TREND FORECAST AND COLLECTION MANAGEMENT IN APPAREL RETAIL

RAMAZAN ARKAN

ADVISOR: ASST. PROF. DR. ESRA AĞCA AKTUNÇ

CO-ADVISOR: PROF. DR. AHMET DENİZ YÜCEKAYA

A thesis submitted to
the School of Graduate Studies of Kadir Has University
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy in
Industrial Engineering Graduate Programme

Istanbul, December, 2022

APPROVAL

This thesis titled TREND FORECAST AND COLLECTION MANAGEMENT IN APPAREL RETAIL submitted by RAMAZAN ARKAN, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Industrial Engineering Graduate Programme is approved by

Asst. Prof. Dr. Esra Ağca Aktunç (Advisor) Yeditepe University	. . . .
Prof. Dr. Ahmet Deniz Yücekaya (Co-Advisor) Kadir Has University	. . . .
Asst. Prof. Dr. Mustafa Hekimoğlu Kadir Has University	. . . .
Assoc. Prof. Dr. Pelin Gülşah Canbolat Kadir Has University	. . . .
Asst. Prof. Dr. Emre Çelebi Yeditepe University	. . . .
Assoc. Prof. Dr. Gülşah Hançerlioğulları Köksalmış Istanbul Technical University	. . . .

I confirm that the signatures above belong to the aforementioned faculty members.

. . . .

Prof. Dr. Mehmet Timur Aydemir

Dean of School of Graduate Studies

Date of Approval: 28.12.2022

DECLARATION ON RESEARCH ETHICS AND PUBLISHING METHODS

I, RAMAZAN ARKAN; hereby declare

- that this Ph.D. Thesis that I have submitted is entirely my own work and I have cited and referenced all material and results that are not my own in accordance with the rules;

- that this Ph.D. Thesis does not contain any material from any research submitted or accepted to obtain a degree or diploma at another educational institution;

- and that I commit and undertake to follow the “Kadir Has University Academic Codes and Conduct” prepared in accordance with the “Higher Education Council Codes of Conduct”.

In addition, I acknowledge that any claim of irregularity that may arise in relation to this work will result in a disciplinary action in accordance with university legislation.

RAMAZAN ARKAN

. . . .
28.12.2022

to my family

ACKNOWLEDGEMENT

I would like to thank my advisor Asst. Prof. Dr. Esra Ağca Aktunç for their consistent support and guidance during the running of this research. Asst. Prof. Dr. Esra Ağca Aktunç continuously provided encouragement and was always willing and enthusiastic to assist in any way she could throughout the research in this dissertation.

I would also like to thank committee members, Prof. Dr. Ahmet Deniz Yücekaya, Asst. Prof. Dr. Burak Çavdaroğlu and Assoc. Prof. Dr. Gülşah Hançerlioğulları Köksalmış for their valuable inputs that make this research possible.

Finally, I would like to thank my wife, Rabia Karaaslan Arkan for supporting me during the compilation of this dissertation and all my life.

TREND FORECAST AND COLLECTION MANAGEMENT IN APPAREL RETAIL

ABSTRACT

This study addresses the new methods and some existing methods with a different approach for trend forecasting and using new trends in the collections in apparel retail industry. There are several approaches to determine the potential of fashion trends. This study describes several approaches for trend forecasting and develops methods for measuring the potential of new fashion trends with unknown potential and without sales data. Firstly, merchandise testing focuses on the process of testing products with new trends. It describes the test store selection, forecasting methods and analyze the accuracy of forecasting with real data. Secondly, Sales-Based Store Network of Stores model is presented to examine cross-store sales similarity and establishes a store network using Collaborative Filtering method as in recommendation systems. A clustering method like K-means is studied to cluster the stores using store network data. Moreover, Distribution of Collection into Store method focuses on distributing the main collection made for a category into each stores using some constraints such as capacity of stores, rates of product attributes in the main collection. Integer programming is used to distribute the collection. The sales potential of the new planned products is crucial. It is necessary to choose the products with highest potential among the hundreds of products. Prediction of products' demand based on stores addresses a prediction model using sales data containing store features and product attributes with different forecasting methods with different parameters. Furthermore, store-based forecasts are used in Distribution of collection into stores method while selecting the best products for the stores.

Keywords: Apparel Retail, Fashion Trends, Merchandise Testing, Forecast, Clustering, K-means, Integer Programming, Collaborative Filtering.

MODA PERAKENDE SEKTÖRÜNDE TREND TAHMİNİ VE KOLEKSİYON PLANLAMA

ÖZET

Bu çalışma, hazır giyim perakende sektöründeki koleksiyonlarda trend tahmini ve yeni trendlerin kullanımı için yeni yöntemleri ve mevcut bazı yöntemleri farklı bir yaklaşımla ele almaktadır. Moda trendlerinin potansiyelini belirlemek için çeşitli yaklaşımlar vardır. Bu çalışma, trend tahmini için çeşitli yaklaşımları açıklamakta ve potansiyeli bilinmeyen ve satış verileri olmayan yeni moda trendlerinin potansiyelini ölçmek için yöntemler geliştirmektedir. İlk olarak, ürün testi konusu (Merchandise Testing), yeni trendlere sahip ürünleri test etme sürecine odaklanır. Test mağazalarının seçimini, tahmin yöntemlerini açıklar ve gerçek verilerle tahminin doğruluğunu analiz eder. İkinci olarak, mağazalar arası satış benzerliğini incelemek için Satış Tabanlı Mağaza Ağı modeli sunulmuş ve tavsiye sistemlerinde olduğu gibi İşbirlikçi Filtreleme yöntemini kullanarak bir mağaza ağı kurmuştur. Mağaza ağ verilerini kullanarak mağazaları kümelemek için K-means gibi bir kümeleme yöntemi incelenmiştir. Ayrıca Koleksiyonun Mağazaya Dağılımı yöntemi, bir kategori için yapılan ana koleksiyonun, mağazaların kapasitesi, ana koleksiyondaki ürün özelliklerinin oranları gibi kısıtlar kullanılarak her mağazaya dağıtılmasına odaklanmaktadır. Koleksiyonu dağıtmak için tamsayılı programlama kullanılmaktadır. Planlanan yeni ürünlerin satış potansiyeli çok önemlidir. Yüzlerce ürün arasından potansiyeli en yüksek olan ürünleri seçmek gereklidir. Mağaza bazında ürün talebi tahmini, mağaza özelliklerini ve ürün özelliklerini içeren satış verilerini farklı parametrelerle ve farklı tahmin yöntemleriyle kullanan bir tahmin modelini ele alır. Ayrıca, koleksiyonların mağazalara dağıtılması yönteminde mağazalar için en uygun ürünlerin seçiminde mağaza bazında tahminler kullanılmaktadır.

Anahtar sözcükler: Hazırgiyim Perakende, Moda Trendleri, Tahminleme, Gruplama, K-means, Tamsayılı Programlama.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$r_{x,i}$	rating of user x on item i.
$sim(u,v)$	similarity coefficient between user u and user v.
S	the $m \times n$ matrix of store features
S_j	the $m \times 1$ column vector for store feature j for all stores, $j = 1,2,\dots,n$.
U_j	the set of unique values in column vector S_j.
SE	$m \times D$ matrix of zeros and ones obtained by one-hot encoding.
r_i	the priority/importance coefficient of store $i = 1,2,\dots,m$.
x_i	the binary decision variable
x_{pi}	the sales quantity of product p in store i.
μ_i	the average sales quantity in store i.
σ_i	the standard deviation of sales quantities in store i.
z_{pi}	the z-value of product p in store i.
d_{ij}	distance between store i and store j.
N_i	the set of products sold in store i.
z_{pc}	the z-value of product p in cluster c.
F_{pi}	the sales forecast for product p in store i.

LIST OF ACRONYMS AND ABBREVIATIONS

AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
BCG	Boston Consulting Group
CF	Collaborative Filtering
CM	Category Manager
CRM	Customer Relationship Management
IP	Integer Programming
LP	Linear Programming
MA	Moving Average
MAD	Mean Absolute Deviation
MAPE	Mean Absolute Percentage Error
MSE	Mean Square Error
OLS	Ordinary Least Squares
RMSE	Root Mean Square Error
ROC	Area Under Receiver Operating Characteristics
SVM	Support Vector Machine
TSS	Test Store Selection
TUIK	Turkish Statistical Institute
WMAPE	Weighted Mean Absolute Percentage Error

1. INTRODUCTION

Fashion trends are very important aspects of fashion retail industry. There is a very dynamic environment for fashion trends in the fashion industry. New fashion trends emerge, some continue, some lose their popularity, and some get stronger every season. While the highlights of the previous season were very different, it is possible to see brand new details in the new season. Consumers tend to keep up with the fashion trends. Including new trends in the collection increases the sales potential. There are many resources that enable consumers to keep up with trends such as social media, celebrities, design companies, influencers, and fashion brands. Since consumers follow the trends and novelty in fashion, fashion retail companies should follow the fashion trends closely, analyze them in terms of their potential customers and include them in their collections. Thus, companies should be able to keep up with their customers and maintain their market share.

In the process of preparing a collection, there is so much subjectivity coming from the category managers, which may not always result in the best decisions for the fashion retail company. They carry out the research for market, competitors, social media, street, etc. and they analyze the potential products that are suitable for their customers. Then they choose the products they are sure about based on mostly their instincts and the limited information they collected. This process does not always result in the optimal selection of trends or products. Therefore, there is a need for smart algorithms and optimization studies in order to keep subjectivity at a minimum level in the process.

There are core/essential/basic products that have potential to sell every season; and key products with trend items whose sales potential changes according to the state of fashion trends. It is easier to plan basic products in the collection and to estimate the sales potential because there is data from the past. However, it is more difficult to estimate the potential of key products. Category managers need to obtain more information about the fashion trends used in key products.

The objectives of this study are to determine the potential of key products using strategic planning techniques, to predict the ratings of products based on stores, to group the stores based on sales performance of products, to distribute the region-based collection into stores based on their capacity and attribute rates. In this thesis, we will develop the smart algorithms for the cases above in an effort to improve the collection preparing process for fashion retail companies.

The research questions of this study are as follows:

1. How can the rating values indicating the potential of new products be calculated? Can the rating values be store-based?
2. How can stores be grouped by category based on their sales performance?
3. Which algorithm can be used in order to distribute the collection prepared for a region into stores?
4. How can the attribute (or product feature) rates in the region-based collection be preserved in each store?

The outcomes of this study will be a tool consisting of a test product algorithm to test the potential of the products with new trends, a collection distribution algorithm to distribute a large product collection into small stores based on their product capacity, a store-based network algorithm to develop a sales-based network between stores and to group them.

1.1 Background about the Fashion Retail Industry

For fashion retail companies, it is not an easy task to follow and analyze the trends. Companies have some alternative ways to do that. Trend forecasting firms like WGSN, fashion fairs, social media, customer research, and competitor analysis are some of the methods that fashion retail companies can use. By applying those methods, fashion retail companies try to determine the increasing trends and apply them into their collections, and to determine the decreasing trends and avoid them. When companies use the methods above, they become able to know which fashion trends are increasing or decreasing globally. However, they need to analyze the fashion trends in a more systematic way. Since they want to prepare and present the best collection for their customers, they need to determine the right trends and the right time to apply them for their customer profiles.

There are some methods to determine whether trends or innovations are rising or falling. Diffusion of Innovation Theory and Boston Consulting Group (BCG) Matrix are the best-known methods used in fashion retail industry.

Diffusion of Innovation Theory is introduced by Everett Rogers in 1962 and it examines the process of adoption of innovations such as an idea, a product, a philosophy, or a technology by the people (Rogers, 1983; Kaminski, 2011). People in a social system have different tendencies to adopt innovations. According to this theory, there are five categories for people adopting innovations. These are innovators, early adopters, early majority, late majority, and laggards. The distribution of five levels of adopting an innovation are shown in Figure 1.1 below.

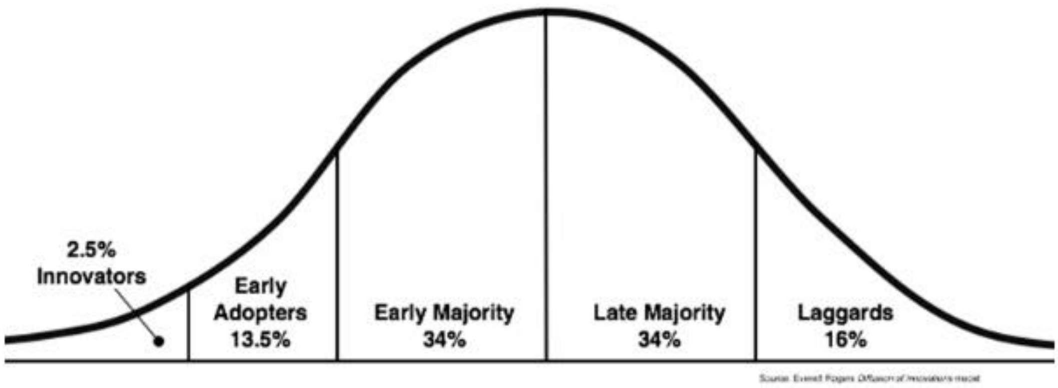

Figure 1. 1: Adopter categories in Diffusion of Innovation Theory (Rogers, 1983)

The definitions of each category based on Rogers (1983) are as follows:

Innovators: This group consists of a minority of people, who are eager to try innovations and new ideas. Innovators must be able to deal with a high degree of uncertainty when the innovator group adapts to an innovation. Innovators have an important impact on the diffusion process, although the other parts of the social system may not respect them. They have a guarding role in the movement of innovations into social system.

