between deep and wide assortment strategies [1]. Deep assortment strategies aim for a large number of products within a particular product category, whereas wide assortment strategies aim to offer a large variety of different product categories. Also, stores can adapt their assortment to the location of the specific store, offering products that sell well in a specific region [17]. Multi-store retailers, in general, tend to combine both common and local assortments. This assortment selection needs to be well-balanced to optimize total sales volumes. When products in the same assortment are highly similar to each other, sales volumes may also be sub-optimal, due to inventory cannibalization effects [58], where the sales volume of one product suffers from the presence of another product in the same store. This shows that both the absence and presence of products in a specific assortment may have a negative influence on the sales volume of its respective store.

In our research, our main focus for recommendations and identification of underperforming products is based on store similarity. For example, the purchasing behavior of customers in one retail location is often similar to the purchasing behavior of customers in neighboring retail locations. This can be illustrated by the fact that there is an increasing interest in buying local products over the last couple of years, especially in Western society [54]. Also, competition between retailers is highly localized where margins are very sensitive to the presence of retailers within a few kilometers [59]. From the perspective of multi-store retailers, in many cases geographical location is the most important source of product differentiation between stores [3]. Therefore, to identify both sub-optimal assortments and incorrect planogram compliance, we use the notion that stores within close geographical proximity are expected to sell similar products and have a similar market share distribution. However, closeness can also be defined in terms of overlapping assortments, total sales volume, or the number of unique products.

Both our methods rely on the notion of store similarity, but this notion is used in different ways for both problems. First, for the identification of incorrect planogram compliance, we cluster similar stores together, based on characteristics as geographical location, assortment selection, total sales volume, and the number of unique products. Subsequently, we determine the differences in market shares between the cluster and one specific store. We call products 'underperforming' if they are sold less than expected compared to the corresponding cluster. Since the products are already offered in-store, this underperformance can likely be attributed to incorrect planogram compliance. More generally, we provide a tool that monitors product performance. This identification is important for retailers since they can try to minimize this underperformance by improving planogram compliance or even implement changes in their planograms. To achieve this, we propose a logically consistent market share model [48] based on a nearest neighbor algorithm that determines a set of look-a-like stores, based on the earlier mentioned characteristics. With this set, we determine the expected market shares of products in a specific store and compare this with the actual market shares of the target store to determine product performance.

Secondly, for sub-optimal assortments, we want to recommend a variety of product candidates that are not yet sold in one store but may potentially sell well when added to the assortment. The predicted market shares are based on the respective market shares of these products in similar other stores. We achieve this with the use of recommendation systems [52]. For this matter specifically, we use a user-based collaborative filtering (CF) method [57]. User-based CF is used in many different domains, including video, music, and retail industries. Well-known companies using CF are for example Netflix and Amazon. These companies recommend items for one specific user based on the behavior and preferences of other users. In the case of Netflix for example, these items correspond with movies offered by their streaming service. For our application specifically, we propose a model that recommends items based on the market shares of products sold in other stores. Here we consider stores as our 'users' and products as our 'items'.

Using the results of this research, retailers can use recommendations and identifications of underperforming products to adapt their assortment strategy, improve planogram compliance or adapt planograms. This can be done effectively and at relatively low costs, without the need for expensive equipment. The way retailers deal with underperformance in practice, falls outside the scope of this research since our data does not provide insights into the specific cause of underperformance.

2 Literature review

In this section, we first discuss previous research related to currently existing methods that can identify incorrect planogram compliance and sub-optimal assortment and why our approach is beneficial compared to these solutions. Subsequently, we zoom in on specific existing methods that we use in our solution concerning both identifying underperforming products as well as recommending new products. Finally, we discuss existing methods to validate both models in an offline manner, since it is unfeasible for our research to make real in-store adaptations with the use of field sales managers. This section is structured according to the general structure of this paper, as illustrated in figure 1.

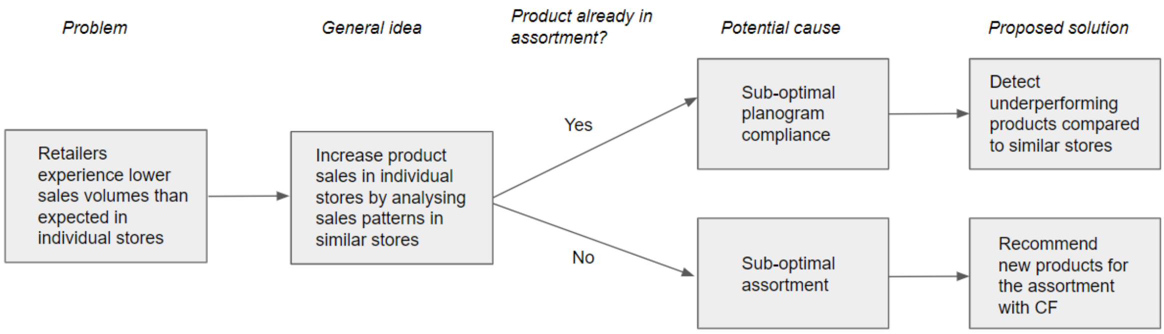


Figure 1: Structure of paper

2.1 Planogram compliance

Zooming in on planogram compliance, we first discuss what planogram compliance is and what important factors contribute to incorrect planogram compliance. Also, we discuss existing techniques that are currently in production or development regarding this matter. Subsequently, we show existing algorithms that are suitable for the identification of underperforming products, where we can use our notion of store similarity. Finally, we discuss existing validation methods that enable us to evaluate the performance of our solution in an offline setting.

2.1.1 Need for planogram compliance

According to Nature Insight, planogram compliance is used to describe the compliance of in-store execution, merchandising, shelving or displays with a planogram [39]. A planogram is a diagram that directs how shelvings and store displays should look and how products should be placed. Planogram compliance is a term for having displays set up according to these planograms. We can derive from this definition that planograms involve a variety of aspects, which all need to be monitored regularly to comply with these planograms correctly. The ultimate goal of a retailer using planograms is to have the right product, in the right place, in the right quantity, at the right price and time [64].

Planogram compliance is essential for retailers, because of multiple reasons. Compliance is a key factor to measure the effectiveness of planograms. Non-compliance is expected to lead to reduced sales and out-of-stock issues due to incorrect usage of shelf space [64]. It is hard for analysts to determine the effect of replacement, changed prices and other adaptations solely based on incomplete planograms. Accurately adhering to planograms is therefore essential and can on average lead to a potential sales uplift of 7.8%. Planogram compliance becomes even more important for products that have a seasonal component or that are sold in a promotional period. Retailers of FMCGs often deal with these types of products.

Maintaining planogram compliance turns out to be a challenging task in practice. Research showed for example that after studying key grocery retailers in the USA for 11 merchandise categories, only 70% of the products in the planogram were actually displayed in-store [64]. Another study showed that in general shelf-level planogram compliance levels typically fall below 50%. This indicates that planogram compliance levels in many retailers are far from sufficient and that the potential gain by improving planogram compliance is large.

Planning a store involves a variety of factors that makes correct compliance a complex task. Examples of these factors include among others product placement, placement of promotions, pricing, inventory levels, shelf tags, and size and location of merchandise categories [64]. However, other factors that are not directly related to the planograms themselves also play a role in non-compliance. For large retailers, corporate headquarters usually make planograms for store clusters, instead of a single store individually. This can result in sub-optimal planograms since the planograms are not fully tailored to the characteristics of one specific store and the demographics in the respective region. Moreover, there is little opportunity for store personnel and other partners to share information about their store planning and execution, which disables retailers to incorporate this information while designing their planograms. Lack of workforce and personnel training can also result in sub-optimal compliance. Even for stores that comply with their planograms perfectly, ongoing maintenance is challenging, since FMCGs typically are sold at a fast pace and need to be re-stocked daily.

Multiple methods to monitor planogram compliance exist. Traditional methods involve store personnel or external field sales managers to manually check in-store execution and product availability with the human eye. These approaches are labor intensive, time-consuming and cost-ineffective [40]. Also, manual approaches are subject to high error rates which reduce the effectiveness of planogram compliance checking. Automated methods are therefore preferred.

Automated methods for planogram compliance checking regularly involve computer-aided visual systems [46, 40]. By definition, this means that cameras need to be installed in the corresponding retail locations. Although these solutions are time efficient, they come with the downside of high costs due to the installation and maintenance of these systems. Also, these systems can only work with a minimum amount of high-quality training data, which is often hard to acquire [65]. One existing visual method for planogram compliance checking is able to measure partial or complete occupancy, product placement and product count [53]. These are highly important factors involved in planogram compliance. At the same time, the method remains user-friendly and robust. Deep learning approaches based on visual compliance also exist and make use of robotics to acquire suitable training data [15]. Although the results were promising, the approach is expensive and not yet feasible to use in a real-world setting, for example due to the small number of identifiable product items. Non-visual methods often involve the use of RFID sensors [14, 49, 23]. Although these methods perform relatively well, they suffer from high costs due to sensor installation and the time-consuming task of attaching all sensors to every single product [53].

2.1.2 Identifying underperforming products

As a result of planogram non-compliance, products can be underperforming in a specific retail location. A product is underperforming when its actual sales volume is significantly lower than the expected sales volume. Actual sales volumes can be measured for one retail location by simply analyzing historical sales data. However, expected sales volumes need to be accurately forecasted to be able to measure underperformance of products in a specific store.

One concept that we can use to measure expected sales in one retail location is 'like-for-like' sales, also referred to as comparable store sales or same-store sales [67]. Like-for-like sales is a method of financial analysis that is used to identify which of a company's products, divisions, or stores are contributing to its growth and which are lagging behind. This analysis is typically based on stores with similar characteristics. Where it is often used to measure growth of newly opened stores compared to existing stores in the region, it can alternatively be used to forecast the expected sales using a specific cluster of similar stores. To determine like-for-like sales and the expected sales volume of individual products, it is necessary to identify these similar stores first.

Different categories of indicators can be used to determine store similarity. These can be generally divided into store, customer, local area, and location characteristics [12]. Store characteristics include revenue, profitability, and size of the specific retail store. On a customer level, demographics and purchase history are important factors [12]. Local area and location characteristics include factors as local competition, demographics of a certain region, and the geographic location. However, data needs to be available about all these categories to be able to cluster on these features.

Clustering can be executed in an unsupervised manner. Prior research has primarily focused on customer segmentation instead of store segmentation. However, for the purpose of retail store clustering, one case study experimented with 4 different clustering algorithms: Hierarchical clustering, Self Organizing Maps, Gaussian Mixture Matrix, and Fuzzy C-means respectively [2]. These algorithms could address the limitations of traditional methods as agglomerative clustering and K-means clustering. The research in [2] showed that the performance of a clustering algorithm is highly dependent on the specific case. Every clustering algorithm performed differently for different markets. In other words, it makes sense to experiment with multiple clustering techniques to determine the one that is best performing for this specific problem.

For retail store clustering and determining store similarity, unsupervised clustering approaches as K-means may not be suitable. The number of clusters in K-means for example needs to be defined arbitrarily and this number is typically not known for this type of problem [50]. Also, the number of data points can differ highly across different clusters. To generate clusters with a fixed number of data points, we can use methods as the K-nearest neighbor algorithm [20]. This method is used for supervised classification or regression, where K can be arbitrarily specified. These nearest neighbors are determined using a pre-specified distance metric, for example the Euclidean distance, the Minkowsky distance or cosine similarity [29]. Every data point is assigned to a set of K neighbors. Including the initial data point, we can interpret this set as a cluster of $K+1$ data points. However, strictly this is not a clustering method, since this method allows data points to be the nearest neighbor of multiple other data points. Most clustering methods usually assign every data point to only one cluster and do not have overlap with other clusters.

One disadvantage of the K-nearest neighbor algorithm is that it is distance-based [20]. As a result, every attribute in the data is weighted equally and contributes equally to the classification or regression after normalization. In practice, this is often undesirable, since attributes generally are not of equal value for the specific problem. Noisy and irrelevant attributes may then influence the neighborhood search to the same degree as highly relevant attributes. It is important to weigh attributes to determine the degree of relevance. Weighting can be done manually by intuition and experiments. However, solutions exist to determine the optimal set of weights semi-automatically. Examples of methods include the Chi-Squared statistical test [61], Gain Ratio [47] or more advanced techniques such as feature importances determined using Random Forest that learn the optimal set of weights [8, 68]. This has the advantage that weights are always tailored to the data concerning the prediction problem without human interference.

2.1.3 Evaluating identification of underperforming products

After assigning every store to clusters of K stores, where K can be arbitrarily selected, the quality of the clusters needs to be evaluated. For our problem, a cluster of stores is considered an indicator of the expected market shares in a specific target store. We can leverage this cluster to predict market shares of each product for the target store. This can be done for example by taking the median or mean market shares of the cluster as our expected market shares in the target store. If the expected sales volume is close to the actual sales volume, we can consider the cluster as a good reference compared to the actual sales volume of our target store. High similarity between expected and actual market shares indicates that the prediction performance of our model is high and vice versa. However, high similarity also indicates that the market shares in the target store are as expected. This may indicate also that planogram compliance is more strictly maintained. Therefore, high prediction performance is not necessarily the main importance.

In the previous paragraph, we mentioned that the predicted market share per product needs to be close to the actual market share per product. However, it is not straightforward how this closeness is defined. For solutions in the retail industry, it is preferred to use performance metrics that are used in the industry and at the same time can provide good insights into the cluster quality for our solution. The forecast accuracy metric can provide a quantitative solution to this. Considering sales volumes instead of market shares, the forecast accuracy metric is normally used to determine the difference between the expected sales volume and the actual sales volume per product within a single store [30]. The metric is often used to optimize inventory levels. However, we can alternatively use this metric to compare the actual sales volumes in one store with the expected sales volume of that same store. Often used metrics are the Mean Absolute Percentage Error (MAPE) or alternatively Weighted Absolute Percentage Error (WAPE) when dealing with low sales volumes [28]. The expected sales volume in this case is the median or mean actual sales volume of the store cluster. This connects to the earlier mentioned concept of comparable store sales. As mentioned before shortly, a 100% similarity between the expected sales volumes and the actual sales volumes shows that every product market share corresponds with the expected market share (high prediction performance). This means in practice that no products are underperforming. As a result, there is no need to improve store execution on this matter. On the other hand, low similarity values can indicate that stores are too different to accurately draw conclusions about underperformance of specific products. Therefore, it makes sense to reject stores locally based on a minimum forecast accuracy threshold, which can be arbitrarily defined [62]. In practice, this means that from the initially K selected neighbors, only K-N neighbors remain for the prediction of the expected market share in a specific store. Because error rates decrease when reject rates increase with this dynamic number of neighbors, it is important to find the optimal balance between error rates and rejection rates [62]. Although our focus is on the detection of product underperformance, we can use the same method for the detection of overperforming products. This makes our solution a generic monitoring tool.