Early Adopters: Early Adopters are part of the social system compared to innovators. This group of people has leadership characteristic in the social system. Other adopting groups look at them and approach innovations more warmly.

Early Majority: About a third of the society are early majority adopters of innovations. Their time of adopting is later than innovators and early adopters.

Late Majority: Just after half of the society adopts an innovation, the late majority group adopts innovations. This adoption may be the result of economic need and increasing community pressure. They approach the innovations disbelieving and poised.

Laggards: Laggards are the last part of the social system for adopting an innovation. While these are adapting to an innovation, there may be a new idea that innovators adopt in the social system.

Since there are different stages of people to adopt innovations, there are different departments such as casual, classic, or smart in so many retail companies to deal with different type of customers. It is important to analyze potential customers and place them in the appropriate innovation adoption categories. New trends are accepted firstly by innovators and their share in total is very low, nearly 2.5%. After a certain time, early adopters tend to be interested in trends. Then, trends start to be seen on a lot of people and brands. Early majority and late majority care about those trends, respectively. After those trends are seen everywhere and after a long time from the beginning, only laggards are interested in those trend details. As each phase passes, the previous category of people tends to lose their interest in the trend, because they think those trends are getting old. Fashion retail companies need to determine their customers in the perspective of those five innovation adoption levels. They need to follow trends and determine the adopting level the trend is suitable for them.

The second method to analyze trends is the Boston Matrix developed by the Boston Consulting Group (BCG). It is a strategic planning technique that helps various firms assign their resources based on the level of market growth and relative market share (Dag Øivind, 2017). This matrix identifies four fundamental groups of trends or products: Cash Cows, Stars, Dogs, and Question Marks. Further details of the Boston Matrix are provided in the Literature Review section below. Fashion retail companies define the market growth of a trend and its relative market share to include the right trends in their collections.

1.1.1 Definitions of basic terms in apparel retail

Product: Anything that is sold in brick-and-mortar stores or in e-commerce.

Option: Color variant of a product. It is lowest unit of the hierarchy before the size.

Category: Group of products separated in order to work with and analyze easily. A category consists of similar products such as shirts, pants, t-shirts, skirts, etc.

Attribute: Features of products. It is also called “range”. There are attribute categories and features under those categories. For example, “pattern” is a range group and “printed”, “solid” are the range definition or attribute definition under pattern. In addition, “fit” is a range group and there are “slim”, “regular”, “extra slim”, “loose” definitions under “fit”.

Store features: They are actually the features of stores’ surroundings. Income level, population density or average residential rental values of the region where the store is located.

Capacity: It is related to product-based capacity in the stores. It refers to how many products can be displayed for a category in the store. It is determined in terms of categories.

Collection: A package consisting of different sizes of a product. It is used to facilitate shipping products from the warehouse to the stores.

1.1.2 Process of preparing collections in apparel retail

The process of preparing a collection in fashion retail is shown in Figure 1.2. Preparing a collection starts with sales. After sales occur, category managers start to analyze their categories’ sales. In addition, they look for their competitor brands, social media, street, TV series, TV programs, influencers, etc. They analyze the products sold in the current season. Category managers decide which products should be removed from collection and which continue in the collection next season. While choosing the products that continue in the collection, the state of products is important: Are they basic or trendy? If they are trendy, then category managers should be sure that the trend used in the product is still rising next season. In addition, they need to select new products instead of removed ones. While selecting new products, they should analyze the attribute rates in the collection.

Attribute rates or range rates are important while preparing collections. Category managers analyze the attribute rates of the current season by week or by month: Which product features are successful, which ones unsuccessful? For example, if black color is successful, they can increase the rate of black in the collection. Here, they need to check the situation of black color in the market looking for trend analysis.

The planning department defines product capacities of categories. They determine the rate of categories, so capacity of categories in each store is calculated based on those rates. Number of products that should be in the collection for categories is shaped according to capacities determined by planning department.

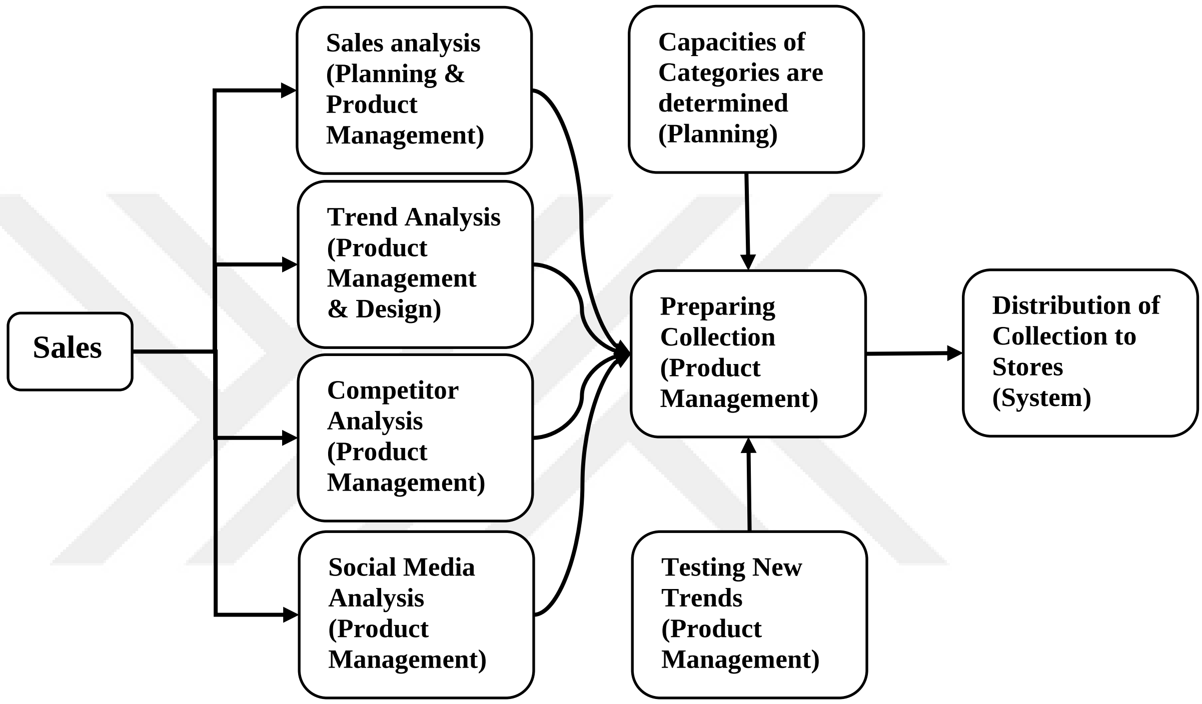

Figure 1. 2: Process of preparing collections in fashion retail

Category managers prepare the collections for each category. Collections are prepared for global, country, region, or city levels. Each store has a different product capacity. Since category managers cannot prepare different collections for each store, the collection prepared for the region or global is distributed by the system. If the capacity of a store is less than the number of products in the collection, some products are eliminated. This elimination should be the result of a smart algorithm, using optimization or heuristic approaches.

While category managers are preparing the collection, they use the results of the trend analysis and competitor analysis. They follow the fashion trends and they use some of the trends that they are sure is suitable for their customers and categories. They have some analysis results, but they want to be sure about the trend. They test the trends that they are not sure about. They use the trend in a product and they order a low quantity. After that product is sold, they analyze the sales and according to the analysis, they decide the quantities that should be produced.

1.2 Motivation

There are a lot of manual tasks and subjectivities in the collection preparing process. In addition, there are no standard methods for those in fashion retail industry. To distribute the region-based collection to stores based on their capacity and attribute rates is a new type of optimization application for the retail industry.

BCG Matrix is generally used for a specific product sold in the market. Firms want to describe the situation of the product in the market and to decide on the investment policy. Market growth and relative market share are two metrics that firms need to determine. It is easy to make an analysis of some type of products such as detergent, chocolate, etc. It is possible to obtain sales quantities of such products in the market. Firms also know their sales quantities and they can calculate the relative market share. However, it is not so easy to make that analysis for fashion trends. Fashion trends are not like the fast-moving consumer goods. It is difficult to obtain total sales quantities of a fashion trend. Experts working in fashion retail can do BCG matrix analysis subjectively. However, it is very difficult for them to decide the order quantities of the products with new trends without being sure of the accuracy of that BCG matrix analysis. Merchandise testing model that predict the potential of the product with new trend will meet this need.

1.3 Subject and Scope

The researcher in this project is currently working in one of the leading international fashion retail companies in Turkey and will be basing his research on the clothing collections of this company and its competitors.

We will group the stores in terms of products' sales performance. This study will be category-based. It means that there will be different clustering studies for each category. We will use only the sales quantities for products in stores. We will not include the store features or product attributes in this method. However, we will use store features in test product process and product attributes in the distribution algorithm and prediction of new products for stores.

There is “Literature Review” in the second chapter. Then there are “Methodology” part which introduces the studies in the dissertation. After that, “Store-Based Merchandise Testing for Apparel Retail” is in fourth chapter. The fifth chapter covers “Sales-Based Similarity Network of Stores” topic. There are “Distribution of Collection into Stores” and “Prediction of Products’ Demand Based on Stores” topics are in the sixth and seventh chapters respectively.

2. LITERATURE REVIEW

2.1 Boston Matrix Analysis

Boston Matrix, also called the growth-share matrix, was developed by Bruce Henderson, the founder of Boston Consulting Group (BCG), in 1970 (Henderson, 1970). It is a strategic planning technique that helps firms with diversified products to improve their performance by allocating their resources wisely. Even though its popularity has slightly declined since the 1990s, it is still one of the most influential and widely used techniques in managerial practice, consulting, and business school education (Dag Øivind, 2017). It consists of high and low levels of two dimensions: market growth and relative market share. Each quadrant of this 2×2 matrix identifies one of the four fundamental results: Cash Cows, Stars, Question Marks, and Dogs, as shown in Table 2.1 below. The most critical principle in portfolio management consists of two items: One of them is desirability of market, i.e., market growth rate or market size, and the other one is level of competition of a firm in the market with the index of market share and brand recognition (Guta, 2016). The purpose of this analysis is to focus the firm on the division of resources between different tactical parts of activity.

Table 2. 1: Boston Matrix

	High Market Share	Low Market Share
High Market Growth	Star	Question Mark
Low Market Growth	Cash Cow	Dog

According to this matrix, we should dispose of the dogs, weigh the risks and rewards of the question marks, keep the cash cows, and invest money in the stars. Although determining the category of each product is not as easy as this matrix makes it seem to be, it provides a simplified overview of portfolio management. Whitehead (2015) defines the BCG Matrix as “one of the most symbolic strategy markers of all time”. Harvard Business Review considered the BCG Matrix as one of the top five charts that “changed the world” (Ovans, 2011). There was a unique contribution of fashion, networks, and consulting companies in the popularization of the BCG Matrix (Morrison and Wensly, 1991). Seeger (1984) discussed the idea that the BCG Matrix may be oversimplifying the instructions for managers and business school students and reminded that “the dogs may be friendly, the cows may produce more than milk, and the stars may already have burned themselves out”. Therefore, business managers or consultants should be careful about using the BCG Matrix and consider other aspects of the products that may impact their future potential.

2.2 Fashion Trend Forecasting

The journey of fashion industry began with haute couture, evolved into ready-to-wear and finally reached fast fashion (Binda and Merlo, 2020). Fashion products are remarkable, as they are used not only outside but also at home (Yoganarasimhan, 2012). Clothes symbolize people’s character and often play a significant role in how others see us in both our real and digital lives (Stefani et al., 2019). Due to the dynamically changing conditions that affect fashion, trend forecasting has always been crucial to predicting consumer preferences and trends in the future (Furukawa et al., 2019; Rousso, 2012). Prediction of fashion trends is a difficult and exciting subject for designers, researchers, and marketing specialists (Samit et al, 2020). With better predictions, fashion companies can improve themselves in product design and marketing strategy development, while customers can make better purchasing decisions.

There are important roles of fashion weeks, trend forecasting firms and images shared in social media in forecasting fashion trends (Park et al., 2016; Rousso and Ostroff, 2018). Long before the season, some organizations like Pantone present the potential colors. Then, trend-forecasting companies declare the potential colors and combinations nearly two years before the seasons. Some design companies prepare their collections and exhibits in fashion weeks. Design brands put coming trends in their collection during the season (Samit et al., 2020). Products including new trends have a short life; therefore, it is difficult to understand the constitution of a fashion trend or a fashion rule in the quickly changing fashion world (Stefani et al., 2019).

The fashion industry benefits from companies like WGSN and Edited for trend prediction and trend analytics to avoid overstocks or loss of sales (Jackson, 2007). Google Trends can also be used to predict future buying decisions and measure the efficiency of marketing operations. There may be some alternative research areas coming up to be discovered, not only in terms of fashion design, purchasing and sales, but also in terms of fashion management. There is an opportunity to benefit from the application of statistical models to analyze seasonal changes in data and to make future insights and seasonal forecasts for the fashion industry (Silva et al., 2019).