Other evaluation metrics often used for measuring the prediction performance of regression problems in general are the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) [13]. The MAE metric measures the average magnitude of error between the actual values and the predicted values. In our study, this error corresponds with the difference between the expected and the actual market shares calculated over the entire product set. The RMSE metric is calculated similarly but does involve a quadratic scoring rule. This means that large errors have a relatively high weight. This is useful when large errors are particularly undesirable. However, when the magnitude of error is not of high importance, MAE might be more suitable. Both metrics do not take the direction of the error into account.

Besides accurate predictions, predictions should preferably be made with high confidence. Especially with a dynamic number of neighbors, confidence generally decreases when the number of neighbors decreases [26]. Therefore, instead of making precise value predictions, it makes sense to make interval predictions instead. When confidence decreases, these confidence intervals become larger. This enables us to determine underperforming products given the confidence of the neighborhood [26].

2.2 Sub-optimal assortments

Focusing on sub-optimal assortments, we first discuss the need for optimal assortment planning and the challenges that are common in practice. We discuss existing frameworks and factors that are commonly used while planning the optimal assortment mix. Subsequently, we show existing algorithms that are suitable for recommendations. Finally, we discuss different methods that can be used for the evaluation of recommendations, both in terms of prediction performance and quality of recommendations.

2.2.1 Need for optimal assortment planning

Product assortment strategy is a central but complex task for a retailer [11, 43]. An assortment enables a retailer to differentiate from competitors. For customers, the assortment is generally more important than the product pricing for the selection of their preferred grocery store [9]. During assortment selection, a retailer needs to decide what product categories to offer and determine the number of Stock Keeping Units (SKUs) within a specific product category. At the same time, retailers are often constrained by the amount of store and shelf space and the amount of money that they can invest in their inventory [42]. In practice, this means that a retailer needs to find the optimal trade-off between product variety, product depth, and service level. Product variety indicates the number of different product categories offered, whereas product depth indicates the number of SKUs within a single category. Service level indicates the number of options for one item specifically (e.g. different brands). A good trade-off satisfies the need of customers because the right number of products is offered at the right time. Not providing the expected assortment may at the same time cause losses in both current and future sales [42]. A product that is not offered in a specific store could be a valuable addition to the assortment. On the other hand, the presence of certain products in an assortment can cause sub-optimal sales volumes as well, for example due to inventory cannibalization [58]. This again illustrates the complexity of assortment planning. A well-known paper provides a conceptual framework that considers 3 main factors that are involved with Product Assortment Planning (PAP), respectively customer preferences, retailer constraints, and environmental factors [42].

First, customer preference influences assortment planning by retailers. For a retailer, it is important to adhere to customer preferences to avoid them switching to competitors or have them buy fewer products than initially intended. As a starting point, it makes sense to identify brand preferences for customers in a specific region. At the same time, customers want to be flexible and be able to switch to other brands as well [34]. This can be illustrated by the fact that customer preference is unstable and that preferences may change over time, dependent on the circumstances [6]. This indicates that it is good for a retailer to offer a wide variety of products. On the other hand, too many products negatively affect customers' buying behavior [11]. Regional brand preferences may also be important for the selection of an optimal assortment. It has been shown that it is beneficial to offer a mix between common assortments and local assortments [17]. Common assortments are offered in a relatively large region (e.g. a country), whereas local assortments focus on products that are manufactured within a relatively small distance from the store (e.g. a province or city). Offering regional products increases the loyalty towards a store and has a direct impact on the image of a retailer [41].

A second factor that has a direct impact on assortment planning is retailer constraints [42]. These are typically physical characteristics of a store that limit the possibilities of a retailer during assortment selection. For example, store space is essentially fixed, because expanding store space is often not possible. Also, shelf space is limited, which means that decisions need to be made about what products to offer and in what quantity. Selecting the assortment variety, depth, and service level can become an increasingly complex task because of this matter. Models exist that optimize the allocation of certain products. Common approaches use heuristics to allocate shelf space in direct proportion to their sales [18]. In terms of recommending new products, it is therefore important not to only recommend products, but also estimate the quantity of their respective sales.

The third factor identified are environmental factors including competition-related assortment trends, changing environmental conditions, and shifting consumer profiles and lifestyle trends [42]. Competition is one example where retailers can learn what products to add to their assortment. Competition tends to be highly localized and retailers usually have to compete with stores in their direct neighborhood [59]. Geographical locations of stores are the most important source of product differentiation [3]. Well-performing products are therefore likely to be sold at a local competitor as well. Changing lifestyle trends are also an important factor that stores need to adhere to by adaptation of their assortment. A clear example of changing lifestyle trends is the growing demand for healthy and natural food. Natural food retailers, therefore, experience a large increase in sales volume per year [63]. Retailers need to be aware of these trends. Recommendation systems can help retailers to give insights into potential well-selling products that are not yet offered in their assortment.

Assortment planning proceeds through both long-term and short-term strategic decisions, including multiple interrelated steps [35]. In the FMCG industry, long-term strategic decisions include for example dividing a retailers' assortment into multiple merchandise categories and subcategories. Short-term decisions are more focused on demand forecasting and the resulting sales and margins. These decisions are complex, especially because of the interrelationships between these decisions [42]. Therefore, it is necessary that strategic decisions are evaluated regularly and that product sales volumes and margins are monitored. The complexity of assortment planning results in the need for specialized teams that semi-manually determine optimal assortment mixes [42]. This is time-consuming and therefore it would be beneficial to provide automated tools to help these teams with their decision-making process. Recommendation systems can be used for this problem as an automatic approach to recommend new but relevant items. These models determine this set of items based on similarities between the target store and other stores, both taking current assortment mixes and sales volumes of products into account.

2.2.2 Creating recommendations

From section 2.4, we see that recommending new products for an existing assortment is useful to propose sets of relevant items that may result in higher sales volumes. For these recommendations, many approaches exist. One simple approach is a Market Basket Analysis (MBA), where the selection of new products is based on association rules [10]. These association rules indicate whether for example product A is often sold together with product B. If this is the case, this could be a relevant recommendation if product B is not sold in a specific store. However, this approach becomes infeasible very quickly with large numbers of products due to the computational overhead that this method requires. Therefore, for a typical setting in the FMCG industry, there is a need for methods that can deal with large quantities of stores and products.

Recommendation engines are frequently based on interactions between users and items. These systems have the goal the determine well-matched user-item pairs. We can distinguish between two types of recommendation systems, collaborative filtering (CF) approaches and content-based filtering approaches respectively [44]. These approaches can deal with large amounts of data, where a typical data set is sparse [57]. A content-based filtering system selects items based on the correlation between the content of the items and the user's preferences [60]. This method has been proven to be effective for different types of recommendation problems. However, contextual attributes about both users and items need to be available to be usable.

A collaborative filtering (CF) system chooses items based on the correlation between people with similar preferences [60]. In a CF system, a user-item matrix is considered. This matrix contains all user-item interactions. These interactions can correspond with for example an explicit rating or a transaction. CF systems are widely and successfully used for recommendation, for example at large companies as Netflix and Amazon. For our application, stores can be considered as users, whereas products can be considered as items. In that case, we could interpret sales data or market shares as our explicit rating of a store-product pair. Within the CF domain, different approaches exist. We can generally distinguish between neighborhood-based CF and model-based CF [44]. Neighborhood-based CF determines a subset of appropriate users, based on a similarity measure between the target user and all other users [27]. The top number of neighbors, also called the neighborhood, determines the prediction of the target user. Here, each user is weighted based on the similarity between the user and the target user. In a model-based CF recommendation system, a user-item matrix is generally decomposed in a latent user and item component, using factorization techniques [36]. After decomposing the user-item matrix in multiple latent matrices, explicit user-item interactions are reconstructed. At the same time, a rating is estimated for non-existing user-item interactions. These ratings can be used for recommendations. A special type of model-based recommendation systems are factorization machines [51]. Factorization machines combine the advantages of Support Vector Machines with factorization models and scale well for large matrices. Dealing with large matrices, this method generally tends to achieve high performance in relatively low training times. However, factorization machines and other model-based recommendation systems come with the downside that recommendations cannot be attributed to individual data points or a neighborhood. This means that explicit store clusters cannot easily be defined with this approach. Considering both content-based and CF recommendation systems, our focus will be mainly on CF systems.

2.2.3 Evaluating recommendations

After recommending new items for specific assortments, the quality of these recommendations has to be evaluated in an offline manner. First, predictions need to be accurate. This means that we need a measure for the prediction performance. These metrics will be evaluated on a separate test set during development. Similar to the nearest neighbor model for planogram compliance, we can achieve this by using metrics as the Mean Average Error (MAE) and Root Mean Squared Error (RMSE).
Besides the prediction performance, a good method for measuring the quality of recommended items is by using precision and recall [19]. However, these metrics are based on the relevance of an item and the ratio of relevant items in the generated set of recommendations. Particularly for anonymized data, it is hard to determine whether a recommended item is relevant or not. Also, the definition of a relevant item is not pre-defined. This means that for our problem specifically, other methods are required.

Other metrics to assess the quality of generated recommendation include coverage, diversity, serendipity, and novelty [32]. Coverage indicates the portion of items a recommendation system is able to recommend on a test set. Generally, high coverage is desired because this will result in more inclusive recommendations. Diversity is a measure of the dissimilarity between different pairs of recommended items. This indicates that a recommendation system can lead to diverse results and not only recommends items within for example a specific product category. Novelty is a measure for the portion of new items that are recommended in comparison with the total set of items [69]. However, for our research, this is not important since we will restrict our final recommendations always to products that are not offered in a specific assortment. Serendipity indicates the number of surprising but relevant items in a recommended item set. This is again hard to measure due to the arbitrary definition of 'surprising' and that assessing relevance for anonymized data is not possible [37]. Summarising, we conclude that that especially the former two metrics, coverage and diversity, are useful and feasible for the evaluation of our recommendation system.

3 Methodology

In this section, we describe our applied methodology in detail. Firstly, we discuss methods that provide detailed insights into the characteristics of the data. Subsequently, we define our methodology to identify underperforming products using a logically consistent market share model. This includes a K-nearest neighbor prediction algorithm, a Random Forest weight optimization method, and various evaluation metrics. Finally, we define the methods used for recommending new products for a specific assortment. This includes a neighborhood-based collaborative filtering recommendation system and evaluation metrics that give insights into the quality of the recommended items and the prediction performance of the model.

3.1 Data characteristics

In the FMCG industry, goods typically show certain characteristics in their sales patterns, for example promotional sales peaks and seasonality. These characteristics might be influential for the selection and performance of our prediction model and need to be accurately quantified. In this section, we describe what techniques are used to identify and quantify different types of sales patterns that are regularly observed in the retail domain.

3.1.1 Promotions

Retailers often offer temporary price promotions to attract customers to its store. This encourages customers to buy regular-priced products as well [45]. On a product level, price promotions may result in a significantly larger sales volume for that specific product in the corresponding promotional weeks. The magnitude of this sales effect can differ across different product categories. We identify a promotional week based on its product price. In equation 1, $\mathbf{P}$ is a matrix that contains the product price of a specific product j ($j = 1, \cdots, N$) in week w ($w = 1, \cdots, W$), where W is equal to the number of weeks in a year and N is the number of products.

$$\mathbf{P} = \begin{pmatrix} p_{1,1} & \cdots & p_{1,W} \\ \vdots & \ddots & \vdots \\ p_{N,1} & \cdots & p_{N,W} \end{pmatrix} \tag{1}$$

Assuming that the number of non-promotional weeks is always larger than 50% of weeks in a year, we determine the non-promo price of every product j by taking the median value of all weekly product prices as defined in equation 2.

$$NP_j = Med\left(\left[p_{j,1}, \cdots, p_{j,W}\right]\right) \tag{2}$$

Subsequently, we determine promo week matrix $\mathbf{WP}$ by comparing the weekly product price P_{jw} with the non-promo price NP_j as follows in equation 3.

$$\mathbf{WP} = P_{jw} < NP_j \tag{3}$$

Here WP_{jw} is a binary variable where 0 corresponds with a non-promo week and 1 with a promo week respectively.

3.1.2 Seasonality

When dealing with FMCGs, a significant portion of products tend to show large seasonality patterns due to a variation of demand over different months [4]. Especially for recommendations, it is important to quantify the level of seasonality in a product set. For example, recommending ice cream in the winter months (while sales volumes are large in the summer) is expected to result in suboptimal sales volumes. If a large portion of products in our data is seasonal, we split our data in both a winter and a summer time period and make recommendations separately on these splits. We perform a seasonality analysis between the summer and winter months. For simplicity, we do not include spring and autumn separately. To achieve this, we first define two sales matrices $\mathbf{S}_{summer}$ and $\mathbf{S}_{winter}$ as described generically in equation 4, where j ($j = 1, \cdots, N$) indicates the sales volume of product j in store i ($i = 1, \cdots, M$).

$$\mathbf{S}_{season} = \begin{pmatrix} s_{11} & \cdots & s_{1N} \\ \vdots & \ddots & \vdots \\ s_{M1} & \cdots & s_{MN} \end{pmatrix} \tag{4a}$$

$$\text{season} = \{\text{season} \in [\text{winter, summer}]\} \tag{4b}$$

We only include products that are both sold in winter and summer. Here we assume that products that are only sold in winter or summer are seasonal by definition. Note that s_{ij} is 0 when a product is not included in the assortment of a specific store. Using the meteorological definitions of seasonal periods, $\mathbf{S}_{summer}$ is aggregated over April until September, whereas $\mathbf{S}_{winter}$ is aggregated over October until March. To determine if there is a significant difference between the summer and winter distribution of product sales, we perform a two-sample Kolmogorov-Smirnov test [56] between these empirical distributions, as defined in equation 5. To prevent the test from detecting short-term seasonal effects due to promotions, we only include non-promotion weeks as described in section 3.1.1.

$$KS_j = |S_{winter}(j) - S_{summer}(j)| \tag{5}$$

The Kolmogorov-Smirnov statistic vector **KS** yields a p-value for every product j in our sales matrix that indicates if product j has a significantly ($\alpha = 0.01$) higher or lower sales volume between winter and summer months.

3.2 Planogram compliance

In this subsection, we describe the applied methods for the identification of underperforming products. As mentioned before, we use the notion of store similarity to determine expected market shares in our target store and compare the actual market shares with the market shares of the store cluster. To determine the most similar stores, we make use of a K-Nearest Neighbor (KNN) algorithm. Since the KNN algorithm makes use of a distance function to determine the nearest neighbors and the fact that features are typically not equally contributing to the specified problem, features need to be weighted. To achieve this, we propose a Random Forest algorithm to weigh the features tailored to each specific store.

3.2.1 K-nearest neighbors

We use a KNN algorithm to determine a fixed set of K neighbors, where we set K equal to 10. In this paper, neighbors are defined as individual stores. This means that K indicates the size of the store cluster. KNN assumes that similar data points exist within close proximity [20]. This proximity is calculated based on the features used for every data point. We first define a store matrix **ST** in terms of its features as described in equation 6.

$$\mathbf{ST} = \begin{pmatrix} st_{11} & \cdots & st_{1F} \\ \vdots & \ddots & \vdots \\ st_{M1} & \cdots & st_{MF} \end{pmatrix} \tag{6}$$

This matrix $\mathbf{ST}$ describes a store i $(i = 1, \cdots, M)$ in terms of features f $(f = 1, \cdots, F)$, where M indicates the number of stores and F the number of features. There are different methods to measure proximity. We use the regularly used Euclidean distance measure $D_{x,y}$ to define the distance between stores [29] x and y as defined in equation 7. For clarity, we use example store numbers $x = 1$ and $y = 2$.

$$D_{1,2} = \sqrt{\sum_{f=1}^{F} (st_{1f} - st_{2f})^2} \tag{7}$$

The K nearest neighbors can now be selected based on this distance metric, where we want our distances to be minimal. Our final market share prediction $\hat{\mathbf{y}}_i$ can now be determined by taking the uniform mean of the stores in the cluster, where c $(c = 1, \cdots, K)$ indicates the store in the nearest neighbor cluster. This is defined in equation 8.

$$\hat{\mathbf{y}}_i = \frac{1}{K} \sum_{c=1}^{K} y_c \tag{8}$$

3.2.2 Absent product predictions and logical consistency

As described in equation 8, for a regression problem the uniform mean of the selected K nearest neighbors determines the final prediction $\hat{\mathbf{y}}_i$. However, this has the side effect that market shares can be predicted for products that are absent in the assortment of the target store if the product is present in the assortment of a store in its cluster. Therefore, we restrict our final prediction by setting these impossible market shares to 0. We first define an assortment vector $\mathbf{A}_i$ of binary values that indicate whether product j $(j = 1, \cdots, N)$ is in the assortment of store i $(i = 1, \cdots, M)$:

$$\mathbf{A}_i = \begin{bmatrix} a_{i1} & \cdots & a_{iN} \end{bmatrix} \tag{9}$$

M indicates the number of stores and N describes the maximum size of the product set. Subsequently, to ensure that absent products have a market share of 0, we update $\hat{\mathbf{y}}_i$ by taking the dot product of the assortment vector $\mathbf{A}_i$ with the market share predictions $\hat{\mathbf{y}}_i$ as described in equation 10, setting market shares of absent products equal to 0.

$$\hat{\mathbf{y}}_i \leftarrow \hat{\mathbf{y}}_i \cdot \mathbf{A}_i \tag{10}$$

Market shares are constraint by the fact that its total sums to 100%. Therefore market share predictions need to be logically consistent [48], i.e. $\sum_{i=1}^{N} \hat{y}_i = 1$. Simply removing market shares for absent products would result in $\sum_{i=1}^{N} \hat{y}_i < 1$. Therefore, we redistribute the removed market shares over all other market shares of products that are present in the assortment, based on the mean market shares vector $\overline{\mathbf{MS}}_i$ of the total set of stores excluding the target store i as defined in equation 11.

$$\overline{\mathbf{MS}}_i = \frac{1}{M-1} \sum_{c \neq i}^{M} \mathbf{y}_c \cdot \mathbf{A}_i \tag{11a}$$

$$\overline{MS}_{ij} \leftarrow \overline{MS}_{ij} + \frac{1 - \sum_{j=1}^{N} \overline{MS}_{ij}}{\sum_{j=1}^{N} a_{ij}} \tag{11b}$$

Again, only products that are present in the assortment are included. At the same time, logical consistency is maintained, i.e. $\sum_{j=1}^{N} \overline{MS}_{ij} = 1$. The portion in which we need to scale our market share vector back to 1 varies across all stores. We define our scale factor θ_i as follows in equation 12.

$$\theta_i = 1 - \sum_{j=1}^{N} \hat{y}_{ij} \tag{12}$$

We re-use this scale factor later for our prediction interval. Subsequently, we update $\hat{\mathbf{y}}_i$ as described in equation 13.

$$\hat{\mathbf{y}}_i \ \leftarrow \ \hat{\mathbf{y}}_i \ + \ \overline{\mathbf{MS}}_i \ \cdot \ \theta_i \tag{13}$$

After this transformation, $\hat{\mathbf{y}}_i$ only contains market share predictions for products that are present in the assortment of store i. At the same time, $\hat{\mathbf{y}}_i$ complies with the definition of a logical consistent market share model.

3.2.3 Weight optimization

Features in our store feature matrix **ST** usually do not equally contribute to the relevance of the neighborhood. However, after standardizing these features to the same range, irrelevant features might contribute to the same extent as relevant features. Therefore, it is necessary to weigh our store features, based on their contribution to the dependent variable. To achieve this, we weigh these features with a Random Forest algorithm [8, 68]. Feature weighting in Random Forest (for regression) is based on variance reduction [25]. Features that induce high variance reduction will have a higher internal node position in a decision tree, given a specific selection criterion. This selection criterion is usually the Mean Absolute Error (MAE). A precise definition is given in equation 14. Due to the multi-output setting of our problem, we average the variance reduction over all output values y_{pf}.

$$\text{MAE} = \frac{1}{F} \sum_{f=1}^{F} \frac{1}{P} \sum_{p=1}^{P} |y_{pf} - \mu_f| \tag{14}$$

In this definition, $\mathbf{y}_p$ corresponds with the model output of a training instance p ($p = 1, \cdots, P$), where P is the number of training instances. Additionally, μ_f is the mean of the training set for feature f ($f = 1, \cdots, F$), where F indicates the size of the feature set. MAE is the criterion of our node split in a decision tree. High differences between MAE before and after a split on a specific feature are preferred because this shows a high variance reduction pattern. This ultimately results in a weight matrix $\mathbf{W}$ that contains a weight w_{ij} for every feature st_{ij}. Our store feature matrix **ST** can now be updated as defined in equation 15.

$$\mathbf{ST} \leftarrow \mathbf{ST} \cdot \mathbf{W} \tag{15}$$

Note that this weight selection has a high influence on the resulting neighborhood determined with KNN. This is because for example, two stores may be close in terms of feature A but not in terms of feature B. Higher weights on feature B might therefore result in these stores not being clustered together anymore and vice versa.

3.2.4 Filter neighbors on forecast accuracy

The initial K-nearest neighbor algorithm comes with the disadvantage that the number of neighbors is fixed ($K = 10$). Ideally, we want to only include neighboring stores that have similar sales volumes and similar assortments to a certain degree. A fixed size of K therefore might yield a cluster of neighboring stores with different degrees of similarity between each neighbor and the target store. However, we want to only include stores that have a minimum level of similarity to ensure that we focus on comparable store sales. Therefore, we calculate the Mean Absolute Error (MAE) between the market shares of each store and the market shares of the target store i as defined in equation 16.

$$MAE_i = \frac{1}{N}\sum_{j=1}^{N} |y_{ij} - \hat{y}_{ij}| \tag{16}$$

Here y_{ij} indicates the actual market share of product j ($j = 1, \cdots, N$) in the target store i and $\hat{y}_{ij}$ the market share of the evaluated neighbor, which we interpret as the predicted market share of product j. Because we predict market shares, the absolute error automatically indicates the absolute percentage error, given that these market shares are logically consistent.

N describes the total number of products or product groups. Since we aggregate products on different aggregation levels, N will change over different aggregation levels as well. Therefore, if we want to maintain comparability between these aggregation levels, division by N leads to misleading results. As a solution, we come with a similar metric AE in equation 17 that does not normalize the forecasted absolute error by dividing by N.

$$AE_i = \sum_{j=1}^{N} |y_{ij} - \hat{y}_{ij}| \tag{17}$$

Note that both $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ are logically consistent [48], i.e. $\sum_{j=1}^{N} y_{ij} = 1$ and $\sum_{j=1}^{N} \hat{y}_{ij} = 1$. This means that the absolute error AE never exceeds a maximum value of $\sum_{j=1}^{N} y_{ij} + \sum_{j=1}^{N} \hat{y}_{ij} = 2$, which is independent from the size of N. We now use our AE to accept or reject a specific neighbor, based on a maximum error threshold θ_{AE}.

3.2.5 Prediction interval

With a dynamic number of neighbors used in our methodology, we need to cope with varying levels of uncertainty. Filtering neighbors on prediction performance concerning the target store results in higher performance. On the other hand, a small number of neighbors results in a lower confidence level associated with these market share predictions. As a result, we need to assess whether a deviating market share prediction falls within a specific prediction interval. Instead of making precise value predictions, we determine a prediction interval PI_{ij} for every market share prediction. The definition of our prediction interval is given in equation 18. Since we deal with small sample sizes ($K \leq 10$), we use a t-distribution instead of a standard normal distribution to determine our interval [21].

$$PI_{ij} = \hat{y}_{ij} \ \pm \ t\,\frac{\sigma(\mathbf{y}_{lj})}{\sqrt{L}} \cdot \ (\theta_i + 1) \tag{18}$$

Here, $\sigma(\mathbf{y}_{lj})$ indicates the standard deviation of all cluster market shares $\mathbf{y}_{lj}$, where l ($l = 1, \cdots, l = L$) and L is the number of neighbors in the filtered cluster for store i. Note that the prediction interval scales with the scale factor θ_i as defined previously in equation 12. The lower bound in this definition is clipped to 0 when market shares become negative. Finally, the t-distribution score t varies with the degrees of freedom, which is defined as $df = L - 1$, and confidence level α, which we set to $\alpha = 0.95$. This indicates that the prediction interval becomes wider when sample size L decreases. For our method, this results in the error margin becoming larger when dealing with higher amounts of uncertainty.

3.2.6 Evaluating detection of underperforming products

For our solution, it is important to include all stores except for the target store for the market share predictions during evaluation. Therefore, standard K-fold cross-validation is not suited, since a significant portion of stores will not be included in the training set in this case. Therefore, we alternatively use a Leave-One-Out cross validation (LOO-CV) technique. In that case, we train our model on all stores except for the target store and validate on the target store. We do this for all stores in our data set. This approach ensures that all stores are included.

We evaluate the performance of our model with a metric that is similar to the AE metric as described in section 3.2.4 and equation 17. However, $\hat{y}$ does not indicate the market shares of one single nearest neighbor, but the market share means of the entire filtered store cluster, which we define as the Absolute Cluster Error ACE. The definition of our ACE_i is given in equation 19.

$$ACE_i = \frac{1}{L} \sum_{l=1}^{L} \sum_{j=1}^{N} |y_{ij} - \hat{y}_{lj}| \tag{19}$$

L indicates the number of neighbors after filtering on the absolute error AE_i as described in equation 17. After determining the prediction performance of our model, we assess the number of products that fall outside our prediction interval. Products outside this interval will be considered as underperforming or overperforming. However, we are only interested in underperformance. We define our hit rate as the portion of products within the specified prediction interval divided by the number of products in the assortment as defined in equation 20.

$$\text{hitrate} = \frac{\text{predicted} \geq \text{PI} \mid\mid \text{predicted} \leq \text{PI}}{|\text{predicted}|} \tag{20}$$

Similarly, we define our underperformance ratio as the portion of products below the prediction interval divided by the number of products in the assortment as defined in equation 21.

$$\text{underperformance} = \frac{\text{predicted} < \text{PI}}{|\text{predicted}|} \tag{21}$$

3.3 Sub-optimal assortments

Missed opportunities for a retailer can also be caused by products that are not in the assortment of a store, but might potentially sell well considering its sales volume in similar stores. Therefore, in this section, we propose a user-based nearest neighbor collaborative filtering model to recommend new products for a specific assortment. This model considers both store-specific features and market shares of products, which we use to define similarity between stores. We evaluate our model both concerning prediction performance, as well as the quality of our recommendations in terms of coverage and diversity.

3.3.1 Store similarity

To develop an accurate recommendation model, we first need to define a suitable similarity metric. For our model, we include both store-specific features, such as longitude, latitude, total sales volume, and #SKU, as well as market shares of individual products. A definition of our store feature matrix **ST** is defined previously in equation 6. Additionally, we define our market share matrix $\mathbf{M}$ as follows in equation 22.

$$\mathbf{M} = \begin{pmatrix} m_{11} & \cdots & m_{1N} \\ \vdots & \ddots & \vdots \\ m_{M1} & \cdots & m_{MN} \end{pmatrix} \tag{22}$$

Here, m_{ij} indicates the market share of product j ($j = 1, \cdots, N$) for store i ($i = 1, \cdots, M$) with N the number of products and M the number of stores. Note that $m_{ij} = 0$ indicates that the product is not in the assortment and $m_{ij} \approx 0$ indicates that the product is in the assortment, but has a low market share that approximates 0. For both our store feature and market share vectors, we define store similarity by calculating the cosine similarity between stores as described in equation 23.