Predicting fashion trends also require following music, art, and other social aspects that may affect fashion trends and customers' lives. Internet has a significant contribution in the rapidly changing fashion world and results in more difficulty in making fashion trend forecasts (Yunshan, 2020). Recently, social media became the main source of fashion forecasting rather than fashion shows or e-commerce sites. It is fed by photos and comments from many sources. Trend analysis and forecasting has always been an important area of research as it feeds the fashion industry (Yunshan, 2020). Designers often look at photos shared on social media to adapt innovations to products. In the last few years, social media has been full of fashion inspiration from celebrities, well-known designers, and fashion influencers (Vijay, 2018).

Quality, material and price are as important as visual features like color, style and any other components of products in purchasing decision for customers. To understand the customers' choice in buying a product, it is important to study visual aspects of products. For instance, when a feature is trendy, so many people may like a shoe with that feature. After fashion changes, some other people may continue to like that shoe because of the other features as leather, color, etc. (Liu and Shen, 2018). Earlier research about fashion trend forecasting have ignored the relationship between color and other aspects, therefore, current research focuses more on these issues for trend prediction (Samit et al., 2020). Color is an important factor in product selection and plays an important role in trend prediction. By color trend forecasting, firms manage the production, marketing and design process (Furukawa et al., 2019; Cassidy and Cassidy, 2012). In recent years, researchers focused on determining the impact of color, shape and other aspects in products using the images collected in internet (Berg et al., 2010; Bossard et al., 2012; Bourdev et al., 2011; Chen et al., 2012; Parikh and Grauman, 2011). For example, black color, which has been the color of status, richness, and authority for centuries, is very important for fashion industry, because demand for it increases when the economic crisis breaks out and it also represents strength and trust in the financial industry (John, 2017).

In addition, sales of products with black color in Fall/Winter Seasons is greater than that in Spring/Summer Seasons (Koh, 2018).

The variables in fashion forecast studies are either time-dependent or not. Since fashion changes rapidly in time, it becomes difficult to determine the time-dependent variables (Liu and Shen, 2018). Rapidly changing fashion trends force designers to diversify every season (Caro and Martínez-de-Albéniz, 2015). Time series methods, such as AR (Autoregressive), MA (Moving Average), and ARIMA (autoregressive integrated moving average), are used to predict the future of a fashion trend using past data (Yunshan, 2020).

Nowadays, it is not enough to predict trends based on historical sales data alone. It is necessary to implement the practice of machine learning in fashion industry (Al-Halah et al., 2017). After the recent advances in machine learning methods, the fashion industry has also tended to use these methods in fashion trend forecasting (Furukawa et al., 2019). Fashion companies are motivated to apply machine-learning studies in predicting the fashion trends and in classifying their customers, because of the massive use of internet (Lord et al., 2004; Mona, 2017; Park et al., 2016).

2.3 Optimization in Retail

Assortment is the product set exposed in the stores. Retail companies revise their assortments periodically, take some products out of the assortment and put any other products instead of them to meet customer demand. When customers do not find the product that they need in the assortment, they may buy another product similar the product out of stock in the store (Fisher & Vaidyanathan, 2014). Assortment optimization problems both provide optimal assortment that meet customer needs and support managers working in this direction (Fisher & Vaidyanathan, 2014).

Smith and Agrawal (2000) studied an integer-programming problem for assortment planning. They solved a problem to indicate the impact of substitution behavior on assortment planning. Ghoniem and Maddah (2015) developed mixed integer programming model inventory management, pricing strategy and assortment planning. They focused on the substitutable products that meet the customer demand, but are different with attributes. They studied Multi-period Mixed Integer Nonlinear programming to optimize assortment, inventory and price. Yucel et al. (2009) studied a mixed-integer model for inventory management, pricing and assortment optimization. They also focused on product substitution problem. Miller et al. (2010) developed Multinomial Logit (MNL) model with integer programming to detect the optimal assortment. They added the constraints about retail such as space and assortment size limits. Fisher and Vaidyanathan (2014) studied demand models for optimal assortment and they obtained 5.8% and 3.6% revenue increase in two applications.

Subramanian & Sherali (2009) worked on mixed-integer programming problem to optimize category pricing in terms of price ladder. Hanson and Martin (1996) studied MNL-based optimization problem by handling prices as continuous variables. Mulhern and Leone (1991) worked on the optimization of promotion and pricing to develop the profitability in price discounts. Natter, et al (2007) developed a dynamic pricing model with a retailer. They focused on promotion effect in the market and item-based profit maximization instead of category-based.

Ferreira, et al (2015) stated a decision support system for pricing for a retailer with the objective function that is profit maximization in terms of category level. They did not consider the dynamic promotional effects. Hall et al. (2010) worked on discrete-time dynamic programming model for substitutable products to determine the optimal inventory level and price.

Ailawadi et al. (2007) introduced a study for promotion influences on profitability by switching categories and stocking. Cohen et al. (2014) stated the linear approximation to optimize promotion timing and depth including the business dynamics as constraints. Their model was based on profit maximization for single item and they did not consider cross-product promotional impacts. Ma and Fildes (2017) developed promotion optimization problem to maximize total profit. They considered finite time horizon and their decision variables were determined for each period.

The personnel-scheduling problems have been important in last decades. This is about creating optimal timesheets for organizations to reduce labor cost and increase productivity (Talarico & Duque, 2015). Melachrinoudis and Olafsson (1995) used an integer-programming model to prepare optimal shifts to increase customer satisfaction. They considered full-time and part-time workers and they used the workload per hour and staff availability as constraints. In addition, Menezes et al. (2006) developed an optimization model to manage workforce for retail stores. The model was based on staff allocation problem. They took over the workers in a staff pool in a geographical area and workers could be reallocated between stores. They tried to solve optimal workforce to increase profit. Mirrazavi and Beringer (2007) worked on staff shift optimization problem. They aimed to minimize labor costs, to balance between under-staffing and over-staffing. They assigned works to working days in weekly horizon. Model reduced the managerial workload of store managers and enabled them to focus more on customers' needs. Furthermore, Miwa and Takakuwa (2010) used integer linear programming model to minimize daily total working time. Their constraints were based on the number of operations to be finished for each period.

For supply chain network optimization, there are many models to use. Some of them are Mixed Integer Linear Programming (MILP), Linear Programming (LP) modeling and Discrete Event Simulation (DES), Metaheuristics. Optimal facility location, capacity allocation and transportation cost are some of aspects used in supply chain optimization problems (Munasinghe & Rupasinghe, 2016). To determine the places of retail stores and service centers, cost factors, pedestrian traffic, parking and transportation facilities and investment restrictions are the fundamental constraints of optimization problems (Gao et al., 2011).

Facility problems aims to open facilities in the best locations to ensure customer satisfaction. Those models can be used for warehouse, retail stores, public facilities. Integer programming, linear programming, heuristic and metaheuristic approach are used in such models. The focus of the problems are location and relocation of facilities to maximize total profit. (Sierra-Paradinas et al., 2020).

2.4 Merchandise Testing

Sales forecasting in fashion retail is a widely researched area; however, since this study focuses on retail or merchandise testing for apparel products with new trends, only the most relevant studies are reviewed here. As Fisher and Rajaram (2000) stated, inventory management and sales prediction are challenging topics for new fashion items since they have highly unpredictable demand and a short lifecycle, making it difficult to understand the constitution of a fashion trend (Stefani et al., 2019). Due to the dynamically changing conditions that affect fashion, trend forecasting has always been crucial to predicting future consumer preferences (Furukawa et al., 2019; Rousso, 2012). Prediction of fashion trends is a challenging and exciting subject for designers, researchers, and marketing specialists (Chakraborty et al., 2020). As Rajaram (2008) stated, although several articles (Doyle and Gidengil, 1977; Fox, 1995; Hollander, 1986; Pollack, 1994; Winston et al., 1995) emphasized the importance and need of merchandise testing, they did not suggest a useful method for this process.

An effective strategy for retail testing is to first select a subset of all stores as test stores, then predict the sales for all the stores. Many retailers apply a test method in which they put a small number of products in a small number of carefully selected stores in order to reduce the cost of forecast errors (Fisher and Rajaram, 2000; Chen et al., 2017). Most testing methods are performed on only a small sample of stores due to store space limitations, labor cost, and other logistical constraints.

As one of the leading studies on fashion merchandise testing, Fisher and Rajaram (2000) select the test stores based on store clustering using the similarities in cross-store sales mix for the previous season. Then, forecasts are made both for each cluster and for the entire chain using the sales ratio of each product in each store. A k-median model is proposed to group the stores into k clusters. One store in each cluster is selected as a test store, resulting in a total of k test stores. Store descriptors are used as the second way of clustering the stores. The store features used are store latitude and longitude to calculate the distance between stores with Euclidean distance, average temperature, total sales, ethnicity, and neighborhood type. The weighted distance between two stores is calculated with nonnegative weights. After test stores are selected, sales are monitored during a test period of two or three weeks before the regular season, and the demand for the regular sales season is forecasted. The weights used to scale test sales in forecasting are optimized with a linear program (LP).

The examples of retail testing in the literature generally focus on selecting test stores according to sales history rather than store features. Chen et al. (2017) use sales-based clustering to select the test stores. Gallien et al. (2015) use the sales history of comparable products to analyze the stores to forecast the demand for new products of Zara. Features of stores are not covered widely in the literature. Despite the importance of such decisions for supply chain management, only a few studies have evaluated the effects of hierarchical location-based decision-making (Syntetos et al., 2015). As Mendes and Cardoso (2006) stated, locations of the stores, consumer behavior, demographic, socioeconomic, and geographical information provide an understanding of the store performances. As Bradlow et al. (2017) stated, it is very important to give the right message to the right customers for retailers. In addition to the right message and the right customer, product availability at the right location is also an important factor in the effectiveness of marketing for retailers. If companies collect the location information of their customers in the CRM database, they can find new opportunities in location-based forecasting models. There are three main sets of data across the wide variety of data sources that companies trust. The first is traditional enterprise data; second is customer identity, characteristics and profile data; and the last one is location-based data. The best way to select test stores would be a random sample with specific control mechanisms to balance the type of store, store size, geographic location, and demographics of trade areas. In this paper, we focus on the features of stores such as store size, store turnover, and features of the store area that are population, education level, average income level, average clothing spending, average restaurant and hotel spending, average entertainment spending, number of stores of competitors, and the number of stops for transportation.

Retail store clustering has been studied either solely or in the context of forecasting. Mendes and Cardoso (2006) used three different cluster methods on data that include the location and store attributes, influence area attributes of stores, and customer characteristics such as behavior and socioeconomic levels. The first method was Ward's hierarchical clustering procedure using dissimilarities matrix filled by experts. The second method was based on the expert choice from several regression trees created with different parameterizations. The last one focused on the interactive expert selection of key cluster variables followed by cluster calculation. Tehrani and Ahrens (2006) divide sales quantities into 3 clusters by k-means algorithm. Based on the result of k-means clustering, the lower and upper bounds on sales are defined. Sales quantities less than the lower bound belong to the first cluster, between the lower and upper bounds belong to the second cluster, and higher than the upper bound belong to the third cluster. Then, ordinal logistic regression is able to forecast the cluster of products. After identifying the cluster of new fashion products, sales quantities of those products are forecasted by using the kernel regression model. Lee and Bradlow (2011) study k-means clustering algorithm to cluster the extracted expressions using customer reviews. They aim to analyze the market structure and define the product attributes for the market structure analysis. Their main focus on market structure analysis is related to marketing practice and text mining of user-generated content.

After the test stores are selected, sales of the test product must be monitored. Chen et al. (2017) use two periods for merchandise testing. The first one is a test period of two or three weeks before the regular season, and the second one is the regular sales season. In the test period, the products are tested in test stores, and the demand for the regular sales season is forecasted. To avoid the substitution effect and ensure that the sales of test products are not affected by comparable or competitive products, the products are tested in the test period. In this study, we focus on the products with new trends such that there is no earlier similar trend. A test period of two or three weeks is assumed; however, this test period can be at any part of the regular sales season. Mostard et al. (2011) propose a demand forecasting method for the entire chain rather than store-based forecasting, whereas Chen et al. (2017) and Gallien et al. (2015) make sales forecast at the store level. In this study, store-based forecasts are made for the demand for test products.

There are several studies about forecasting in retail, including demand forecasting and price forecasting. Wang et al. (2020) studied Amazon tablet computer data set and tried to predict the sales price. They used the attributes of products such as RAM, GHz, storage, battery life, and age; market price dynamics such as sales rank, number of new features, number of currently used features; customer reviews that include average review rating, the average number of words in the reviews. They used four different methods, ordinary least squares (OLS), support vector machine (SVM), regression trees, bagged trees, and compared the results. The worst performance belongs to OLS. SVM, regression trees, and bagged trees have better RMSE and R-square values than OLS. Schwartz et al. (2014) analyzed data sets based on summary statistics. They focused on classification methods from machine learning and studied a decision tree that suggested which methods to use and when. They studied regression trees, classification, and random forest algorithms. Cui and Curry (2005) used Support Vector Machine with the kernel to predict to results of developing environments in marketing, for instance, automated modeling, mass-produced models, intelligent software agents, and data mining. Jacobs et al. (2016)

studied a text-modeling algorithm by using customer characteristics to make accurate predictions. They analyze the sales data and try to predict what the customers will buy in the future. Fildes et al. (2022) studied multiple linear regression to forecast the store-based sales of the products and used weather conditions, seasonality, price, calendar events, and promotion features as input. Ma and Fildes (2021) worked on a sales forecast for a retail product evaluating the performance of meta-learning based on the deep convolutional neural network. The model learns the feature from the sales data with time series and uses the learned features with weights.