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{23}$$

We use the cosine similarity metric, since it ignores products that are not in the assortment, i.e. have a market share of 0. Both vector $\mathbf{A}$ and $\mathbf{B}$ can be substituted by either $\mathbf{ST}_i$ or $\mathbf{M}_i$. This ultimately results in two store similarity matrices **SSIM** and **MSIM** as described in equation 24 and 25.

$$\mathbf{SSIM} = \begin{pmatrix} ssim_{11} & \cdots & ssim_{1M} \\ \vdots & \ddots & \vdots \\ ssim_{M1} & \cdots & ssim_{MM} \end{pmatrix} \tag{24}$$

$$\mathbf{MSIM} = \begin{pmatrix} msim_{11} & \cdots & msim_{1M} \\ \vdots & \ddots & \vdots \\ msim_{M1} & \cdots & msim_{MM} \end{pmatrix} \tag{25}$$

Both matrices are of size $M \times M$, where $ssim_{ij}$ and $msim_{ij}$ indicate the similarity between store i and j concerning either store features or market shares respectively. To determine or final similarity coefficients, we weigh both similarity matrices. This results in combined similarity vector $\mathbf{CSIM}_i$ as defined in equation 26.

$$\mathbf{CSIM}_i = w_{0i} \cdot \mathbf{SSIM}_i + w_{1i} \cdot \mathbf{MSIM}_i \tag{26a}$$

$$\begin{gathered} w_{0i} + w_{1i} = 1 \\ 0.1 \leq w_{0i} \leq 0.9 \\ 0.1 \leq w_{1i} \leq 0.9 \end{gathered} \tag{26b}$$

$\mathbf{CSIM}_i$ indicates the weighted similarity between store feature similarity and market share similarity of store i concerning all other stores. Note that we restrict or model by setting the total sum of weights equal to 1. This means that if the contribution of **MSIM** increases, the contribution of **SSIM** automatically decreases and vice versa. We adapt our weights for each individual store by doing a grid search starting from $w_{0i} = 0.1$ and $w_{1i} = 0.9$, increasing w_{0i} and decreasing w_{1i} with a step size of 0.05. This approach will keep our search space feasible, considering the limitations in equation 26b. To make sure that both store features and market shares contribute to our combined similarity, weights are restricted to have a minimum value of 0.1. After our grid search, we set our weights for store i at the point where we reach the minimum of the absolute cluster error ACE_i between $\mathbf{y}_i$ and the predicted market shares $\hat{\mathbf{y}}_i$ (only considering $a_{ij} = 1$) as described in equation 19.

3.3.2 Collaborative filtering

After defining our similarity metric, we use these similarities to develop our collaborative filtering recommendation model. As described in section 2.2.2, we generally distinguish between 2 types of collaborative filtering models; neighborhood-based collaborative filtering and model-based collaborative filtering [44]. Model-based collaborative filtering uses factorization or deep learning methods to estimate ratings for unknown entries in a user-item matrix. These methods involve a low-dimensional latent representation for users (stores) and items (products). Considering the target store, this comes with the downside that we cannot reproduce the store cluster of the estimated ratings from these latent representations, since they do not rely on explicitly defined store similarity. Therefore, to maintain model explainability, we use a neighborhood-based collaborative filtering approach. Since we still use the notion of similar store clusters, we again use a K-nearest neighbor approach to select the most similar stores, i.e. stores within closest proximity. Proximity here is defined by combining store similarity and market share similarity as defined in section 3.3.1. Our neighborhood-based collaborative filtering model is user-based since we use stores as the basis for product recommendation, i.e. we use store similarity to recommend products.

Given our store similarity metric, we now select the K nearest neighbors of target store i, similarly as described previously in equation 8. Unlike our planogram compliance model, we do not use a dynamic number of K here. Because our goal is to recommend relevant but at the same time novel and serendipitous products, we set K equal to a fixed value of $K = 10$. For recommendations, we should not aim to maximize the prediction performance of our model at all costs, since this can have a negative influence on the coverage and diversity. Note that we select the K neighbors for store i that have the smallest absolute error AE_i between $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ as described in equation 17, given the optimized weights w_{0i} and w_{1i}.

For store i, this yields a predicted market share vector $\hat{\mathbf{y}}_i$ that contains predicted market shares for all available products j, either absent or present in the assortment. We are particularly interested in the predicted market shares for absent products, i.e. $a_{ij} = 0$. We consider these predictions as the expected market shares when the product would be offered in the assortment of store i.

3.3.3 Evaluating product recommendations

Where we want to recommend relevant items, market share predictions need to be accurate. We measure the prediction performance of our model using the absolute cluster error ACE_i for target store i as described in equation 19, where we only include predicted market shares of products that were already present in the assortment of store i, i.e. $a_{ij} = 1$. Note that we again use a LOO-CV approach, since we want to include every store in the neighborhood search of target store i.

Evaluating our recommendation model in terms of prediction performance ensures that recommendations are accurate. However, good recommendation systems also need to recommend novel and diverse items. Also, our recommendation system needs to be able to recommend items from the entire product set to prevent bias towards a specific product selection. We measure this in terms of *diversity* and *coverage*

[66]. We first define a recommendation matrix $\mathbf{R}$ as described in equation 27.

$$\mathbf{R} = \begin{pmatrix} r_{11} & \cdots & r_{1N} \\ \vdots & \ddots & \vdots \\ r_{M1} & \cdots & r_{MN} \end{pmatrix} \tag{27a}$$

$$= \begin{pmatrix} \mathbf{RC}_1 & \cdots & \mathbf{RC}_N \end{pmatrix} \tag{27b}$$

Recommendation matrix $\mathbf{R}$ and alternatively our column vector $\mathbf{RC}_j$ indicates if product j is recommended for store i, i.e. $r_{ij} = 1$ for a recommended item and $r_{ij} = 0$ for an item that is not recommended. Note that this matrix changes with the number of products that are recommended for a specific store, e.g. $\mathbf{R}$ is different for top-3 compared to top-10 recommended products.

Coverage indicates the portion of recommended products over all stores M concerning the total set of products N. In practice, this indicates what portion of the total product set the model is able to recommend. Therefore, high coverage is generally preferred. Following the definition in [66], we now define *coverage* as described in equation 28.

$$\textit{coverage} = \frac{\sum_{j=1}^{N} \max(\mathbf{RC}_j)}{N} \tag{28}$$

Note that products that are already sold in all stores cannot be recommended by the model. This means that our coverage cannot reach a value of 1.

Diversity indicates the dissimilarity between different sets of recommended items, i.e. the difference between $\mathbf{R}_0$ and $\mathbf{R}_1$, averaged over all sets of recommended items [38]. High diversity indicates that recommendations are customized to the characteristics of a store. This is generally preferred because we want to recommend items tailored to the needs of customers across different regions. We use the Jaccard distance to measure the dissimilarity between two sets of recommended items as described in equation 29 [5], for example store $i = 0$ and $i = 1$ respectively.

$$\text{jaccard}(\mathbf{R}_0, \mathbf{R}_1) = 1 - \frac{|\mathbf{R}_0 \bigcap \mathbf{R}_1|}{|\mathbf{R}_0 \bigcup \mathbf{R}_1|} \tag{29}$$

This results in a Jaccard dissimilarity matrix $\mathbf{J}$ of size $M \times M$, that contains every pairwise Jaccard dissimilarity j_{ij} for every store i as defined in equation 30.

$$\mathbf{J} = \begin{pmatrix} j_{11} & \cdots & j_{1M} \\ \vdots & \ddots & \vdots \\ j_{M1} & \cdots & j_{MM} \end{pmatrix} \tag{30}$$

We now define the *diversity* of our recommendation model as the average between all pairwise Jaccard dissimilarities, ignoring pairs of the same store, i.e. store $i \neq$ store j. Also following the definition in [66], this is defined in equation 31.

$$\textit{diversity} = \frac{\sum_{i \neq j}^{M} \sum_{j \neq i}^{M} j_{ij}}{M^2 - M} \tag{31}$$

4 Data

In this section, we discuss the data and data characteristics in detail. We describe our pre-processing steps and feature engineering steps for our data to be usable in our prediction models. Finally, we describe what aggregation levels we use on the product set concerning our planogram compliance model.

Our initial data set contains all anonymized sales data (in euros) of products in a specific product segment over a period of 57 weeks in 2020/2021, collected from all stores of a large FMCG retailer in the Netherlands. This results in a total of 1839 products and 886 stores. We have extensive data available about each individual product. This enables us to aggregate on different product characteristics, for example brand, product subcategory, and packtype. Finally, we have geographical data available of all stores in terms of longitude, latitude, and zipcode, which we use to determine product preferences and differences in market shares on a regional level.

4.1 Pre-processing

A relatively small portion of our product set is responsible for 99% of the total sales volume. The majority of products are typically sold in small quantities and a limited number of stores. This can be observed in figure 2 and 3. We decide to only include the product set that is responsible for 99% of total sales volume because this is the most interesting set concerning potential sales uplift. This results in a set of 631 products, which we use for further analysis.

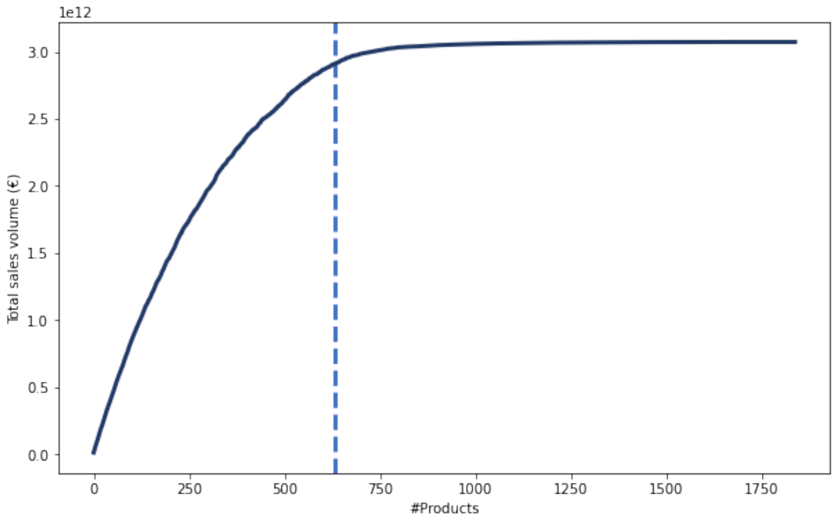


Figure 2: Cumulative product attribution to total sales volume

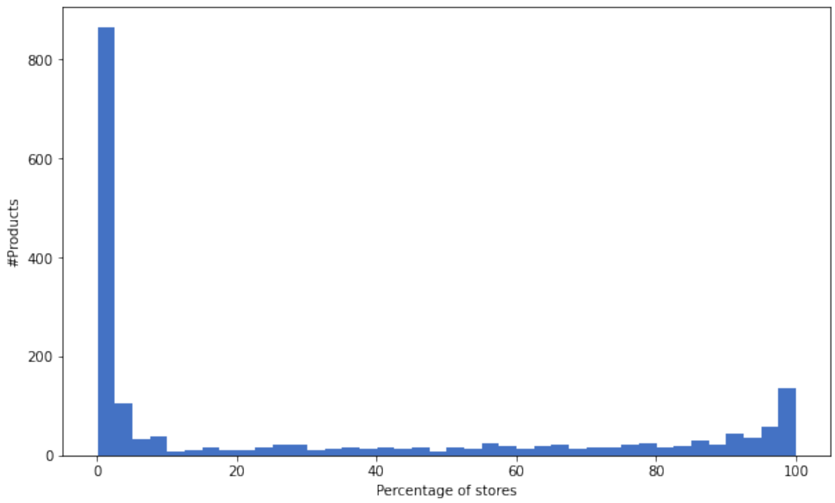


Figure 3: Percentage of stores where product is in assortment before filtering

In figure 3, we observe that most products are sold in a small number of stores. Removing the set of products responsible for approximately 1% of total sales volume, this affects specifically these type of products. This can be observed in figure 4.

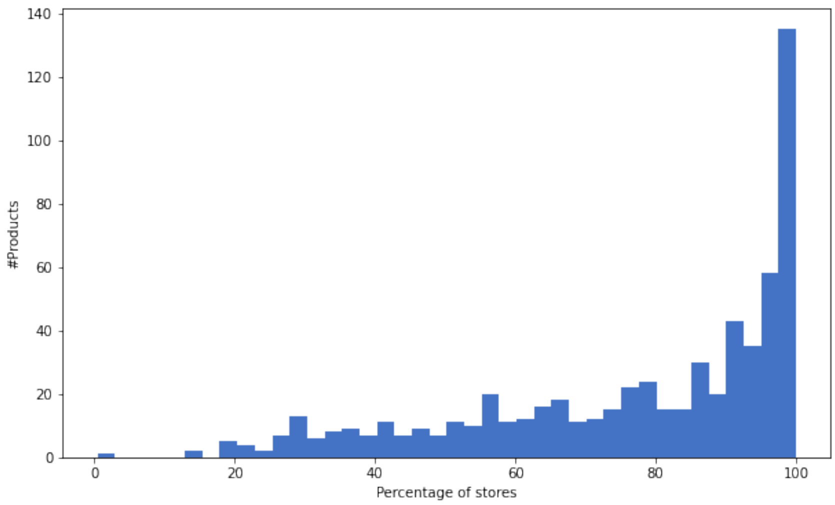


Figure 4: Percentage of stores where product is in assortment after filtering

A few products show unrealistically high sales volumes in particular weeks in some stores. We consider the sales of a specific product in a particular week an outlier if $S_{wi} > \mu(\mathbf{S}_w) + 3 \cdot \sigma(\mathbf{S}_w)$, where S_{wi} indicates the sales volume of a product in store i in a particular week w and vector $\mathbf{S}_w$ indicates the sales volumes of that product in all stores in the same week w [31]. After detecting these outliers, we impute the sales volume of an outlier with the median of the sales volume in that week. We do this because removal of the outlier will result in the misleading observation that the product is not sold at all in store i, which is not realistic.

4.2 Descriptive statistics

To gain further insights into the data, we provide various descriptive statistics and visualizations. After pre-processing, our data set contains 84 unique brands, 4 product categories, and 6 different packtypes. Each product represents a unique combination of these brands, categories, and packtypes. Due to the anonymization of our data, there is not more detailed information available about the specific content of these product characteristics.

In this section, we show what brands are often offered together in specific assortments. Also, we show which brands are preferred in a specific region for a subset of brands. Finally, we show how stores of the retailer are distributed over the Netherlands.

4.2.1 Brand co-occurrence

Especially for our recommendation model, it is useful to gain insights into brands offered together in a specific assortment. For example, the presence or absence of a specific brand can influence the sales volumes of other brands. We create the brand co-occurrence matrix following our assortment definition in equation 9, as shown in figure 5.

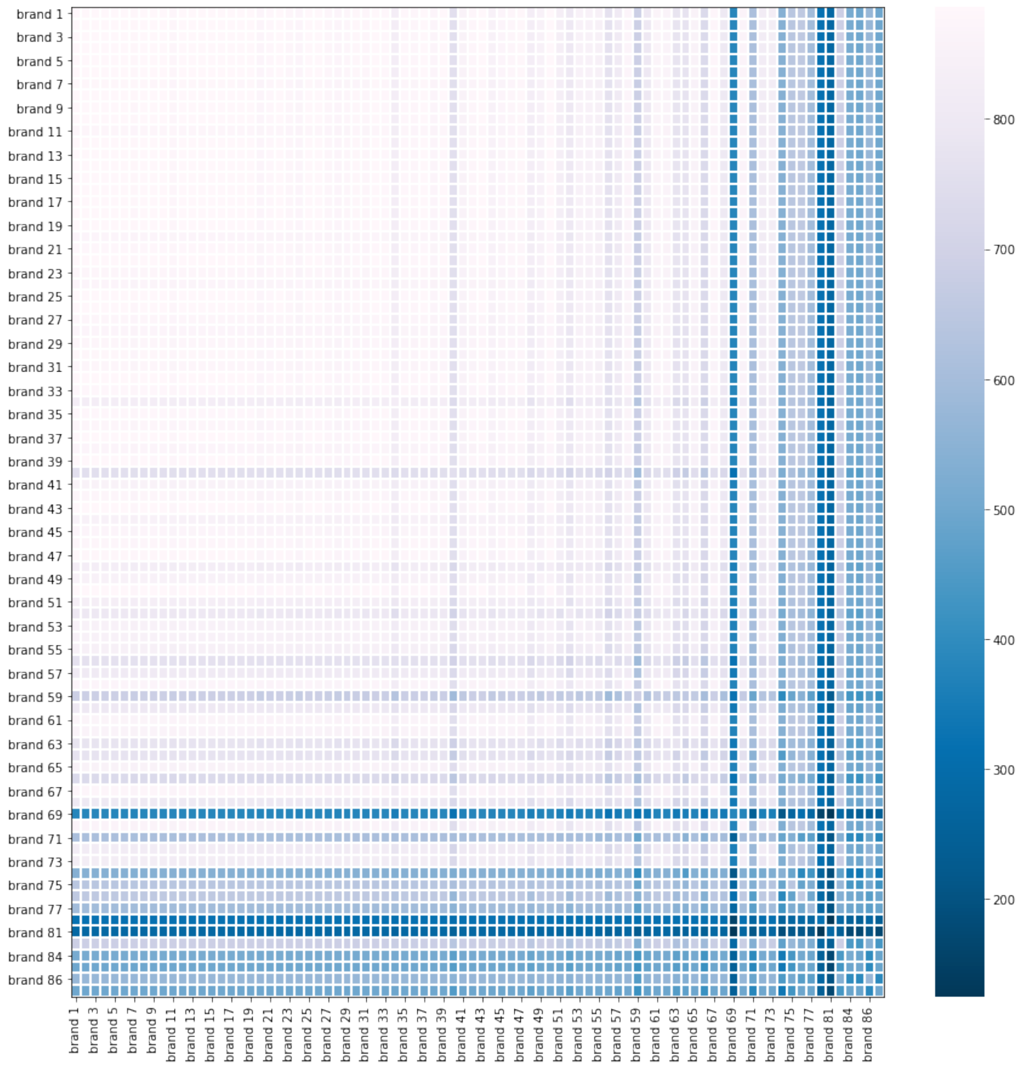
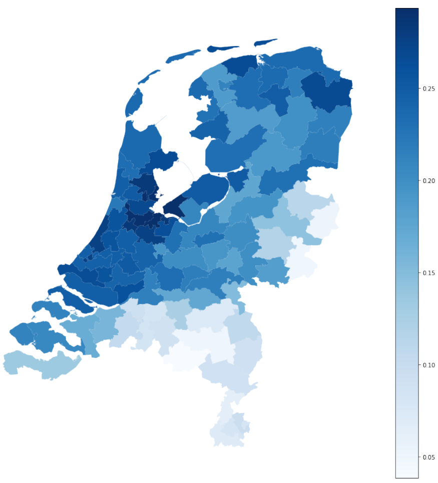


Figure 5: Brand co-occurrence

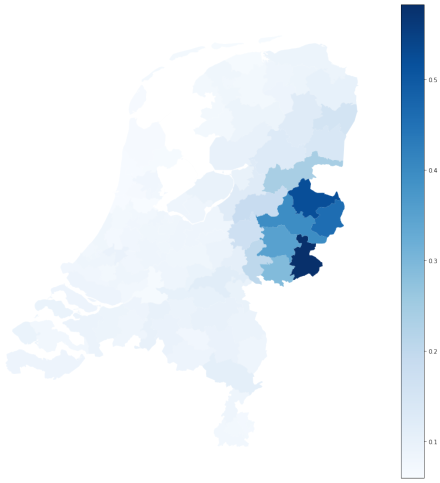
All brands are sorted based on their contribution to the total sales volume, i.e. brand 1 is the largest and brand 86 the smallest in our set of brands. We observe that large brands are frequently offered in stores and offered together with other large brands. On the other hand, we also observe that smaller brands are not offered in a majority of assortments and also not offered together with other small brands. For our recommendations, this can be a point of interest, because adding these brands to assortments might potentially result in large sales volumes in stores where the brand is not offered yet.

4.2.2 Market share distribution of brands

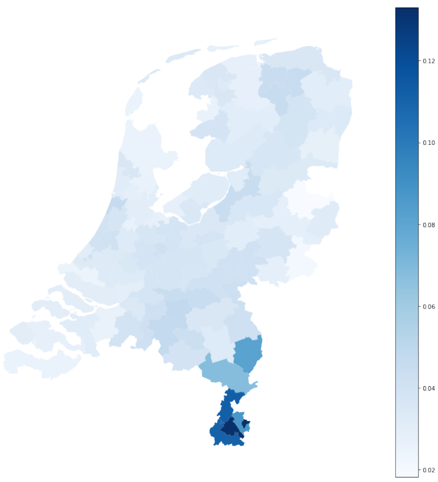
Nowadays, customers have a tendency to buy local products. As a result, brand preferences for a specific product category may differ highly across regions. To illustrate this, we show the geographical market share distribution of 4 different brands in our product set that show large regional market share differences. This distribution is depicted in figure 6.



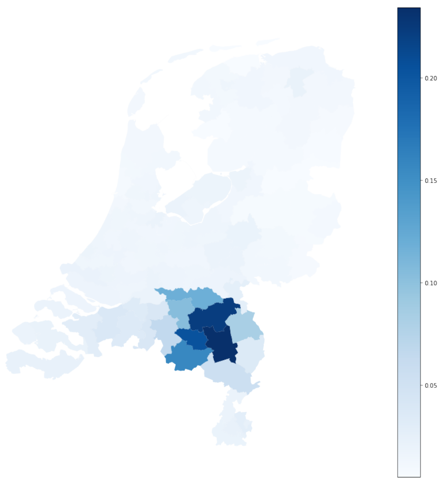
(a) Brand 1



(b) Brand 3



(c) Brand 6



(d) Brand 10

Figure 6: Brand market share distribution

The market share distributions are depicted on a regional (PC2) zipcode level. It can be observed that the market shares for this subset of brands in our product set differ highly between different regions in the Netherlands. Here we assume that market shares implicitly represent brand preference in a specific region. This product characteristic strengthens our earlier statement that brand preference differs highly across different regions.

4.2.3 Store distribution

The retailer considered has 886 stores across the Netherlands. These stores are distributed over the country as depicted in figure 7.

Figure 7: Store distribution

In figure 7, we distinguish between rural areas (low store density) and urban areas (high store density). This can influence our store clusters, wherein low store density areas clusters could have a larger spread over the region and vice versa. For customers, this spread can have effects on, for example, the willingness to visit an alternative store because it is easier for customers to go to other stores when these are within close proximity.

4.3 Feature engineering

For both our planogram compliance and product recommendation models, we consider different store features to differentiate between stores. Although our research focuses on one specific product segment within one specific retailer, we want our solution to be generic and applicable to different types of product segments and retailers. This means that we are restricted not to use product segment-specific or retailer-specific features. We include 4 different store features as an addition to our assortment and market share information, as shown in table 1.

Feature	**Additional information**
#SKU	Number of unique stock keeping units in store
Total sales volume	Total sales volume of all products sold in specified time frame
Longitude	X-coordinate (WGS coordinate system)
Latitude	Y-coordinate (WGS coordinate system)

Table 1: Store features

#SKU refers to the number of unique stock-keeping units sold in a specific store. This implicitly gives information about the size of the shelf space made available for this product segment. *Total sales volume* indicates the quantity of sales of the entire product segment within a specific time frame, which ranges from a few weeks (e.g. planogram compliance model) up to multiple months (e.g. recommendation model). We assume that the total sales volume is an implicit indicator for store size. Finally, we include both *longitude* and *latitude* as two distinct store features, represented with the World Geodetic System (WGS). This gives us detailed spatial information about the geographic location of each store and the density of stores in a specific region.

4.4 Aggregation levels

The available product data gives us detailed information about each product within the product segment. For our market share prediction model, brand, product subcategory, and packtype are examples of product characteristics that we can aggregate our product set on. Also, we experiment with aggregation on different time frame lengths. The time and product category aggregation levels are shown in table 2.

Product aggregation	**Time frame**
Brand	4 weeks
Brand, category	4 weeks
Brand, packtype	4 weeks
Brand, category, packtype	4 weeks
Brand	8 weeks
Brand, category	8 weeks
Brand, packtype	8 weeks
Brand, category, packtype	8 weeks

Table 2: Product and time frame aggregation levels

Note that we always include product brand in our aggregation. We do this because in practice the product brand is the most interesting factor for both retailers and manufacturers since this enables them to distinguish between their own products and products of competitors.

For recommendations, we do not aggregate on these product features and time frames, since it is more useful for a retailer to have individual products recommended instead of product groups, considering seasonal time frames of 6 months (e.g. summer and winter) instead of aggregation on 4 or 8 weeks to prevent recommending products that are based on temporal customer preferences.

5 Results

In this section, we present the results of both our planogram compliance and suboptimal assortment experiments after implementing our methodology as described in section 3. First, we present our quantitative results. For planogram compliance, these include the percentage of underperforming products, prediction performance, and the mean number of neighbors used in our clusters. For suboptimal assortments, this includes the prediction performance, coverage, and diversity. Also, we show visually what store clusters are used for the prediction of market shares and recommendations. Due to the anonymization of our data set, we do not provide detailed qualitative insights about for example our recommendations.

5.1 Planogram compliance

For our planogram compliance model, our goal is to develop a model that determines the rate of underperformance in a particular store, where we use the notion of store similarity to determine the expected market shares. First, we show that our model is able to predict market shares accurately in terms of our custom prediction performance measure. Secondly, we show the relation between our cluster size and the size of our prediction interval. Subsequently, we zoom in on the underperformance rates across all stores. We visualize what features contribute the most to the prediction of our expected market shares. Finally, we present a visual representation of our clusters for a subset of stores.

5.1.1 Prediction performance and neighbor pruning

For our identification of underperforming products, we experiment with different product aggregation levels and time frames as described previously in table 2. We report our prediction performance in terms of ACE, mean number of neighbors used in a cluster, and the average percentage of underperforming products. For all metrics, we also report a confidence interval. The results of these experiments are shown in table 3.

Product aggregation	Time frame	ACE	#Neighbors	Underperformance (%)
Brand	4 weeks	0.232 ± 0.004	8.90 ± 0.11	0.84 ± 0.09
Brand, category	4 weeks	0.252 ± 0.004	8.32 ± 0.13	0.85 ± 0.09
Brand, packtype	4 weeks	0.275 ± 0.004	7.00 ± 0.16	0.99 ± 0.09
Brand, category, packtype	4 weeks	0.280 ± 0.004	6.28 ± 0.17	1.13 ± 0.09
Brand	8 weeks	0.219 ± 0.004	9.19 ± 0.09	0.90 ± 0.11
Brand, category	8 weeks	0.240 ± 0.004	8.71 ± 0.12	0.92 ± 0.10
Brand, packtype	8 weeks	0.261 ± 0.004	7.51 ± 0.15	0.99 ± 0.09
Brand, category, packtype	8 weeks	0.278 ± 0.004	6.93 ± 0.17	1.07 ± 0.09

Table 3: ACE, number of neighbors and percentage of underperforming product categories including a confidence interval for each aggregation level

We observe that our model on the **brand** aggregation level with a time frame of **8 weeks** yields the best performance (low ACE), but has the lowest specificity concerning product category. However, because we want to determine product performance on the most specific product aggregation level possible, we opt to discuss our results on the most detailed aggregation level. Therefore, we discuss only the **brand, category, packtype** aggregation level on 8 weeks for the remainder of this section and report our figures based on this aggregation level and time frame. We select 8 weeks because this time frame yields a better prediction performance when compared to 4 weeks.

As mentioned in section 2.1.3, we need to find the optimal balance between the neighbor rejection rates and model performance, which varies with our pruning threshold θ_{AE}. We set θ_{AE} for neighbor pruning equal to 0.5, to ensure that we maintain high model performance without extreme rejection rates. Also, we only include stores in our results that have a minimum cluster size of 2. This is the minimum number of data points needed to accurately define a prediction interval, although this prediction interval can be large for small cluster sizes. We observe the average size of the prediction interval in relation to the cluster size in figure 8.