A summary of the most relevant literature is presented in Table 2.2. Based on the above review of the literature, most of these studies solve one of the three problems that are clustering, demand forecasting, or merchandise testing individually. To the best knowledge of the authors, the work of Fisher and Rajaram (2000) is the only one that discusses the three problems together like this study. The closest studies to our work are Fisher and Rajaram (2000) and Chen et al. (2017), in terms of dealing with test store selection, allocation of low quantities of test products to test stores, and forecasting demand using test products' sales. The study of Gallien et al. (2015) is also similar to our study in terms of making store-based forecasts of the demand for new products. However, they use the sales history of comparable products determined by experts. The current study focuses on the store features and uses them for both test store selection and forecasting. Stores are clustered based on each feature separately (feature-based clustering). Moreover, store feature descriptions are improved so that feature data can be more useful in clustering the stores. Test stores are selected such that the distribution of store feature values among all stores is maintained in the selected test stores. Demand forecasts for the non-test stores are made using the sales at the test stores in the three-week test period.

Table 2. 2: Literature Summary

Study	**C[1]**	**DF[2]**	**MT[3]**	**Method**	**Independent Variables**	**Detail Level**
Fisher and Rajaram (2000)	x	x	x	k-median, LP for weight optimization	Sales history, Store location	Cluster-based

Alon et al. (2001)		x		Artificial Neural Network (ANN), Winters Exponential Smoothing, Box-Jenkins ARIMA Model, Multiple Regression,	Sales history	Product-based
Cui and Curry (2005)		x		Support Vector Machine with kernel	Sales history	Product-based
Mendes and Cardoso (2006)	x			Ward’s hierarchical clustering procedure, regression trees with different parameterization by experts, interactive expert selection of key cluster variables	Store location, Consumer’s behavior, demographic, socioeconomic, and geographical information	Store-based
Tehrani and Ahrens (2006)	x	x		k-means clustering, ordinal logistic regression, kernel regression model	Sales history	Store-based
Lee and Bradlow (2011)	x			K-means clustering	Customer reviews	Store-based
Mostard et al. (2011)		x		Heuristic	Sales history, Expert estimates	Entire chain, SKU-based
Brusco et al. (2012)	x			p-median clustering, model-based clustering, artificial neural network for clustering, overlapping clustering, multi-objective clustering, cluster-wise regression, tree methods	Sales history	-
Schwartz, Bradlow, and Fader (2014)		x		Classification, Regression Trees, Random Forest	Sales history	Product-based
Gallien et al. (2015)		x		Heuristic, Mixed-Integer Program	Sales history of comparable products	Store-based
Jacobs, Donkers, and Fok (2016)		x		Text analysis	Sales history, Customer characteristics	Product-based
Chen et al. (2017)		x	x	Bayesian Framework, demand censoring	Sales history	Store-based
This Study	x	x	x	k-means, Test Product Algorithm	Sales History, Store Features	Store-based
[1] Clustering, [2] Demand Forecasting, [3] Merchandise Testing						

2.5 Sales-Based Similarity Network of Stores

Based on the choice of reference characteristics, a recommendation system could be based on a content-based approach or collaborative filtering (CF) approach, or both. As their names indicate, the content-based approach is based on the “matching” of user profile and some specific characteristics of an item (e.g., the occurrence of specific words in a document), while the collaborative filtering approach is a process of filtering information or pattern based on the collaboration of users, or the similarity between items. Goldberg et al. (1992) mentioned the term collaborative filtering first to describe an email filtering system called Tapestry, designed to filter email from mailing lists and newsgroup posts. According to Tan et al. (2008), recommendation systems used for e-commerce are based on demographic information of customers or based on analysis of purchasing history of customers. As Huang, et al. (2007) stated, the most used data types for recommendation algorithms in e-commerce are product attributes, consumer attributes, and history of interactions between consumers and products such as rating or buying. CF algorithm focuses on the interaction between consumer and product and disregards the attributes of consumers and products (Zeng and Chen, 2007; Ranjan et al., 2019; Yun et al., 2018). Collaborative filtering is a technology that uses known preferences of past users to predict unknown preferences of a new user, and recommendations for the new user are based on previous estimates (Qing, 2014). According to Zhang et al. (2009), a collaborative filtering algorithm aims to recommend new items or predict the usefulness of a particular item for a particular user based on the user's previous likes and opinions of other like-minded users. Generally, consumer ratings on products, movies, videos, and songs are used as consumer-product interactions in CF algorithms.

The basis of collaborative filtering is that if two users have similar ratings on certain products, they have similar interests. (Qing, 2014). We use sales data for products in different stores and treat the stores like users in the CF algorithm. We try to calculate the dissimilarity or distance between stores using the sales performance of products. The main idea of our method is that if a product has a good sales performance in two different stores according to the average sales of those stores, similar products at those stores may perform similarly. In this case, the distance between them is small, and those two stores are very close to each other. If a product has a good performance in one store and bad performance in another, the distance between those stores is large.

In Huang et al. (2007), the performances of six collaborative filtering algorithms (user-based, item-based, dimensionality reduction, generative model, spreading activation, and link analysis algorithms) and a simple “Top-N most popular” recommendation algorithm are compared in terms of precision, recall, F score, rank score, Area Under Receiver Operating Characteristics (ROC) Curve (AUC), and computation time. Precision, positively predicted value, is the metric that shows what percentage of the values predicted as positive are actually positive. It is the ratio of 'True Positive (TP)' to the sum of 'True Positive (TP)' and 'False Positive (FP)' according to the Confusion Matrix shown in Figure 2.1. Recall is a metric that shows how many of the transactions that should be predicted as positive are predicted as Positive. It is equal to the ratio of 'True Positive (TP)' to the sum of 'True Positive (TP)' and 'False Negative (FN)'. F score is the harmonic mean of the precision and recall values (Sarwar, et al. 2000, Yang and Liu, 1999).

$$Precision = \frac{TP}{TP+FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP+FN} \tag{2.2}$$

$$F\ score = \frac{2\ x\ Precision\ x\ Recall}{Precision+Recall} \tag{2.3}$$

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 2. 1: Confusion Matrix

Wang et al. (2014) studied hybrid-based recommendation systems for movie, which uses enhanced K-means clustering combined with genetic algorithms to split transformed user space. To decrease the data space of the movie population, they used principal component analysis (PCA). The purpose of clustering they used is to divide users into similar groups to create nearest neighbors instead of searching the entire user space, which can significantly improve system scalability. They proved that cluster-based recommendation systems are better than pure CF in terms of predictive quality and efficiency. After users are clustered, the system works offline. New-arrived users are assigned to the most similar cluster and rating is predicted by using only the data in a relevant cluster instead of whole data.

Qing (2014) presented a collaborative filtering algorithm for e-commerce by defining users as neighbors with their similar interests. Rating data of users for products were used in calculations. Moreover, he used two different collaborative filtering algorithms, Rating-CF and Tag-CF, and compared their results. Rating-CF calculates the similarities by ratings; Tag-CF calculates them by tags. Tag-CF has the best performance; however, it is not so effective in the case that there are few tags provided by users. According to precision, recall, and F1 c-value, his proposed method is the best algorithm.

Zhang et al. (2009) studied an optimized item-based collaborative filtering algorithm to improve the quality of recommendations in the case of sparse data of rating. To overcome the issue of a few co-rated products, they studied an optimized item-based collaborative filtering algorithm. They calculated the percentage of the co-rated products and they used weights to calculate the similarity. They used the percentage of co-rated products, N/M, where N is the number of consumers that rated both products i and j, and M is the number of consumers that rated product i or j. They use a function $f(N/M) = 1 – \alpha*(1-N/M)$ as weight while calculating the similarity and they try to find the best α to optimize the model.

Sarwar et al. (2000) applied traditional data mining, nearest-neighbor CF, and dimensional reduction methods on two different data sets to produce practical recommendations for the customers. Data sets are obtained from an e-commerce company and MovieLens, movie recommendation site.

Nguyen et al. (2020) presented a three-layered structure that contains item layer, cognition layer and user layer. Item layer is related to the network between the items; cognition layer is about the network between the cognitive similarities of users; user layer is the network between users. They firstly calculated the item similarity, then measured the cognitive similarity and formed the nearest neighbor list. They have over 150 users and more than 5000 feedback in dataset. The results showed that cognitive similarity-based CF has more accuracy than standard approach. They have improvement of 1.8% to 3.2% in MAE calculation.

Yun et al. (2018) proposed a method for product recommendation systems using user evaluation data with opinion mining. They applied the proposed method with Amazon product data with two variants: with and without the additional opinion mining results on Amazon data. According to the comparison of precision, recall, true recommendation (TPR) and false recommendation (FPR) results, the case the opinion mining data is involved to the calculation has higher accuracy.

Tan et al. (2008) studied user based CF method for an e-learning site. They calculated the correlation between learners using rating scores. They calculated rating scores using the proportion of studying hours for a topic to the total hours spent. Then, they transformed the rating scores to corresponding rating scores, from 1 to 5. The proximity between customers is calculated using person correlation coefficient.

Kharita, et al. (2018) studied a movie recommender system using item-based CF on the MovieLens dataset. They constructed an item similarity matrix with item-similarity weights using similar ratings of movies. Then, they found similar users applying the user-based CF. By selecting K most similar items; they recommended high-rated movies to the users. They obtained 1.01 RMSE and 79.72% accuracy.

Deshpande and Karypis (2004) studied an item-based collaborative filtering algorithm by grouping items in various sets and they obtained recommender lists. They used the similarities between items and combined the similarities. This method is faster than classical user-neighborhood-based CF.

Arora et al. (2014) studied a movie recommendation system for users. They combined the existing algorithms: collaborative-based, context-based, and content-based algorithms. This recommender engine increases the performance of the recommendation system by coming through the disadvantages of the traditional recommender systems.

Ponnam et al. (2016) studied an Item-based CF over the Netflix dataset. They created the user-item rating matrix and calculated the similarities between items with the cosine similarity method. By using the relationships between items, they recommend the items to the users. To calculate the predicted rating for a movie by the user who did not rate for a particular movie, they used the similarities between items as weights. Then, they recommended the top N items with the highest predicted rating to the users.

$$r_{x,i} = \frac{\sum_{j \in N(i;x)} s_{i,j} * r_{i,j}}{\sum_{j \in N(i;x)} s_{i,j}} \tag{2.4}$$

In Eyjolfsdottir, et al. (2010), a movie recommendation system, MovieGen, was introduced. They presented a hybrid recommendation method that contains machine learning and k-means clustering. They studied on Support Vector Machine (SVM) model by taking users' personal information and making movie predictions for them. They clustered the movies, asked the questions to the users, and according to answers and user's personal information, they recommended the movies to the users.

Werner (2020) analyzed the music recommendation systems used in Spotify. He focused on the similarities of songs over genre and gender. He analyzed the three functions of Spotify: related artists (suggested artists according to listened artists by users), discover (personalized suggestions in browse page), and browse page (highlighting some playlists, new releases, and genres). According to the paper, Spotify aims to simplify listening to music by promoting similarity, emphasizing what is currently popular, and building on a genre system.

Gomer-Uribe and Hunt (2015) discussed several algorithms in the Netflix recommender system. “Personalized Video Ranker (PVR)” provides recommended movies according to the genre in each row on the home page. “Top-N Video Ranker” recommends the top n movies for different users related to their watching history. The third one is “Trending Now” that recommends the movies based on the short-term trends like Valentine’s Day. In the “Continue Watching” row, there are recommended movies to resume watching or rewatch. Those movies may have been left at the beginning, middle, or end of the program. “Because You Watched (BYW)” rows present the movies based on video-video similarities. The similarity is calculated according to the Netflix catalog without personalization, but the subset of the BYW list is shown based on the personalization depending on the similar videos the user enjoyed.

Benati and Garcia (2014) studied a clustering problem that tried to cluster people who answered a survey. There is a set, U, that contains people answering the survey. There is a set, F, which contains questions in the survey. The aim is to select a set, Q, the subset of F and to cluster people over the questions in the set Q. They provided a combinatorial model for clustering to select the best $Q \subset F$ to minimize the total distance between elements in the clusters and centroids of clusters. They used dijk to determine the distance between person i and person j with respect to question k. There is a variable zk, that is 1 if feature $k \in Q$.

They proposed a mixed-integer nonlinear optimization model for selecting the best subset of variables and the best set of medians for p clusters such that the total distance between the median and the units in each cluster is minimized, which is an extension of the p-median problem such that it considers only the most important variables out of all variables involved. They propose a direct linearization and a radius formulation to linearize the model and show that the radius formulation can be solved much faster than the classical p-median and direct linearization formulations.

To sum up, CF is one of the most used methods in recommendation systems. As seen above, CF is used in many areas. E-commerce, movie sites, e-learning, surveys, music are the some of these areas. The main logic in the CF is to obtain a similarity between users in terms of their ratings for the items (movie, video, song, product, etc.). Pearson correlation coefficient is the one of the most used similarity coefficients.

$$sim(u,v) = \frac{\sum_{i \in C_{u,v}}(R_{u,i}-R_u)(R_{v,i}-R_v)}{\sqrt{\sum_{i \in C_{u,v}}(R_{u,i}-R_u)^2 \sum_{i \in C_{u,v}}(R_{v,i}-R_v)^2}} \tag{2.5}$$

Where R_(u,i) is the rating of user u on the item i; R_(v,i) is the rating of user v on the item i; C_(u,v) is the set of common items rated by the user u and v. R_u is the average rating given by user u, and R_v is the average rating given by user v. In the most cases where users rate the items, users give the rates within a certain scale. Five and ten are the most used rating scales.