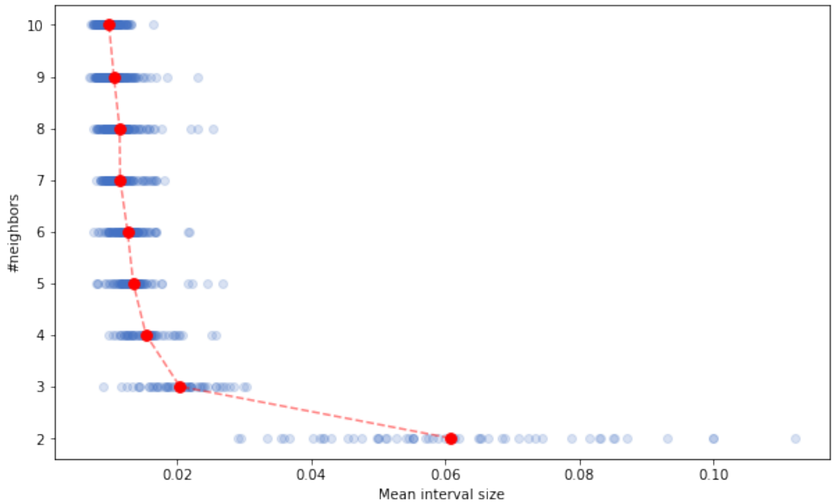


Figure 8: Size of prediction interval in relation to cluster size

For each number of neighbors, the group mean is shown in red. We observe that our mean interval size increases when the cluster size decreases. For small cluster sizes, e.g. cluster sizes of 2 or 3, we see a considerable increase in the average interval size due to higher uncertainty. The selected value for θ_{AE} results in the number of neighbors being rejected as presented in table 4.

Product aggregation	**Time frame**	**Rejections**
Brand	4 weeks	5
Brand, category	4 weeks	13
Brand, packtype	4 weeks	57
Brand, category, packtype	4 weeks	99
Brand	8 weeks	5
Brand, category	8 weeks	10
Brand, packtype	8 weeks	40
Brand, category, packtype	**8 weeks**	**58**

Table 4: Number of rejected neighbors

These rejected stores have a cluster size smaller than 2. Therefore we consider the similarity between this store and other stores too low to draw conclusions about product performance. For the remaining stores, the ACE_i is distributed as shown in figure 9.

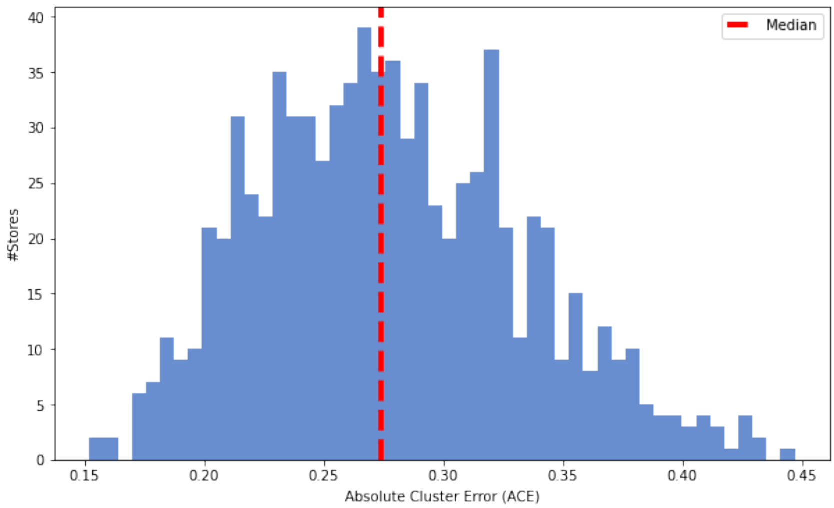
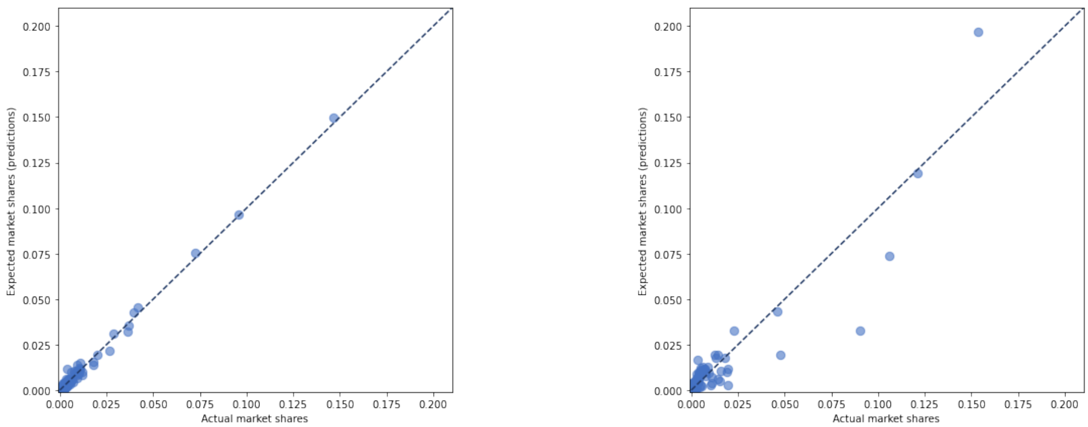


Figure 9: ACE distribution of stores with cluster size ≥ 2

We observe that there are large differences in prediction performance across different stores. For some stores (high performance) the expected market shares correspond with the actual market shares. For stores with low prediction performance, the expected market shares deviate strongly from the actual market shares. This could relate to an underperforming model, but can also indicate product underperformance, which is our hypothesis. In figure 10 we zoom in on 2 stores specifically, where the first store has the highest prediction performance and the second one the lowest respectively.



(a) Store 8714 (#neighbors = 10)

(b) Store 8674 (#neighbors = 3)

Figure 10: Prediction performance; highest (a) and lowest (b)

We see from figure 10 that in store number 8714 our model is able to predict market shares accurately, whereas in store 8674 there is a large deviation between expected and actual market shares for some products. Also, these example stores illustrate that a store with a low prediction performance is likely to base its prediction on small neighborhood sizes and vice versa. As we concluded before, the lower the number of neighbors, the larger the prediction interval.

5.1.2 Underperformance rate

In our research, we are specifically interested in the rate of underperformance. We expect the rate of underperformance to be low in most stores since they comply to a reasonable extent with their planograms. This can also be illustrated by our average underperformance rates in table 3, which generally equal approximately 1%. However, this average can be misleading, since a small number of stores can have high rates of underperformance without causing large differences in our average underperformance rate due to the high number of stores. Therefore, we show a more detailed visualization of underperformance in figure 11, which shows the underlying distribution of our underperformance rates.

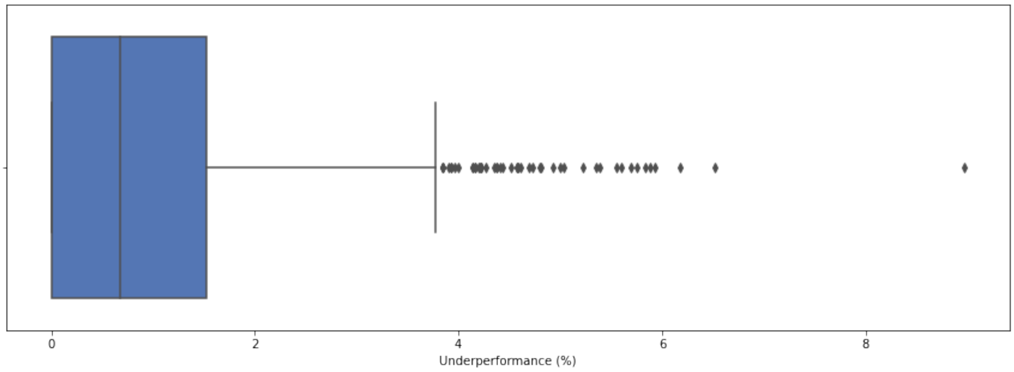


Figure 11: Underperformance rates for individual stores

In figure 11, we observe that most stores have low rates of underperformance (0-2%). However, a small number of stores have high underperformance rates, with a maximum rate of approximately 10%. These stores are particularly interesting, since we observe high underperformance rates for these stores, even though the store was not rejected by our model.

Low model performance can be attributed to an incorrect model or to planogram non-compliance. Our prediction interval should disentangle these two cases, at least to a certain extent. To illustrate that the rate of underperformance grows with the ACE of a store, given that we compensate for uncertainty, we observe the relation between the ACE of a store and the rate of underperformance in figure 12.

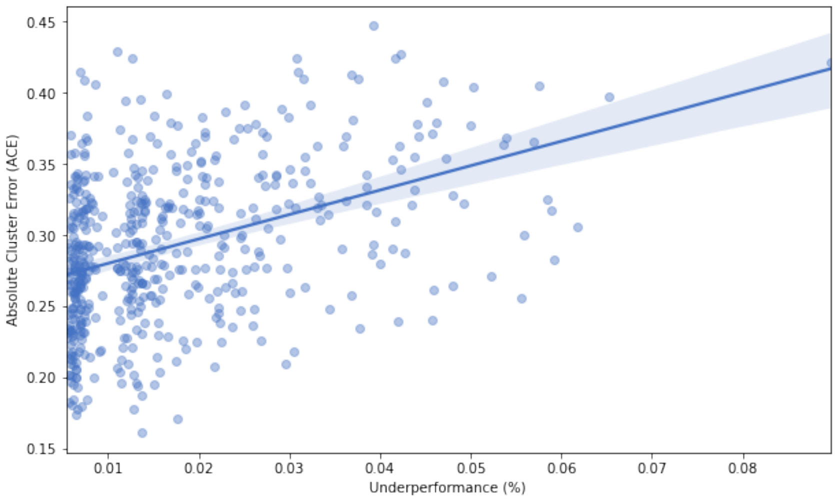


Figure 12: ACE in relation to underperformance rate

Figure 12 shows that there is a strong positive relationship between the rate of underperformance and the cluster error, i.e. underperformance rates generally increase when the ACE also increases. It is remarkable that even for large cluster errors, we still observe that products are underperforming because we see in figure 8 that prediction intervals for small clusters are usually large. This means that products are likely to show highly deviant sales patterns in these stores.

5.1.3 Feature importance

As described in section 3.2.3, we determine market share predictions using a Random Forest feature weighting approach. The average relative importance of the top 15 features that contribute the most to our dependent variable is shown in figure 13.

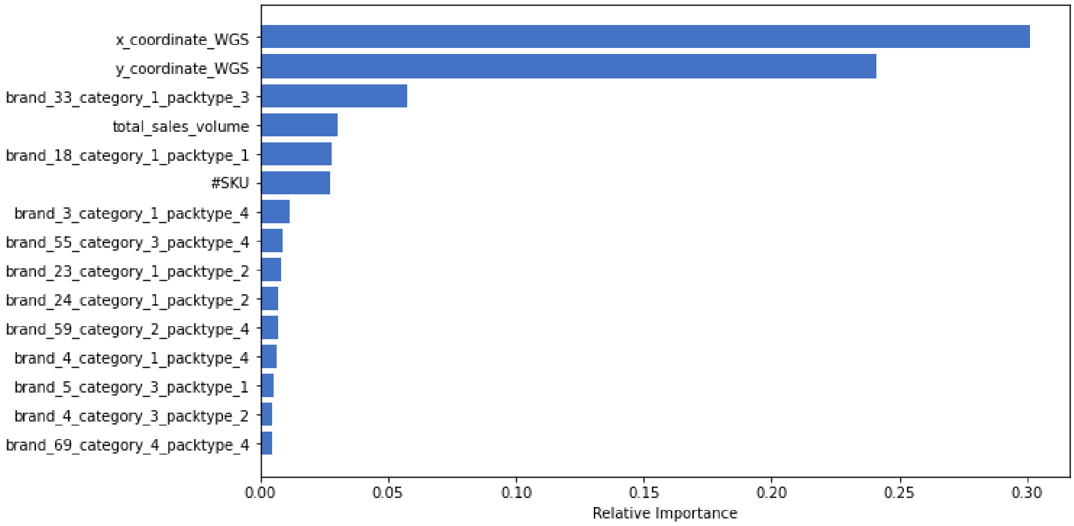


Figure 13: Average relative feature importance determined using Random Forest

We observe from figure 13 that our geographical features *longitude* (x_coordinate_WGS) and *latitude* (y_coordinate_WGS) contribute the most to our dependent variable. Additionally, *total sales volume* and *#SKU* do also contribute considerably to our dependent variable. The other features are binary variables that indicate whether a product category is present or absent in a specific assortment. We see that for some product categories, the absence or presence of a specific product contributes highly to the prediction of market shares, whereas other product categories do not contribute at all to these predictions. We discuss this in detail in section 6.1.

5.1.4 Cluster quality

Besides our quantitative results, we also discuss the quality of our clusters. To do so, in figure 14 we zoom in on 2 example stores in both rural and urban areas and different cluster sizes. Note again that for our planogram compliance models, we cluster our stores on the assortment selection (binary), total sales volume, number of stock-keeping units , and the geographical location. These are weighted as depicted in figure 13.

(a) Store 1003 (urban, #neighbors = 10) (b) Store 8776 (rural, #neighbors = 3)

Figure 14: Example clusters of stores with a small and large cluster size

We observe that these store clusters are formed with stores within close geographical proximity. This corresponds with our findings in figure 13, where we see that both *longitude* and *latitude* are of high importance. This figure also indicates that geographical location is important for product differentiation. Note however that clusters are not necessarily formed from the closest stores only, since we also include a set of non-geographical features. Again, we discuss further how to interpret these results in section 6.1.

5.2 Sub-optimal assortments

In this section, we describe our recommendation system for the recommendation of new products in a specific store. First, we describe the results of our seasonality analysis to determine the time window of our product aggregation. Secondly, we show the distribution of our weights between market share similarity and store similarity after our weight search. Thirdly, we describe our model quality both concerning prediction performance as well as coverage and diversity. Finally, we visualize the clustering of stores.

5.2.1 Seasonality

From our seasonality analysis, we find that **64.3%** of the products in our product set shows significant seasonal differences between winter and summer periods ($\alpha = 0.01$). This means that the market shares of a majority of the products significantly differ between winter and summer. Therefore, we focus our recommendation model on either the winter or summer assortment instead of an entire year to ensure that our recommendations are tailored to a specific season. More specifically, we use our summer assortment for weight selection and our winter assortment for validation.

5.2.2 Weight selection

For our collaborative filtering model, we use a combined similarity matrix composed of a store similarity matrix and a market share similarity matrix. We weigh this combined matrix for each store individually by applying a grid search for weight optimization. This results in the distribution of weight pairs W_{MS} and W_S as depicted in figure 15.