In our study, we try to determine the dissimilarity between stores using the sales of products sold in these stores. Since the range and average of sales quantities may differ from store to store, we use z value for the products in each store. Therefore, we can determine the performance of products in the relative store in terms of the mean and the standard deviation of sales quantities in this store. We find how many standard deviations the sales quantity of the product is from the mean.

$$z = \frac{x-\mu}{\sigma} \tag{2.6}$$

3. METHODOLOGY

Four fundamental methods will be developed in this thesis as shown in Figure 3.1 in relation to each other. Firstly, “Trend Analysis” is done by category managers. They determine the place of trends in the BCG Matrix based on customer profiles. Then, in the “Merchandise Testing” phase, we will develop a new method for merchandise testing process for the products with new trends. Thirdly, in the “Sales-Based Similarity Network Algorithm” phase, we will work on a store-based network algorithm by setting up a network of stores based on the sales performance of products in stores. Then, we will use a clustering similar to k-means clustering using the store-based network. Furthermore, we will use the results of that clustering study in the “Merchandise Testing Method”. Fourthly, in the “Distribution Algorithm” phase, we will work on an optimization algorithm including integer programming in order to distribute the region-based collection into stores. We will try to provide product attribute rates in the region-based collection. Finally, we will develop a prediction algorithm for new products that may be designed by designers, obtained from the competitor analysis, and obtained from social media. The prediction algorithm will predict the sales quantities of those new products for each store. This will help category managers select the best products to be included in the collection. In addition, we will use these predicted sales quantities in the objective function of “the distribution algorithm.”

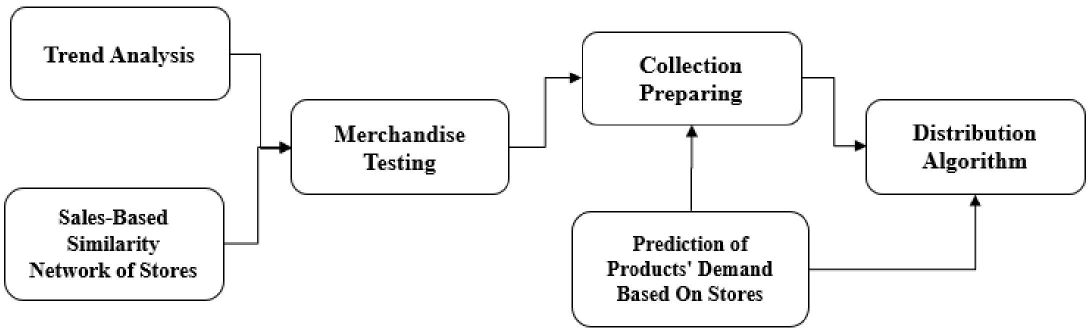

Figure 3. 1: Relationship between the methods to be developed in the thesis

4. STORE-BASED MERCHANDISE TESTING FOR APPAREL RETAIL INDUSTRY

4.1 Introduction

Balancing supply and demand to avoid mismatches and inventory costs due to either insufficient or excess supply is a common concern across industries. This balancing act gets more challenging when the products have short lifecycles and uncertain demand, as in apparel retail. The apparel retailing industry offers many new products with mostly short lifecycles every season, in addition to the basic products with predictable demand. Although there are several successful methods for forecasting the demand for basic products, such as black or white t-shirts, predicting the demand for new products is more challenging and thus crucial in reducing inventory and operating costs. Fashion trends are critical for fashion retail since they guide the key decisions about processes such as production, marketing, and logistics. Every season, new trends occur, some of which become more prevalent, and some disappear in the fashion world. Fashion retail companies follow fashion trends and design products that contain trends properly. If a brand does not have collections for all customer types, it prepares the collections at the right stage of the trend lifecycle. Many brands strive to understand their customers' demand for new trends and need a system to test the new products and determine the real potential at the beginning of the lifecycle. The merchandise planning process involves the activities to determine the new products, their order quantities, prices, and distribution to stores to ensure that the right products are available for the right consumers at the right time, place, quantity, and price (Rajaraam, 2008). This study focuses on selecting test stores and sales forecasting based on test sales within the scope of merchandise planning.

Companies do not have enough information about new trends until they put the products on the market. It is easier to predict the potential of basic products or products that already have a sales history. Correctly estimating the demands of newly launched products or services is very important from a managerial point of view (Steenkamp et al., 1999; Stremersch and Tellis, 2004; Van den Bulte and Stremersch, 2004). The lack of sufficient data makes forecasts for fashion products less accurate (Choi et al., 2014; Green and Harrison, 1973; Sichel, 2008; Sun et al., 2008). Due to the unpredictability of customer preferences and the inability to react quickly, overstocking and stockout costs arise

(Rajaram, 2008). According to Fildes et al. (2022), there are three methods of forecasting the demand for new products: (i) the judgment of experts based on experience, (ii) using market research, and (iii) a comparable product approach. Gallien et al. (2015) analyze the stores based on the sales history of comparable products to forecast the demand for new products of Zara. However, it is difficult to obtain the potential popularity of the trends for brands. It may be a new trend that the brand has no idea about the potential of, or they might have sales data in different categories. For instance, the brand may have sold pink t-shirts, but they may have no idea about the potential of pink pants. In this case, they need to test pink pants. Moreover, brands need to test new categories for which they do not have sales data. For instance, a men's brand might start selling products in the women's category. In this case, they cannot use their past sales data for new categories. They need to do market research first and then need to test new products extensively. Below, we summarize the relevant literature on merchandise testing and sales forecasting in the apparel industry.

In this topic, the goal is to predict the sales quantity of new products at the store level. Therefore, the features of stores and the method used in the selection of test stores play significant roles in our study. A small number of test stores should be selected to reduce the investment cost into new products with unknown demand. The previous term's sales can be analyzed to predict a new product's potential sales based on stores. Product attributes and store features may be useful for this analysis and prediction. Since the sales quantity of a product in all stores is predicted by the sales data obtained from test stores, test stores should be representative of the whole store chain. Therefore, test stores are selected to represent all stores in terms of feature distribution after clustering all stores based on the similarities of their features. An integer programming model named Test Store Selection (TSS) is formulated to select the best test stores to represent the store feature distribution. To analyze the test product performance and predict the potential for all stores, we develop a forecasting algorithm that focuses on the stores where test products are put up for sale and analyzes the sales performance of the test products based on the features of the store such as store size and trend level, and the features of the region where the store is located, such as population, number of houses, and average income. This method uses the relation between features and sales to predict the sales amount for the remaining stores based on the test stores.

4.2 Data and Methods

The goal of the merchandise testing process is to predict the potential of a product based on its performance in the selected test stores. Store features are critical input parameters to predict the sales potential of a product for all stores based on the performance in the test stores. Therefore, store features and their values for each store must be determined, and stores must be clustered based on their similarities. Store features used in this study and the method used for clustering stores are explained next.

4.2.1 Store features and clustering

First, the store features that may affect the sales of a product were determined. Data for the store features were collected from the Turkish Statistical Institute (TUIK) and market research companies. The coordinates of stores were marked on the map. Then, for each store, in the area with a 10-kilometer diameter, certain demographic information was collected. Location-based store features are listed in Table 4.1.

Table 4. 1: Store features used

Description and Units	
Categorical Store Features	
Region	Geographic region
Climate	Climate classification of the store region
Education Level	Categorized by the company according to the number of university graduates living in the area
Store Size	Categorized by the company according to the size of the stores in m^2
Trend Level	Categorized by the company according to the sales rates of trendy products
Numerical Store Features	
Sales Turnover	Sales amount
Population	Number of residents in the area
Number of Residences	Number of residences
Number of Workplaces	Number of workplaces
Working Population	Number of working individuals
Customer Profile	Population size of the specified customer profile
Rent of Houses	Average rent of houses in the store area in TL
Income	Average income per person in TL
Clothing Expense	Average clothing spending per person in TL
Restaurant Expense	Average restaurant spending per person in the area in TL

Entertainment Expense	Average entertainment spending of the area population in TL
Transportation	Number of public transportation stops in the area
Competitor A/B/C	Number of Competitor A/B/C stores in the area

The aim is to choose test stores such that they accurately represent the feature distribution of all stores. Integer programming is used to formulate test store selection problem where the store feature distributions are defined as constraints. In this integer programming model formulation, using categorical data is more practical; hence, numeric data collected for each store are converted to categorical data. Moreover, categorical data makes it easier to identify the characteristics of a new store based on the area it is planned to be opened in, and new stores can be easily added to the store pool for future uses of this method. In the reality in fashion retail, category managers may need to manually categorize newly opened stores that do not have sufficient data.

In addition to the demographic store features mentioned above, which are affected by external factors, features about the company's performance, affected by internal factors, such as sales region, customer profile, size, and turnover, need to be determined. After adding these to the store feature list, clustering is applied for all features.

The store features used in the merchandise testing study are chosen according to the category. For instance, if the category is "men-casual", then the features that are not related to this customer group are not used in the model. The number of retired men and women, the population for the ages below 15 and above 35, or any other feature that is not related to the "men-casual" group are not used.

Stores were clustered based on each store feature using the k-means clustering method. The k-means algorithm is one of the basic unsupervised clustering algorithms developed in 1967 (MacQueen, 1967). It works for a predetermined number of clusters. The algorithm defines k centroids, one for each cluster. Those centroids should be as far as possible from each other in terms of Euclidean distances. Then, each point in the data is evaluated and assigned to the nearest cluster. After each point is placed in a cluster, the center of that cluster is recalculated. This step is repeated until all the points are considered. When the clusters of points no longer change, the algorithm is terminated.

The k-means clustering method is used to determine store clusters based on each numeric store feature. Our aim is not to cluster the stores but to categorize numeric data by converting numeric data into categorical data. Numeric data in Table 2 is converted to categorical data using k-means. Different numbers of clusters from 1 to 20 are examined to find the best number of clusters for each feature that provide both a satisfactorily low total distance to cluster centers and ease of use in terms of computational effort. In Figure 4.1, the total distance of stores to their cluster center is presented for different numbers of clusters for numeric store features. Distance is at its maximum when the number of clusters is 1. We used the Elbow Method to determine the most suitable number of clusters. Elbow Method is a heuristic method to determine the number of clusters in a data set in cluster analysis. The method consists of plotting the variation as a function of the number of clusters and choosing the elbow of the curve as the number of clusters to be used (Purnima&Arvind, 2014; Humaira&Rasyidah, 2020).

While the number of clusters is increasing, distance decreases; however, beyond 7 clusters, distance does not decrease considerably. We looked into the cases with 5-8 clusters. When the number of clusters is 5, there would be 91 distinct store feature values, and the number of feature values becomes 105 for 6 clusters, 119 with 7 clusters, and 133 with 8 clusters. The number of store feature values is important for the test store selection stage because it is harder to capture the ratio of store feature values represented by the test stores as the number of feature values increases. The number of constraints in the integer programming model increases as the number of clusters increases. Furthermore, the Mean Absolute Percentage Error (MAPE) results for ratios of store feature values show that choosing 5 clusters is more appropriate than a higher number of clusters. MAPE value in 5 clusters is 8%, whereas it is 18% in 6 clusters, 24% in 7 seven clusters, and 29% in 8 clusters. Moreover, 5 clusters are suitable to use for both the current stores and stores to be opened in the future, as it is easier to assign the cluster of a store in terms of store features for business experts when there is no data about newly opened stores.

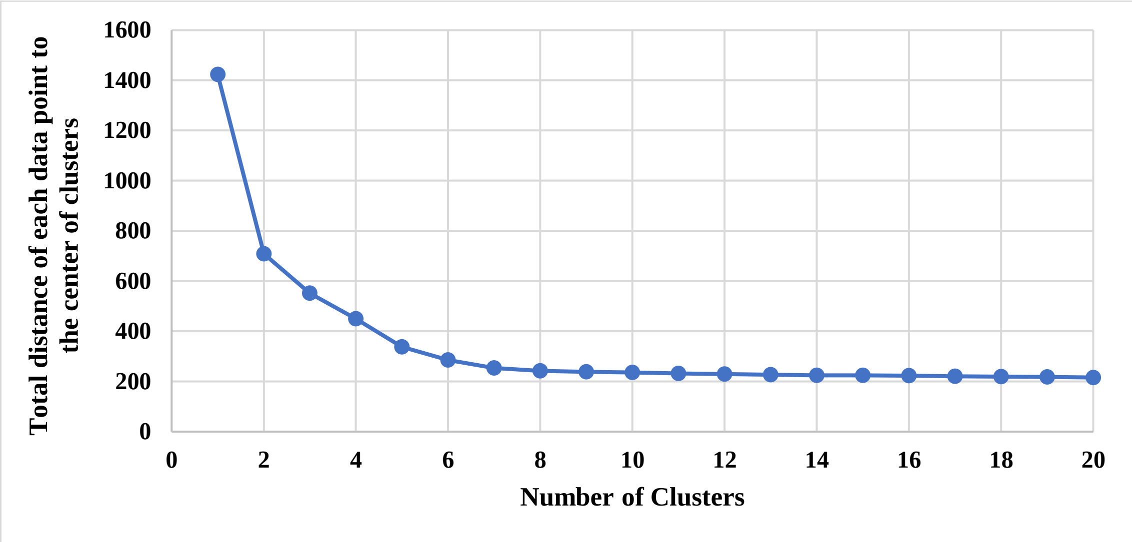

Figure 4. 1: Total distance of stores to their cluster centers for different numbers of clusters

Total distance of stores to their cluster centers for different number of clusters for some store features is shown in Table 4.2.

Table 4. 2: Total distance of stores to their cluster centers for different numbers of clusters

Clusters	Population	Number of Residences	Number of Workplaces	Customer Profile	Rent of Houses	Income	Clothing Expense	Restaurant Expense	Entertainment Expense
C1	399,857	153,384	108,728	35,314	19,064	146,415	4,770	7,128	3,666
C2	230,840	84,131	68,768	19,673	11,312	86,792	2,655	3,799	2,023
C3	168,853	57,441	50,332	14,452	8,926	60,826	2,076	2,910	1,394
C4	122,017	43,966	34,424	10,460	7,590	48,649	1,490	2,290	1,172
C5	100,782	36,437	28,280	8,304	6,176	42,802	1,278	1,828	904
C6	87,229	31,464	24,609	7,235	4,568	26,472	1,082	1,620	837
C7	79,082	26,541	22,763	6,672	3,824	22,823	932	1,457	654
C8	75,563	22,824	20,753	6,318	3,380	21,011	794	1,388	602
C9	71,315	20,418	20,289	5,376	2,897	18,900	645	1,006	558
C10	51,276	18,137	20,053	4,451	2,550	16,626	667	868	467
C11	46,829	17,168	14,120	3,932	2,214	14,587	558	798	412
C12	42,181	16,359	12,520	3,648	2,054	13,184	514	722	391
C13	39,065	14,016	12,237	3,264	1,931	12,752	470	681	372
C14	37,335	12,784	12,137	3,025	1,834	11,492	426	657	365
C15	36,547	12,313	12,088	2,716	1,769	11,259	398	624	359
C16	35,475	11,429	12,080	2,672	1,743	10,959	368	620	293
C17	34,161	10,715	12,032	2,575	1,622	10,930	350	524	281
C18	33,713	10,513	11,914	2,500	1,427	10,267	344	518	259

C19	32,334	9,778	11,886	2,428	1,419	8,687	304	457	232
C20	32,285	9,642	11,879	2,421	1,214	7,335	297	449	230

The charts for distance versus the number of clusters for the other numeric store features show a similar pattern. Therefore, we take the number of clusters as 5 for all numeric store features. Some features have less than five different values already, so they are not clustered using k-means. For example, there are only four different regions and four different store sizes already, which makes clustering redundant. In Table 4.3, there are the minimum, average, and the maximum numbers of clusters for numeric store features. It should be noted that the minimum population size for cluster 1 is 1, which means there is a store outside the city in an unpopulated area.

Table 4. 3: Store features and value ranges

		CLUSTERS				
Store Features		**1**	**2**	**3**	**4**	**5**
Number of Houses (quantity)	min	14	3,109	6,076	9,892	14,473
	average	1,481	4,543	7,576	11,939	17,576
	max	3,069	6,026	9,672	14,285	24,641
Number of Workplaces (quantity)	min	2	1,148	2,859	5,366	9,430
	average	440	1,827	3,821	6,704	14,293
	max	1,134	2,761	5,229	8,778	32,034
Population (number of people)	min	1	567	1,232	2,296	3,725
	average	257	856	1,600	2,850	4,857
	max	564	1,209	2,116	3,633	6,590
Rental Expenses (TL)	min	404	926	1,316	1,787	2,364
	average	759	1,093	1,539	2,022	2,807
	max	923	1,299	1,768	2,304	4,802
Income (TL)	min	1,270	3,362	6,275	10,714	16,833
	average	2,560	4,140	7,810	12,254	20,855
	max	3,339	5,510	9,272	14,233	35,793
Clothing Expenses (TL)	min	33	163	246	353	515
	average	118	205	283	420	638
	max	162	244	343	506	885

4.2.2 Test store selection

The product to be tested should be sent to a subset of stores that would provide an accurate representation of all stores of the brand. In order to select the test stores, an integer programming model is used. The goal is to select test stores such that the ratios of store feature values in test stores are as close to the ratios of store feature values in all stores as possible.

The aim is to select the best subset close to the feature distribution of all stores. For instance, if the ratio of Cluster 1 in “Income” is 10% of all stores, we aim to provide nearly 10% of the test stores in the region corresponding to Cluster 1 in “Income”. If the number of test stores is 50, 5 of them should belong to Cluster 1.

The notation used in the following discussion is provided below.

Notation

$M = \{1,2, \dots, m\}$: the set of stores

$N = \{1,2, \dots, n\}$: the set of store features

S: the $m \times n$ matrix of store features, where m is the number of stores, n is the number of features.

$S_j = S(:, j)$: the $m \times 1$ column vector for store feature j for all stores, $j = 1,2, \dots, n$.

U_j: the set of unique values in column vector S_j.

$d_j = |U_j|$: the number of unique elements in column j of the S matrix such that $d_j \leq m$.

$D = \sum_{j=1}^{n} d_j$: the total number of unique values of all store features.

SE: an $m \times D$ matrix of zeros and ones obtained by one-hot encoding, converting the columns in matrix S by defining a separate column for each different value of each store feature.

$U_j(k_j)$: the column of SE for feature $j = 1,2, \dots, n$ and feature value $k_j = 1,2, \dots, d_j$.

SE_{ik}: the binary value corresponding to store i feature value k.

TS: the maximum number of test stores that can be selected.

r_i: the priority/importance coefficient of store $i = 1,2, \dots, m$. Initially assumed equal for all stores, i.e., $r_i = 1, \forall i$. However, store location, proximity to the headquarters, past sales volume, etc., could be factors that increase the priority of certain stores.

x_i: the binary decision variable that takes on value 1 if store i is selected as a test store and 0 otherwise.

Based on the notation above, note that each row sum for the SE matrix is equal to n:

$$\sum_{k=1}^{D} SE_{ik} = n, \forall i \in M \tag{4.1}$$

Since, for a given feature, a store can be only in one of the feature clusters, it can only take one of the unique feature values, i.e.,

$$\sum_{k=\underline{k}}^{\underline{k}+d_j} SE_{ik} = 1, \forall i \in M, j \in N \tag{4.2}$$

where $\underline{k} = \sum_{a=1}^{j-1} d_a + 1$.

In Table 4.4, an example of the binary form of the store feature matrix SE is presented. For example, Store 1 takes on the first feature's first value; therefore, there is an entry of 1 under the first column related to the first feature and entries of 0 under the other columns related to the first feature.

Table 4. 4: Representation of the binary form of store feature matrix SE

	Binary values related to a_1				Binary values related to a_2					Binary values related to a_n		
Unique feature value	$U_1(1)$	$U_1(2)$		$U_1(d_1)$	$U_2(1)$	$U_2(2)$		$U_2(d_2)$		$U_n(1)$		$U_n(d_n)$
Column index in matrix SE	1	2	…	d_1	$d_1 + 1$	$d_1 + 2$	…	$d_1 + d_2$	…	$\sum_{j=1}^{n-1} d_j + 1$	…	D
Store 1	1	0	…	0	0	1	…	0	…	0	…	1
Store 2	0	0	…	0	0	0	…	0	…	0	…	0
Store 3	0	1	…	0	0	0	…	0	…	0	…	1
Store 4	1	0	…	0	1	0	…	0	…	1	…	0
…												
Store m	0	0	…	1	0	0	…	1	…	0	…	1

The Test Store Selection (TSS) mathematical model formulation is as follows.

$$\text{Maximize} \qquad z = \sum_{i=1}^{m} r_i x_i \tag{4.3}$$

Subject to

$$\sum_{i=1}^{m} (SE_{ik} * x_i) \leq TS * \left(\frac{\sum_{i=1}^{m} SE_{ik}}{m}\right) \quad \forall k = 1,2,\ldots,D \tag{4.4}$$

$$\sum_{i=1}^{m} x_i \leq TS \tag{4.5}$$

$$x_i \in \{0,1\}, \forall\, i \in M \tag{4.6}$$

The objective function (4.3) of this binary program is to maximize the total number of test stores, and it is reduced to $z_1 = \sum_{i=1}^{m} x_i$ based on the initial assumption of the equal importance of stores such that $r_i = 1, \forall i \in M$. When the stores are equally important z_1 will be equal to TS. If different r_i values are used based on proximity to the center or other criteria decided by the management, this integer program finds the maximum objective function value subject to the constraints.

Constraint (2) ensures that the number of test stores that have a certain value k for a feature does not exceed the existing number of such stores in total, where the ratio of stores with a certain feature value is represented by $\left(\frac{\sum_{i=1}^{m} SE_{ik}}{m}\right)$. Constraint (3) limits the number of test stores by the specified maximum number of test stores. Constraint (4) defines the binary decision variables of the model.

4.2.3 Sales forecasting for non-test stores

After selecting z_1 test stores, let T be the set of test stores, the potential demand for the tested product will be determined for the remaining $m - z_1$ stores in set $M \setminus T$.

The original matrix S of store features is partitioned into two to distinguish the test stores from the others. Let S_T be the store feature matrix for test stores such that it is a $z_1 \times n$ matrix and S_P be the store feature matrix for the other stores for which the demand will be predicted such that it is an $(m - z_1) \times n$ matrix.

The demand for a product at a certain store can be affected by not only a single feature but a combination of a subset of store features. These subsets can include at least 2 and at most n features. Therefore, the store feature matrix is expanded to include all possible combinations of 2 or more of the existing n store features. For the test stores, S_T becomes S_T^C, and for the stores whose demand is to be forecasted, S_P becomes S_P^C with $n^C = \sum_{j=1}^{n} \binom{n}{j}$ columns.

After the S_T^C and S_P^C matrices are obtained, the sales in test stores are analyzed. We use average sales quantity for unique elements in S_T^C matrix. For example, there are z_1 elements in the q^{th} column of the S_T^C matrix. Let there be d_q different elements in this column, where $d_q \leq z_1$. We calculate average sales quantities for d_q unique elements in the q^{th} column where $1 \leq q \leq n^C$.

After calculating the average sales for S_T^C matrix, the importance coefficients are calculated for each column. These importance coefficients are used to calculate weighted average sales quantities for forecasting the demand of the non-test stores. To calculate importance coefficients, standard deviation and information gain methods are used. First, the standard deviation for each column q is calculated. Then, the information gain for each q is obtained by subtracting the standard deviation of the column from the overall standard deviation for test stores, as explained in detail below.

There are d_q unique elements in column q and to calculate the standard deviation for this column, the standard deviation of sales quantities of unique elements in the q^{th} column, σ_{qe}, and the proportion of these unique elements in that column, P_{qe}, are calculated. Then, using these proportions as the weights of the element-wise standard deviations, the standard deviation for the column is found as follows.

$$\sigma_q = \sum_{e=1}^{d_q} P_{qe} * \sigma_{qe}\,, \ \forall q = 1,2, \dots n^C \tag{4.7}$$

Letting σ be the overall standard deviation of sales quantities for the S_T^C matrix, the importance coefficient for column q is calculated as the standard deviation reduction as follows.

$$c_q = \sigma - \sigma_q, \ \forall q = 1,2, \dots n^C \tag{4.8}$$

Then, the weight for each column is calculated as the ratio of its importance coefficient to the sum of the importance coefficients of all columns of the S_T^C matrix as follows.

$$w_q = \frac{c_q}{\sum_{q=1}^{n^C} c_q}, \ \forall q = 1,2, \dots n^C \tag{4.9}$$

The average sales quantity for the non-test stores are calculated as a weighted sum of the average sales quantity for elements in the test stores using w_q as the weights.

We have z_1 rows, one for each test store in the S_T^C matrix, and we have the sales data for each of these test stores. Each test store is classified to have certain values for each store feature. Therefore, we can find the average sales of stores that have a certain feature value individually. Let these average sales quantities be $\bar{S}_{Tql}, \forall q = 1,2, \dots n^C, \forall l = 1,2, \dots, d_q$. We match these average sales quantities with the non-test stores that have the same feature values and define this new matrix of the rate of sales as R. Then, using the column weights, w_q, the sales of non-test stores are forecasted as the weighted sum of these average sales quantities in the R matrix. The resulting vector of forecasted sales is defined as follows.

$$F_i = \sum_{q=1}^{n^C} w_q * R_{iq}, \ \forall i \in M \backslash T \tag{4.10}$$

A flowchart of the methodology explained above for store clustering and sales forecasting is provided in Figure 4.2. The algorithm explained above is applied using Python and the computational results are presented in the next section. Moreover, the result of the proposed algorithm is compared with the results of Linear Regression, Decision Tree, Gradient Boosting, Extreme Gradient Boosting, and SVM algorithms in terms of MAPE.