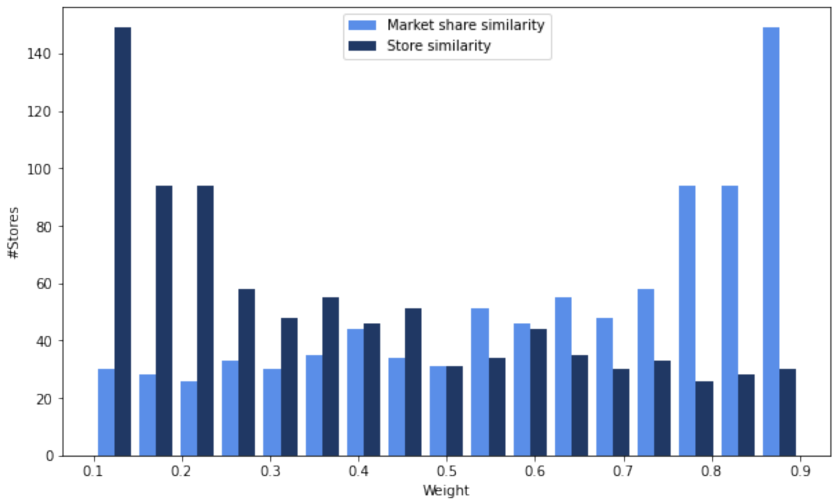


Figure 15: Weight distribution per store between store and market share similarity matrices

Note that the distributions are symmetrical to each other because we restrict our weight pairs to sum to 1. We observe that for most stores the market shares are of higher importance than our store features as described in table 1. However, store features are still considered to be the most important for a considerable number of stores.

5.2.3 Prediction performance

For our recommendation model, we first determine whether our recommendation model is accurate. We measure this using ACE as defined in equation 19. We use the summer assortment to determine our weights for each store. Because the weights are different for each individual store, we cannot use the summer assortment for validation. Therefore, we use the winter assortment for validation using the weights from our summer assortment acquired during our weight search. Here we assume that the weight distribution of W_{MS} and W_S between both seasons is similar for each store. Again, we use LOO-CV to ensure that all stores are included during clustering, except for the target store. This results in an average ACE of $\mathbf{0.318 \pm 0.004}$. For each store i, our ACE_i is also shown visually in figure 16.

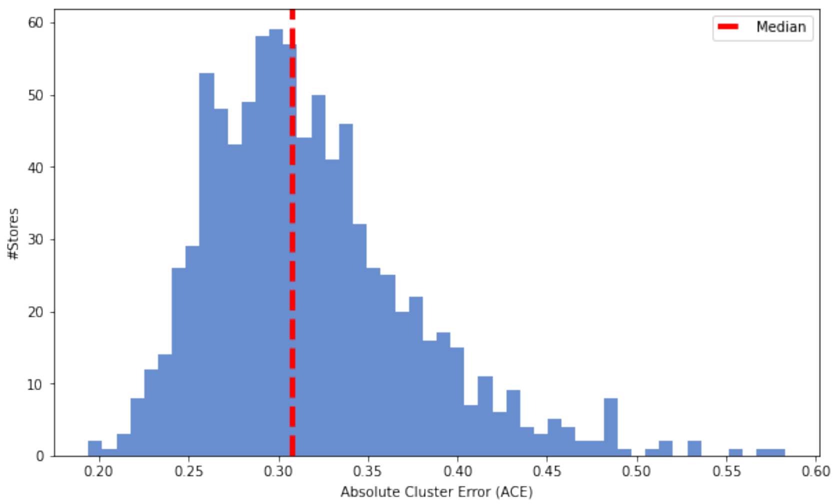


Figure 16: ACE of recommendation model per store

Again, we see that the model has a low prediction performance for a small number of stores, but a high prediction performance for a majority of stores. We maximized the prediction performance of our model using our weight search. Note as well that our prediction performance is lower than that of our planogram compliance model due to the absence of neighbor pruning.

5.2.4 Coverage and diversity

For all stores and sizes of recommendation sets, we report the coverage and diversity. We experiment with different sizes of recommendation sets. i.e. top 3, top 5, and top 10 recommendations for each store respectively. For these recommendation set sizes, the results are shown in table 5.

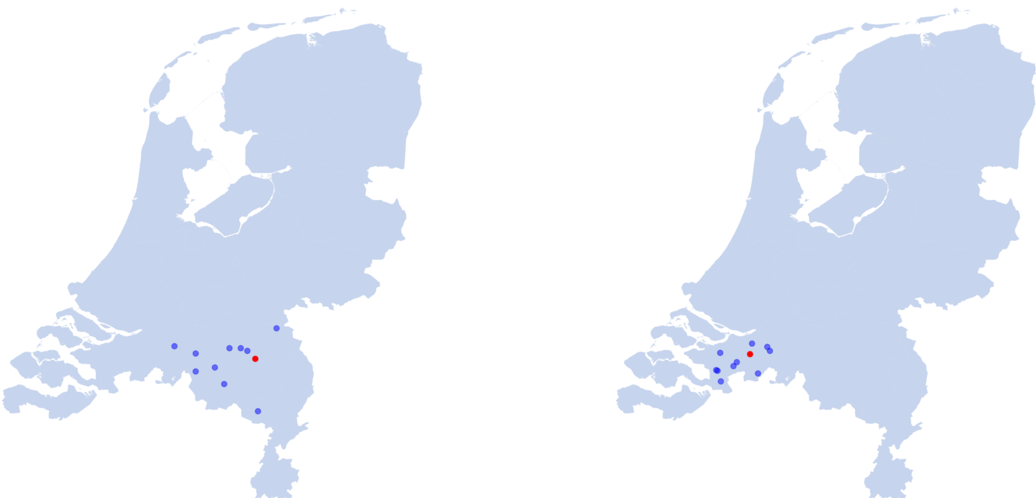
	Coverage	**Diversity**
Top 3	0.304	0.937
Top 5	0.394	0.935
Top 10	0.543	0.927

Table 5: Coverage and diversity of recommendations

Our coverage score increases when using larger sets of recommendations. This is expected since the probability of wider recommendations grows with the size of the recommendation sets. Also, we observe a decreasing diversity score. This is expected as well because a larger recommendation set has a higher probability of overlapping with other recommendation sets.

5.2.5 Cluster quality

Similar to our planogram compliance model, we see our store clustering visually as shown in figure 17. We show an example clustering of a store with a large market share similarity weight and on the other hand a clustering of a store with a large store similarity weight.



(a) Store 8786 ($W_{MS} = 0.9$, $W_S = 0.1$) (b) Store 8533 ($W_{MS} = 0.1$, $W_S = 0.9$)

Figure 17: Clusters of stores using different optimal weight pairs

In figure 17a, geographical location, which is a large component of store similarity, is only explicitly taken into account to a small extent ($W_S = 0.1$). Nevertheless, we see that the cluster of this store includes primarily stores that are within close proximity. This indicates that market share distributions implicitly contain geographic information. In figure 17b we can still observe an even stronger geographical clustering. This is expected, since for large values of W_S, geographical information is explicitly taken into account to a large extent.

6 Discussion

In this section, we discuss our results both on planogram compliance and suboptimal assortments in detail. Subsequently, we describe the limitations of our research and the possible entry points for future work.

6.1 Interpretation of results

The objective of this paper is twofold. On the one hand, we predict expected market shares for products that are present in the assortment of a specific store. Subsequently, we determine the rate of underperformance from the differences between the actual and expected market shares. On the other hand, we develop a recommendation model that can recommend novel products that are not offered in an assortment yet but may potentially sell well in that store. In our research, we address these objectives from the perspective of store similarity.

6.1.1 Planogram compliance

For our planogram compliance model to predict underperformance in a specific store, we experiment with different aggregation levels. First, we observe that longer time frames of 8 weeks result in a higher prediction performance compared to short time frames of 4 weeks. This may be attributed due to the fact that an average sales pattern over 8 weeks is less prone to weekly product sales outliers (less noisy) than aggregation over 4 weeks. We achieved the highest prediction performance on our **brand** aggregation level. However, from a retailer and manufacturer perspective, it is more useful to draw conclusions on the aggregation level that is as specific as possible. Therefore, we focused on the **brand, category, packtype** aggregation level over 8 weeks.

Zooming in on the prediction performance on the **brand, category, packtype** aggregation level, we see that the model is able to predict market shares accurately for some of our stores, for example as shown in figure 10a. On the other hand, our model predictions can also be deviant from our actual market shares as depicted in figure 10b. Note that products with large market shares drive our model performance to a greater extent than products with a low market share. This is desirable since these products have the largest uplift potential when complying accurately with the planogram. From a high model performance, we conclude that a store complies well with the planogram or at least complies with the planogram similarly to its store cluster. However, we cannot draw this conclusion for stores that achieve a low prediction performance. This is because we cannot determine whether this prediction performance is low due to incorrect planogram compliance or due to model inaccuracies. However, the latter is not likely since our model is able to predict market shares accurately for a subset of our stores. To address this issue, we introduced a prediction interval that grows with a decreasing number of neighbors and prediction performance. This is preferred since we want our margin of error to become larger when dealing with larger amounts of uncertainty. Using this method, we are able to determine underperformance even for stores where market share predictions are uncertain due to small cluster sizes. From figure 8, we observe that our average prediction interval grows with a decreasing number of neighbors. Also, we see from figure 12 that our underperformance rate increases when the cluster error increases. Combining these observations, we conclude that on average the underperformance rate increases when the model performance decreases. Zooming in on the underperformance rates, we observe that our average underperformance rate is approximately 1%. However, for a small subset of stores, we see considerably higher rates of underperformance up to almost 10% of our product category set. These stores are particularly interesting since these rates of underperformance are a strong indication for planogram non-compliance.

In our planogram compliance model, we weigh our feature set to ensure that relevant features are contributing more to the neighborhood search than irrelevant features. We include binary product category features that indicate whether a product category is absent or present in an assortment. Also, we included store-specific features as described in table 1. From our Random Forest weighting, we conclude that geographical location in terms of longitude and latitude is the most important feature for predicting market shares. This observation is in line with the strong regional differences that we observe for different popular brands in figure 6. Other store-specific features such as total sales volume and number of stock keeping units also contribute to the prediction of expected market shares, but to a lesser extent. Focusing on the binary assortment features, we observe high differences in feature importances. For this group of features, we see that in general large brands have low feature importances and smaller brands high feature importances. This can be attributed due to the fact that the presence of large brands does not differentiate stores from each other, because these brands are offered in almost every store. Smaller brands however are not always offered nationwide, but only in certain regions or even specific stores. Note that we determine the feature weight set for each store individually. This might induce some degree of model overfit for each store. However, assigned weights between each individual store only slightly deviate from the average feature weights as shown in figure 13.

Evaluating our cluster quality from figure 14, we observe well-formed clusters geographically, although geographic features were not the only features contributing to the store cluster. However, if we compare these clusters with the store distribution in figure 7, we see that a store is not necessarily clustered with its closest stores. This is desirable since we define closeness not only in terms of geographical distance but also in terms of assortment, total sales volume and the number of unique products. Also, we observe that store clusters differ in size due to our neighbor pruning criterion. Note that we need to find a good balance between the store rejection rate and the cluster error. For aggregation on a **brand** aggregation level, we observe low store rejection rates. However, on the more specific **brand, category, packtype** aggregation level, we observe relatively high rejection rates. On this level almost 100 stores of a total of 886 stores get rejected for small time frames, i.e. have a cluster size smaller than 2. Although this improves the market share prediction for the non-rejected stores, the rejection rate would be lower when we use a more lenient pruning threshold. We conclude that the store rejection rate and cluster error is well-balanced for low aggregation levels, but could be improved for high aggregation levels.

6.1.2 Sub-optimal assortments

Our recommendation system aims to recommend products that are not offered in a store yet but have the potential to sell well in that store based on their market shares in the store cluster. For our recommendation system to be accurate, we predict market shares of products that are already offered in the assortment with high prediction performance. In general, we observe a lower prediction performance when compared to our planogram compliance model. This can be attributed to the absence of neighbor pruning. However, unlike our planogram compliance model, this has the advantage that no stores are excluded, so we can recommend products for all stores in our data set.

We observe that the majority of products show significant differences between the winter and summer seasons. This results in aggregation on a specific season instead of an entire year. Aggregating on one season ensures that we do not recommend seasonal products in the wrong season. For example, we want to prevent recommending mulled wine in summer, since people only consume this product in winter periods.

In our product recommendation system, we use a weighted combination between market share similarity and store similarity. This combination is weighted individually for each store. Weighting this combination based on our summer assortment shows that market share similarity is the most important component to achieve high prediction performance. However, for a considerable number of stores, the store similarity component is found to be the most important. In an ideal situation, we want the market share similarity component to be completely independent of our store similarity component. However, we have seen before that regional preferences of certain products are prominent. Therefore, geographical location and market share distribution are strongly interrelated. This can also be illustrated by figure 17, where we observe a strong geographical store clustering even for low store similarity weights.

Evaluating our recommendation system in terms of coverage, we observe that our coverage score increases with the size of the recommendation set, where we use recommendations sets of sizes 3, 5, and 10 respectively. This is desirable since we expect our total recommendation set to grow with the number of recommended items per store. We do not reach the maximum theoretical coverage value of 1. This can be attributed to the fact that a reasonable portion of the product set is already offered in most stores, which by definition excludes these products from the recommendation set. Also, we only recommend products that potentially sell well. A large portion of products has a relatively low market share in most stores, which makes it unlikely that these products will end in a top-N recommended product set. We conclude that considering these issues, we are able to recommend a relatively large portion of our total product set.

For our diversity scores, we observe that the diversity decreases when the recommendation set increases. This is expected since the probability of overlapping recommendation sets increases when recommending larger sets of products. However, we observe that these diversity scores decrease to a small extent only. This indicates that our product recommendations are tailored to each individual store without a large overlap between recommended product sets. Since our average diversity score is high for all recommendation set sizes, we conclude that our recommendation system is able to recommend diverse items.

6.2 Limitations and future work

We propose a novel method to automatically detect underperformance and recommend products in a specific store. However, there has not been conducted much research in this field previously. Therefore, it is hard to compare the results of our experiments with other papers. Due to the absence of a baseline, especially on this data set, it is hard to quantify whether our model performs accurately. This can only be validated by physical in-store adaptations. Both for planogram compliance and product recommendations, this brings us to the most important limitation of our research; the absence of practical validation. In an ideal situation, we would validate our findings by in-store adaptations. After a longer period, one could measure what the influence of the adaptation had on the market share or sales volume of the respective product. Due to the anonymization of our data and time constraints, this is not a feasible option for our research. However, this paper can be a good entry point for a future study on this matter.