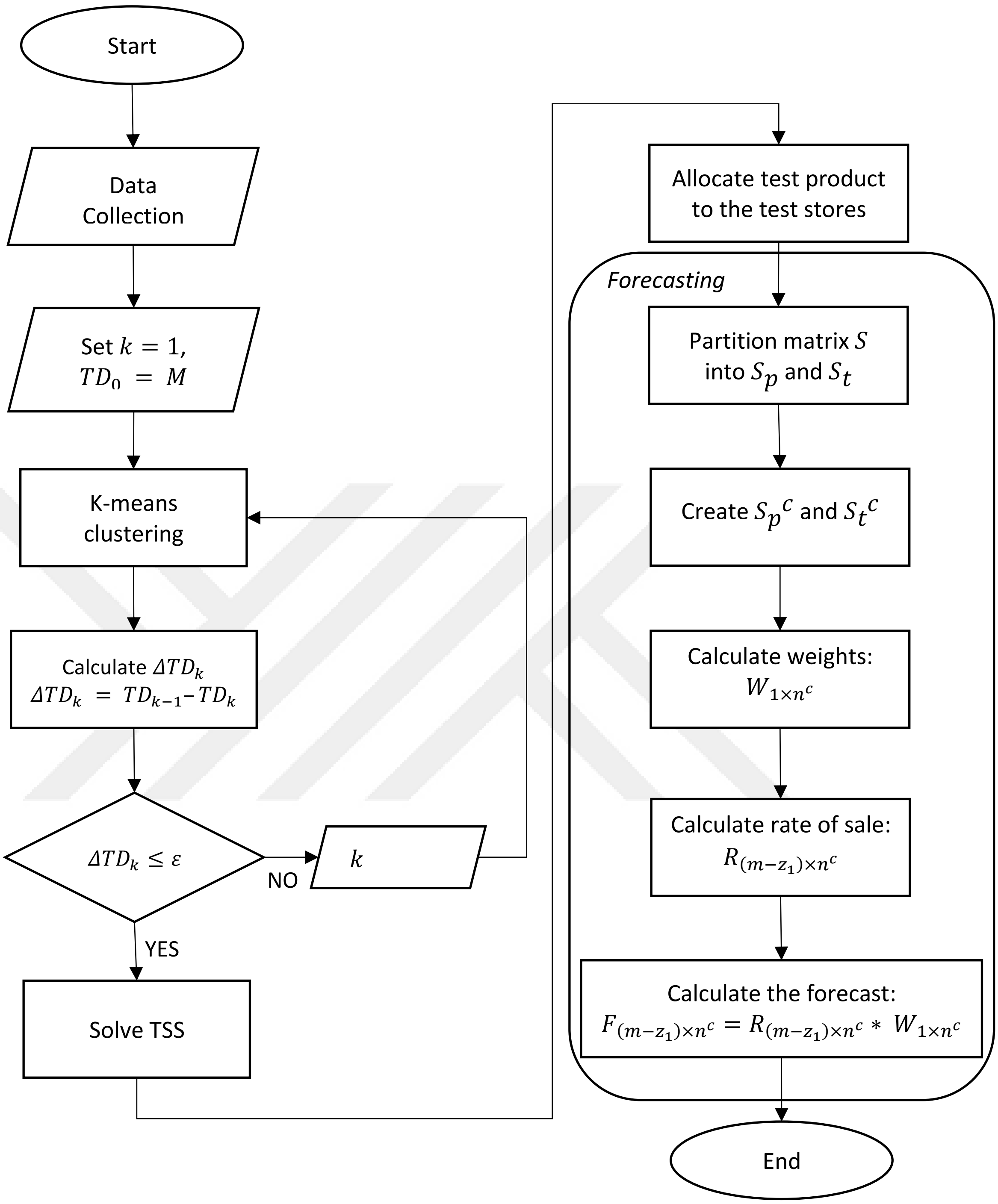

Figure 4. 2: Flowchart of clustering, test store selection, and sales forecasting algorithm

4.2.4 An example for sales forecasting for non-test stores

Let us assume we have data for three store features. Store features are A, B and C. There are three store feature values under each store feature: A1, A2, A3, B1, B2, B3, C1, C2 and C3. There is an example table for 10 stores and 3 store features in Table 4.5.

Table 4. 5: Example data for 10 stores and 3 store features

Stores	A	B	C	Sales Quantitiy
S1	A1	B3	B3	8
S2	A2	B3	B3	9
S3	A1	B2	B1	6
S4	A3	B3	B1	6
S5	A2	B1	B2	5
S6	A3	B2	B3	3
S7	A1	B3	B2	1
S8	A1	B3	B1	3
S9	A2	B1	B3	6
S10	A3	B1	B3	1

S_T^C matrix explained above is similar to the matrix shown in Table 4.6. There are columns for combination of store features.

Table 4. 6: The matrix similar to S_T^C

A	B	C	A_B	B_C	A_C	A_B_C
A1	B3	C3	A1_B3	B3_C3	A1_C3	A1_B3_C3
A2	B3	C3	A2_B3	B3_C3	A2_C3	A2_B3_C3
A1	B2	C1	A1_B2	B2_C1	A1_C1	A1_B2_C1
A3	B3	C1	A3_B3	B3_C1	A3_C1	A3_B3_C1
A2	B1	C2	A2_B1	B1_C2	A2_C2	A2_B1_C2
A3	B2	C3	A3_B2	B2_C3	A3_C3	A3_B2_C3
A1	B3	C2	A1_B3	B3_C2	A1_C2	A1_B3_C2
A1	B3	C1	A1_B3	B3_C1	A1_C1	A1_B3_C1
A2	B1	C3	A2_B1	B1_C3	A2_C3	A2_B1_C3
A3	B1	C3	A3_B1	B1_C3	A3_C3	A3_B1_C3

Forecasting method starts with calculating the average sales quantity for store feature values. For example, S1, S3, S7 and S8 stores have store feature values, A1 under the A feature. The average sales units for “A1” is 4.5. Calculations of average sales units for each store feature value are done in S_T^C matrix.

Then, we calculate the weights for each column in S_T^C matrix. According to the equation (8), we find the standard deviation for all the stores in S_T^C. σ is 2.6 in this example. Moreover, we calculate the σ_q for all of the columns in S_T^C. σ_q for store feature A is calculated as follow:

The sales quantities for stores with A1 are 8, 9, 3 and 1; standard deviation is 3.34. Standard deviation for A2 is 1.7 and for A3 is 2.05. The frequency of A1 in the table is 0.4, and this values is equal to P_{qe}. P_{qe} for A2 and A3 is 0.3. Using the formula (7), we find the σ_q for store feature A is $\sigma_q = 0.4 * 3.34 + 0.3 * 1.7 + 0.3 * 2.05 = 2.46$. Information gain or reduction in standard deviation for store feature A is $c_q = 2.6 - 2.46 = 0.14$.

c_q and w_q values are calculated for each column in S_T^C. In the equation (10), we calculate the forecast for each store. For example, we take the average sales quantities for A1, B3, C3, A1_B3, A1_C3, B3_C3 and A1_B3_C3 for S1 store. These average sales quantities refer to R_{iq} in equation (4.10).

4.3 Results and Discussion

The methodology explained above was applied in a fashion retail company. The company has 364 stores, and 20 store features are used as input. The features and their values are shown in Table 4.7. Input variables are categorical variables. The "Region" feature is the geographical region that the company uses in business. There are four different regions. "Size" is determined by the square meter size of the stores. Also, there are sales quantities of the test product as a dependent variable. Persona, Trend Level, Population, Number of Houses, Number of Plazas, Education Level, Working Population, Customer Profile, Rent of Houses, Income, Clothing Expense, Restaurant Expense, Entertainment Expense, and Transportation are grouped based on the numerical data. The values of Competitor A, Competitor B, and Competitor C are the number of competitors' stores in the related region of the store of the company, where G0 symbolizes zero stores, G1 means one store, and G2 means two stores. The "Climate" variable is gathered from Bölük (2016) according to the Köppen (1936) climate definition of the district where the stores are located.

Table 4. 7: Features and their possible values

Features	Value 1	Value 2	Value 3	Value 4	Value 5
Region	Region 1	Region 2	Region 3	Region 4	
Size	G1	G2	G3	G4	
Persona	G1	G2	G3	G4	G5
Trend Level	G1	G2	G3	G4	G5
Climate	Cfak	Csak	Cshk	Dcak	Dcbo

Population	G1	G2	G3	G4	G5
Number of Houses	G1	G2	G3	G4	G5
Number of Plazas	G1	G2	G3	G4	G5
Education Level	G1	G2	G3	G4	G5
Working Population	G1	G2	G3	G4	G5
Customer Profile	G1	G2	G3	G4	G5
Rent of Houses	G1	G2	G3	G4	G5
Income	G1	G2	G3	G4	G5
Clothing Expense	G1	G2	G3	G4	G5
Restaurant Expense	G1	G2	G3	G4	G5
Entertainment Expense	G1	G2	G3	G4	G5
Transportation	G1	G2	G3	G4	G5
Competitor A	G0	G1	G2		
Competitor B	G0	G1			
Competitor C	G0	G1	G2		

Four scenarios with different numbers of test stores were considered. The method was applied to seven different products. Also, five data science methods, namely Linear Regression, Decision Tree, Gradient Boosting, Extreme Gradient Boosting, and Support Vector Machine, were used, and their results were compared. Since our dependent variable is numerical and input variables are categorical, the data science methods described above are selected as appropriate methods for comparison. Due to the fact that our data is not a time series and the aim of the forecast is to predict the sales of stores using store features as input variables, we do not use generic forecasting methods such as Moving Average, Weighted Moving Average, or Exponential Smoothing. We implement the above data science methods in Python with the library and parameters shown in Table 4.8.

Table 4. 8: Library and parameters of data science methods used in Python

Method	**Library**	**Parameters**
Decision Tree	sklearn.tree.DecisionTreeRegressor	(*, criterion='squared_error', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, ccp_alpha=0.0)
Linear Regression	sklearn.linear_model.LinearRegression	(*, fit_intercept=True, normalize='deprecated', copy_X=True, n_jobs=None, positive=False)

Gradient Boosting	sklearn.ensemble.GradientBoostingRegressor	(*, loss='squared_error', learning_rate=0.1, n_estimators=100, subsample=1.0, criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_depth=3, min_impurity_decrease=0.0, init=None, random_state=None, max_features=None, alpha=0.9, verbose=0, max_leaf_nodes=None, warm_start=False, validation_fraction=0.1, n_iter_no_change=None, tol=0.0001, ccp_alpha=0.0)
Extreme Gradient Boosting	sklearn.ensemble.GradientBoostingRegressor	(*, loss='squared_error', learning_rate=0.1, n_estimators=100, subsample=1.0, criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_depth=3, min_impurity_decrease=0.0, init=None, random_state=None, max_features=None, alpha=0.9, verbose=0, max_leaf_nodes=None, warm_start=False, validation_fraction=0.1, n_iter_no_change=None, tol=0.0001, ccp_alpha=0.0)
Support Vector Machine	sklearn.svm.SVR	(*, kernel='rbf', degree=3, gamma='scale', coef0=0.0, tol=0.001, C=1.0, epsilon=0.1, shrinking=True, cache_size=200, verbose=False, max_iter=-1)

4.3.1 Determining test stores

One of the fundamental aspects of the test product method is to allocate the test products to the test stores. The aim is to make a forecast for the stores other than test stores based on the sales performance in the test stores. In merchandise testing, the number of test stores should be few. Because of that, the number of test stores is less than the number of other stores. In the prediction algorithms, we should divide the data into training (nearly 75%) and test (nearly 25%) data. In this case, our training part is nearly 10% of the total data. Therefore, we should determine the training data (our test stores) in a controlled way. In this study, the data is available for 364 stores. We used four different test store numbers: 25, 50, 75, and 100.

To determine the test stores that can be representative of all stores, we first calculated the ratios of store features in the data for all stores. We have nearly 20 features that actually change by category group. We want to find test stores that are representative of the store population, such that the test stores provide the ratios of features as close as possible to the ratios in the data for all stores. Thereby, we would have a few test stores as a small sample. For instance, we have 10% of a total of 364 stores as the 5th cluster in terms of average household income. Then, we should have nearly 50*10% = 6 stores in test stores as the 5th cluster in average household income according to the case for 50 test stores. We should obtain those ratios for all values under 20 store features.

There are as many constraints as the product of the number of features and the number of feature values. There are 20 features for the category group and 91 store feature values; therefore, there are 91 constraints. In addition, we have a constraint for the maximum number of test stores to be chosen in this case. By this binary integer programming, explained in equations (1)-(4), we obtained the best 25, 50, 75, and 100 test stores, which have the closest ratio of features to the ratio in the data for all stores. In Figure 4.3, the percentages of store feature values are shown for 364 stores and different numbers of test stores. For each case, the ratios of test store feature values are satisfactorily close to the ratios in the entire store set.