Another limitation of this paper is the fact that the data set is anonymized. A disadvantage of using anonymous data is that it is not possible to intuitively validate the outcomes of our research qualitatively. Especially for recommendations, an individual could observe quickly that for example recommending mulled wine in summer or ice cream in winter periods is not useful. However, simply recommending product IDs lacks this intuition.

Store restrictions are another aspect of our research that we do not focus on. We could for example recommend items to the assortment of a specific store, while this recommendation might not be usable in practice. This can be attributed to a variety of factors, for example due to limited shelf space, planogram restrictions by the retailer, or supplier contracts the retailer needs to adhere to. We do not have data available about these restrictions and therefore cannot include this information during model development.

Finally, we include a relatively small number of features in our feature set. As mentioned in section 4.3, we do not include product- or retailer-specific information. If our data would not have been anonymous, we could have included this information. Also, we could have included extensive demographic information about the population in a specific region, for example wealth levels or age distribution in a specific neighborhood. This could make it easier to discriminate stores and enhance cluster quality. For simplicity, we decide not to include this information. Again, this might be an interesting entry point for a future study.

7 Conclusion

In this paper, we propose and investigate methods to increase product sales volumes for stores within the FMCG industry. We focus on products that are present in a specific assortment, as well as products that are not yet present in an assortment. For products that are present in a specific assortment, we focus on planogram compliance and subsequently introduce a novel method to detect non-compliance automatically. For absent products, we develop a recommendation system that proposes new products for a specific assortment. Considering both problems, we use the notion that similar stores sell similar products.

Planogram compliance is the way retailers comply with their plan that describes how products are displayed in a retail location. This includes factors like correct merchandising, inventory levels, and the number of facings. Non-compliance generally results in lower sales volumes, which is undesirable. Cheap and scalable methods to automatically monitor planogram non-compliance do not exist. We propose a new method that detects non-compliance on a product category level where we compare the target store with a cluster of similar stores. This similarity is defined by the geographical location of the store but also by its sales volume, assortment selection, and the number of unique products offered. Using the KNN algorithm, we find high-quality clusters that enable us to accurately predict the expected market shares of a product category. These clusters are of different sizes because we prune neighbors that do not reach a minimum forecast accuracy level. This has the result that we reject some of our stores because we do not find neighboring stores that meet this minimum forecast accuracy threshold. Therefore, a careful selection of this neighbor pruning threshold is necessary to maintain a good balance between the store rejection rate and model performance.

Using our store clusters, we consider the differences between the actual and expected (predicted) market shares as an indication for non-compliance. However, smaller store cluster sizes increase the uncertainty with respect to our predicted market shares. Therefore, we introduce a prediction interval that grows with the level of uncertainty. This enables us to determine underperformance rates even for stores with high uncertainty. We conclude that most stores have a high model performance resulting in low rates of underperformance. On the other hand, for some stores, we observe low model performance with high rates of underperformance. These stores are particularly interesting because this indicates that a considerable rate of the product categories offered differ in sales from expectations, which could indicate planogram non-compliance.

In the second part of our research, we focus on sub-optimal assortments and develop a recommendation system that recommends products that may potentially sell well in the target store. To achieve this, we use a weighted combination of market share similarities and store feature similarities tailored to each store individually. We find that market share similarities are of the highest importance to optimize prediction performance. However, for a small number of stores, we observe that store feature similarities have a higher contribution to this weighted similarity. After using this weighted similarity to maximize the prediction performance, we determine the coverage and diversity for our recommendation sets. We observe that the recommendation system achieves high coverage, which indicates that our model is able to recommend a large portion of our product set. Also, we find high diversity values, which shows that our recommendations are at the same time diverse and tailored to the characteristics of each individual store.

To conclude, we propose a method based on store similarity that is able to detect missed opportunities accurately, both for products that are present and absent in the assortment of a target store, without human interference and at relatively low costs.

References

[1] Assortment strategies. *Corporate Finance Institute*, Jun 2020.

[2] Kanika Agarwal, Prateek Jain, and Mamta A Rajnayak. Comparative analysis of store clustering techniques in the retail industry. In *DATA*, pages 65–73, 2019.

[3] Victor Aguirregabiria and Gustavo Vicentini. Dynamic spatial competition between multi-store retailers. *The Journal of Industrial Economics*, 64(4):710–754, 2016.

[4] Xavier Andrade, Luís Guimarães, and Gonçalo Figueira. Product line selection of fast-moving consumer goods. *Omega*, page 102389, 2020.

[5] Sujoy Bag, Sri Krishna Kumar, and Manoj Kumar Tiwari. An efficient recommendation generation using relevant jaccard similarity. *Information Sciences*, 483:53–64, 2019.

[6] James R Bettman, Mary Frances Luce, and John W Payne. Constructive consumer choice processes. *Journal of consumer research*, 25(3):187–217, 1998.

[7] Shankar Bhandalkar. Global fmcg market expected to reach $15,361.8 billion by 2025. *Allied Market Research.*

[8] Chandrasekaran Anirudh Bhardwaj, Megha Mishra, and Kalyani Desikan. Dynamic feature scaling for k-nearest neighbor algorithm. *CoRR*, abs/1811.05062, 2018.

[9] Richard A. Briesch, Pradeep K. Chintagunta, and Edward J. Fox. How does assortment affect grocery store choice? *Journal of Marketing Research*, 46(2):176–189, 2009.

[10] Tom Brijs, Gilbert Swinnen, Koen Vanhoof, and Geert Wets. Using association rules for product assortment decisions: A case study. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 254–260, 1999.

[11] Susan M Broniarczyk and Wayne D Hoyer. Retail assortment: more$\neq$ better. In *Retailing in the 21st Century*, pages 225–238. Springer, 2006.

[12] Jacob Carr. A simple approach to retail clustering. *Wilson & Perumal; Company's Vantage Point*, 2013(3), 2013.

[13] Tianfeng Chai and R.R. Draxler. Root mean square error (rmse) or mean absolute error (mae)?– arguments against avoiding rmse in the literature. *Geoscientific Model Development*, 7:1247–1250, Jun 2014.

[14] Leonardo Weiss Ferreira Chaves. Planogram compliance using automated item-tracking, January 27 2015. US Patent 8,941,468.

[15] Timothy Chong, Idawati Bustan, and Mervyn Wee. Deep learning approach to planogram compliance in retail stores. *Semantic Scholar*, pages 1–6, 2016.

[16] Consultancy.uk. The 50 largest fmcg / consumer goods companies in the world. *Consultancy.uk*, Sep 2018.

[17] Hans Corsten, Michael Hopf, Benedikt Kasper, and Clemens Thielen. Assortment planning for multiple chain stores. *OR Spectrum*, 40(4):875–912, 2018.

[18] Marcel Corstjens and Peter Doyle. A model for optimizing retail space allocations. *Management Science*, 27(7):822–833, 1981.

[19] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46, 2010.

[20] Padraig Cunningham and Sarah Delany. k-nearest neighbour classifiers. *Mult Classif Syst*, Apr 2007.

[21] Joost CF De Winter. Using the student's t-test with extremely small sample sizes. *Practical Assessment, Research, and Evaluation*, 18(1):10, 2013.

[22] Renee Dudley. Wal-mart sees $3 billion opportunity refilling empty shelves. *Bloomberg*, Mar 2014.

[23] Leonardo Weiss Ferreira Chaves, Erik Buchmann, and Klemens Böhm. Finding misplaced items in retail by clustering rfid data. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 501–512, 2010.

[24] Marshall Fisher, Jayanth Krishnan, and Serguei Netessine. Retail store execution: An empirical study. *SSRN Electronic Journal*, Jan 2007.

[25] Robin Genuer. Variance reduction in purely random forests. *Journal of Nonparametric Statistics*, 24(3):543–562, 2012.

[26] Mohammad Ghasemi Hamed, Mathieu Serrurier, and Nicolas Durand. Simultaneous interval regression for k-nearest neighbor. In *Australasian Joint Conference on Artificial Intelligence*, pages 602–613. Springer, 2012.

[27] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *ACM SIGIR Forum*, volume 51, pages 227–234. ACM New York, NY, USA, 2017.

[28] Jim Hoover et al. How to track forecast accuracy to guide forecast process improvement. *Foresight*, 14:17–23, 2009.

[29] Li-Yu Hu, Min-Wei Huang, Shih-Wen Ke, and Chih-Fong Tsai. The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus*, 5, Dec 2016.

[30] Rob J Hyndman. Measuring forecast accuracy. *Business forecasting: Practical problems and solutions*, pages 177–183, 2014.

[31] Ihab F Ilyas and Xu Chu. *Data cleaning*. Morgan & Claypool, 2019.

[32] Marius Kaminskas and Derek Bridge. Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst.*, 7(1), December 2016.

[33] Will Kenton. Why fast-moving consumer goods matter. *Investopedia*, Sep 2020.

[34] Barbara E Khan and Donald R Lehmann. Modeling choice among assortments. *Journal of Retailing*, 67(3):274–300, 1991.

[35] A Gürhan Kök, Marshall L Fisher, and Ramnath Vaidyanathan. Assortment planning: Review of literature and industry practice. *Retail supply chain management*, pages 99–153, 2008.

[36] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[37] Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. A survey of serendipity in recommender systems. *Knowledge-Based Systems*, 111, Aug 2016.

[38] Matevž Kunaver and Tomaž Požrl. Diversity in recommender systems–a survey. *Knowledge-based systems*, 123:154–162, 2017.

[39] Caroline Lane. Planogram compliance. *Natural Insight*, Aug 2017.

[40] Song Liu, Wanqing Li, Stephen Davis, Christian Ritz, and Hongda Tian. Planogram compliance checking based on detection of recurring patterns. *IEEE MultiMedia*, 23(2):54–63, 2016.

[41] Cindy Lombart, Blandine Labbé-Pinlon, Marc Filser, Blandine Antéblian, and Didier Louis. Regional product assortment and merchandising in grocery stores: Strategies and target customer segments. *Journal of Retailing and Consumer Services*, 42:117–132, 2018.

[42] Murali Mantrala, Michael Levy, Barbara Kahn, Edward Fox, Peter Gaidarev, Bill Dankworth, and Denish Shah. Why is assortment planning so difficult for retailers? a framework and research agenda. *Journal of Retailing - J RETAIL*, 85:71–83, Mar 2009.

[43] Shelby H McIntyre and Christopher M Miller. The selection and pricing of retail assortments: An empirical approach. *Journal of Retailing*, 75(3):295–318, 1999.

[44] Prem Melville and Vikas Sindhwani. Recommender systems. *Encyclopedia of machine learning*, 1:829–838, 2010.

[45] Francis J Mulhern and Daniel T Padgett. The relationship between retail price promotions and regular price purchases. *Journal of Marketing*, 59(4):83–90, 1995.

[46] M Muthugnanambika, T Bagyammal, Latha Parameswaran, and Karthikeyan Vaiapury. An automated vision based change detection method for planogram compliance in retail stores. In *Computational Vision and Bio Inspired Computing*, pages 399–411. Springer, 2018.

[47] AA Nababan, OS Sitompul, et al. Attribute weighting based k-nearest neighbor using gain ratio. In *Journal of Physics: Conference Series*, volume 1007, page 012007. IOP Publishing, 2018.

[48] Philippe A. Naert and Alain Bultez. Logically consistent market share models. *Journal of Marketing Research*, 10(3):334–340, 1973.

[49] William Noonan. System and method of updating planogram information using rfid tags and personal shopping device, February 17 2009. US Patent 7,493,336.

[50] Duc Truong Pham, Stefan S Dimov, and Chi D Nguyen. Selection of k in k-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1):103–119, 2005.

[51] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[52] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*. Springer, 2011.

[53] Anurag Saran, Ehtesham Hassan, and Avinash Kumar Maurya. Robust visual analysis for planogram compliance problem. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 576–579. IEEE, 2015.

[54] Kåre Skallerud and Anders H Wien. Preference for local food as a matter of helping behaviour: Insights from norway. *Journal of Rural Studies*, 67:79–88, 2019.

[55] Joe Skorupa. Optimizing in-store merchandising. *RIS News*, Jul 2013.

[56] Nickolay Smirnov. Table for estimating the goodness of fit of empirical distributions. *The annals of mathematical statistics*, 19(2):279–281, 1948.

[57] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

[58] Chirag Surti, Prakash Abad, and Elkafi Hassini. Impact of inventory cannibalisation on a retailer selling substitutes. *International Journal of Operational Research*, 33(2):179–207, 2018.

[59] Stéphane Turolla. Spatial competition in the french supermarket industry. *Annals of Economics and Statistics/Annales d'Économie et de Statistique*, (121/122):213–259, 2016.

[60] Robin Van Meteren and Maarten Van Someren. Using content-based filtering for recommendation. In *Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop*, volume 30, pages 47–56, 2000.

[61] Diego P. Vivencio, Estevam R. Hruschka, M. Nicoletti, E. Santos, and Sebastian D. C. de O. Galvão. Feature-weighted k-nearest neighbor classifier. *2007 IEEE Symposium on Foundations of Computational Intelligence*, pages 481–486, 2007.

[62] Jigang Wang, Predrag Neskovic, and Leon N. Cooper. Locally determining the number of neighbors in the k-nearest neighbor rule based on statistical confidence. In Lipo Wang, Ke Chen, and Yew Soon Ong, editors, *Advances in Natural Computation*, pages 71–80, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[63] Barton A Weitz and Mary Brett Whitfield. Trends in us retailing. In *Retailing in the 21st Century*, pages 59–75. Springer, 2006.

[64] Stephanie Wiles. Planogram compliance: Making it work. *Cognizant*, Nov 2013.

[65] Tess Winlock, Eric Christiansen, and Serge Belongie. Toward real-time grocery detection for the visually impaired. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 49–56. IEEE, 2010.

[66] Wen Wu, Liang He, and Jing Yang. Evaluating recommender systems. In *Seventh International Conference on Digital Information Management (ICDIM 2012)*, pages 56–61. IEEE, 2012.

[67] Julie Young. What does the like-for-like sales number show? *Investopedia*, Dec 2020.

[68] Huanian Zhang and Fanliang Bu. Weighted knn algorithm based on random forests. pages 231–235, Feb 2019.

[69] Liang Zhang. The definition of novelty in recommendation system. *Journal of Engineering Science & Technology Review*, 6(3), 2013.