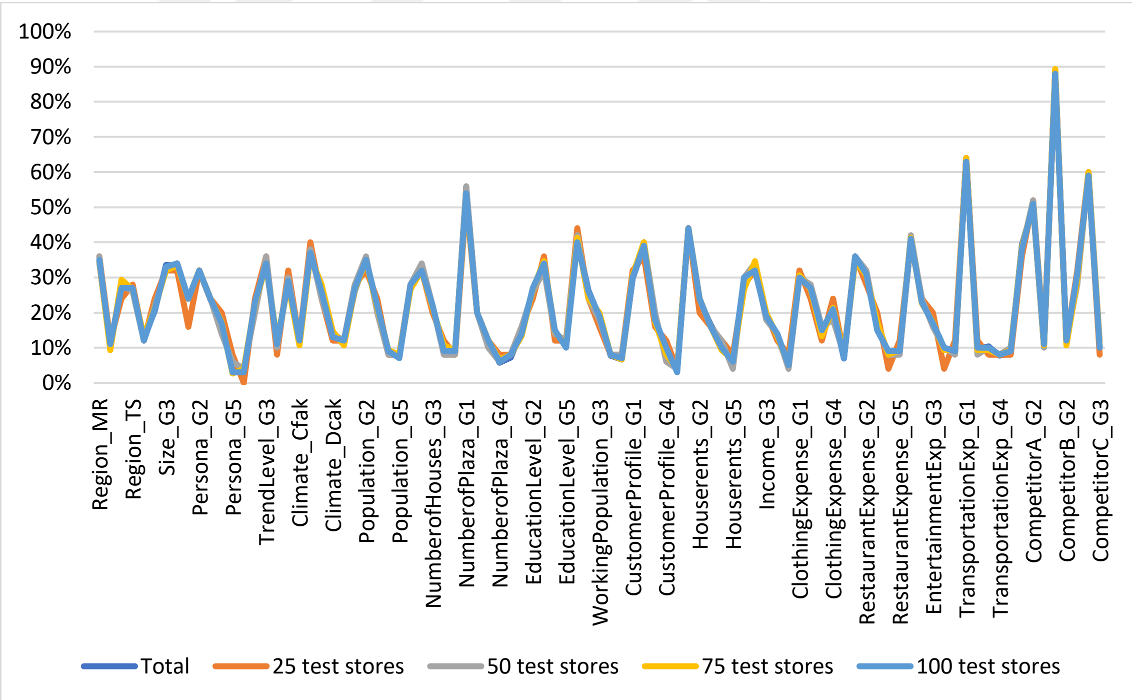

Figure 4. 3: Percentages of store feature values in different number of test stores and total number of stores

We used the "PuLP" module functional for linear programming in Python to implement the TSS model described above. PuLP is a library in Python that enables users to develop mathematical programs with and objective function, decision variables, and constraints.

The focus in the PuLP library is linear and mix-integer programming (Mitchell et al., 2011).

Linear Programming (LP) provides an optimal point in the n-dimensional feasible region that is linearly constrained. This point maximizes the value of the defined objective function. Integer Programming (IP) is a special form of LP where there are integer-valued variables in the solution. Furthermore, IP has a special form as Binary IP that has only 0 and 1 variables in the solution (Mitchell, 2009).

We used the “LpProblem” function that enables us to write a linear programming model. While describing the decision variables, we determine the type of variables as binary because we want to select test stores and our variables should be 0 or 1. If a store is selected, its decision variable should be 1. We would determine de type of variables as integer, and we would determine lower bound, 0 and upper bound 1 for all variables.

In Figure 4.4, there is the graph that shows the case in which test stores are selected randomly rather than using the method described in equations (4.1)-(4.4). For each case, 10 different random store selections are made. Then, average of the ratios of store features values are used. Although the ratios of store feature values seem different from the ratios in all stores for each random selection, the average ratios of ten selections become closer to the ratios in all stores. In any case, using the method described above provides the closer ratios to the total.

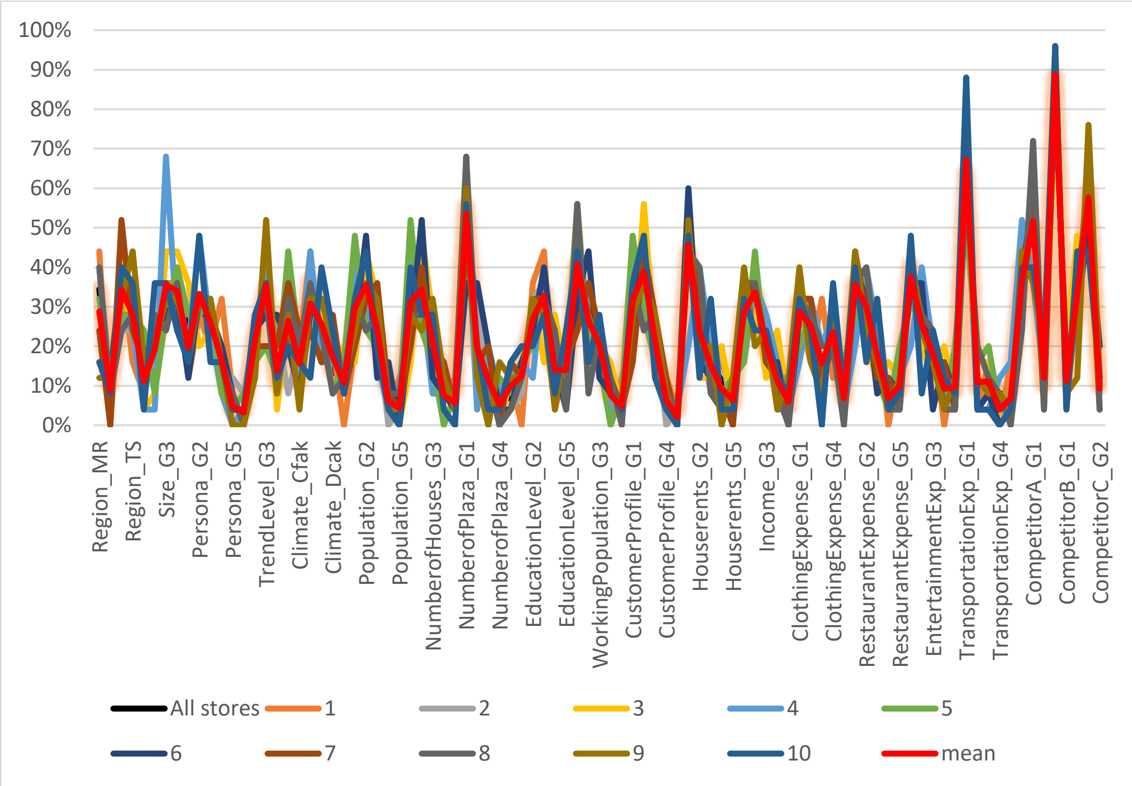

Figure 4. 4: Percentages of store feature values in different numbers of test stores and total number of stores when test stores are randomly selected

For each case, we select the test stores randomly 10 times. We calculated the average ratios of store feature values and compared them to ratios in all stores. We calculated MAPE for each random selection and also for the selection with binary programming. While calculating MAPE, we use the differences between ratios of store feature values of 364 stores and ratios of store feature values in selected number of test stores as error. The MAPE values for four different numbers of test stores using two different selection methods are shown in Table 4.9. Selection by binary programming results in significantly lower error rates than random selection for any number of test stores. For instance, the MAPEs for selection by binary programming is 14% for 25 test stores and 4% for 100 test stores, whereas these errors range from 31% to 41% and 20% to 27% for random selection. As the number of test stores is increased the error decreases more drastically with binary programming than with random selection, which is a strong support for using the proposed LP model for test store selection.

Table 4. 9: MAPE values for four different numbers of test stores by two methods of test store selection

	MAPE	
Number of Test Stores	**Binary Programming**	**Random Selection**
25	14%	31% - 41%
50	7%	21% - 28%
75	5%	20% - 29%
100	4%	20% - 27%

4.3.2 Analysis and sales forecasting process

After determining the test stores, test products are bought for those test stores. The number of orders should be calculated for just the test stores. A standard amount of test products are allocated to test stores. If the stock of test product in a test store is sold, then a new allocation would be made within three weeks of the test period. After the test period of three weeks, we analyze the sales performance in test stores and make forecasts for the other stores.

After the three-week sale period, we have sales data for test stores for a test product. We calculate the average sales for each cluster under the store features. Also, we create combinations of features. We obtain S_T^C and S_P^C matrices. We calculate the average sales amount for all the values in the S_T^C matrix. The average sales calculated for S_T^C matrix are used to create a new matrix of rate of sales as R for non-test stores. Then, using the column weights, w_q, the sales of non-test stores are forecasted as the weighted sum of these average sales quantities in the R matrix with the equation (4.8). We used entropy and information gain in calculations to determine the weights for store feature values. Entropy represents the amount of the information index of the degree of clutter or uncertainty (Li, 2014). Information gain value is equal to the amount of decrease in entropy. We calculated the information gain values for each column of S_T^C matrix and used those values as weights.

We applied this method for seven test products in four cases to forecast sales. In addition, we used five well-known data science algorithms: Linear Regression, Decision Tree, Gradient Boosting, Extreme Gradient Boosting, and Support Vector Machine. Figure 4.5 shows the results of these six algorithms. There are no significant differences in MAPE for different numbers of test stores. MAPE results of the new algorithm are lower than the other algorithms.

Figure 4. 5: Average MAPE values for different forecasting methods

The run times of the Linear Regression, Decision Tree, Gradient Boosting, Extreme Gradient Boosting, and SVM are close to each other, between 0.13 and 0.17 seconds. The run time of our new method ranges from 2.2 to 120 seconds, depending on the data, which is a considerably short time for sales forecasting tasks. Therefore, based on both the low forecasting error and short run time performance, the proposed method is a powerful solution.

4.4 Conclusions

It is necessary to test the products that have unpredictable potential and no historical sales data for both economic and environmental sustainability. Merchandise testing provides means of predicting the potential of products to companies with small budgets. In this study, test store selection is modeled, and sales forecast for non-test stores is made, both based on store features. The data for store features are obtained both from market research companies for external features and from the company for internal features. The complete data set consists of categorical and numerical data, and the numerical data are converted to categorical data by k-means clustering. After preparing the data, the integer programming model TSS is used to select the test stores such that they represent the distribution of store feature values of all stores. In the case study, four different numbers of test stores (25, 50, 75, 100) are used for seven products. The test store feature distributions in each case followed the overall distribution closely. For instance, MAPE score for 100 test stores by binary programming is 4%, whereas it is 24% on average by random selection.

After determining the test stores, test products are bought for those test stores. Test products are allocated to test stores with a standard allocation size. After the test period of 3 weeks, the proposed forecasting algorithm is used to forecast the sales for non-test stores. The forecasting results are compared with the results of Linear Regression, Decision Tree, Gradient Boosting, Extreme Gradient Boosting, and Support Vector Machine algorithms. The average MAPE for the proposed algorithm is 3-12% for the 28 problem instances, whereas the average MAPE is 6-35% for Linear Regression, 4-17% for Decision Tree, 4-14% for Gradient Boosting, 4-16% for Extreme Gradient Boosting, and 4-14% for the Support Vector Machine algorithms, which shows the relative improvement of forecasting accuracy by our algorithm.

Although the running time of the proposed forecasting model is longer than the other forecasting methods due to being very detailed, the running times within a few minutes are not considered long in the merchandise testing process in the retail industry. For instance, the average running time of the other forecasting methods is 0.15 seconds, while our new forecasting method takes 2 to 120 seconds. As a future extension of this work, to decrease the running time of the new forecasting method, it may be necessary to add certain parameters that users can select, and both the accuracy and speed of the method can be improved.

In addition, product features are not used in this study since the focus of this study is to obtain accurate results for the products with new trends over the store features. For possible future studies, forecasting the potential of new trends and integrating it into sales data can enable using product features, including recent trends.

Introducing new products into the market is a tremendous responsibility that requires accurate design, planning, placement, and timing for retailers. In an era where sustainability is of utmost importance, fashion retail companies and consumers would significantly benefit from utilizing novel methods that incorporate store-based features to make better allocations and forecasts in merchandise testing, as presented in this study.

5. SALES-BASED SIMILARITY NETWORK OF STORES

5.1 Current State in Apparel Retail Industry

In Merchandise Testing, clusters for different features are used to determine the test stores. At the same time, those clusters are used to determine real potential demand of products for the stores different from test stores. The features used in Merchandise Testing are related to socioeconomics, transportation, rival brands, and trend level. There is a need for a clustering study for sale performance for categories. We need to cluster stores based on the performance in categories. In a category, stores in which some similar products have good performance, some similar products have bad performance at the same time. For example, product A is the best product and product B is the worst product in Store X and Store Y. In this case, store X and Store Y have the similar production performance. Those two stores have similar performance in the related category. We need to have an algorithm to select the stores with similar sales performance into the same cluster. Therefore, we will be able to determine the clusters based on performance. Similarities between stores in terms of sales performance is the basis of clustering study. We use calculations similar to those in recommendation systems when establishing the sales-based similarity relationship between stores.

5.2 Objective, Outcomes, and Deliverables

With the prevalence of e-commerce due to advances in technology, searching for the most suitable products has become more challenging for consumers, and personalized recommendation systems have been developed to help them find relevant products faster. A recommendation system is a specific type of information filtering technique that attempts to present information items such as movies, music, websites, or news that are likely of interest to the user. Several companies in the e-commerce and IT industry, such as Netflix, Amazon, Spotify, and YouTube, have developed successful recommendation systems. Intuitively, a recommendation system builds up a user's profile based on their past records, compares it with reference characteristics, and seeks to predict the rating that a user would give to an item they had not yet evaluated (Yun, et al., 2018). In most cases, the recommendation system corresponds to a large-scale data-mining problem and can be observed in various settings where there are a set of users and a set of items that users interact with through viewing, rating, or purchasing. The most frequently used recommendation algorithm is collaborative filtering (CF), where content preferences for a target user are predicted based on the history of content preferences of similar users.

In this thesis, we use collaborative filtering (CF) to set a network between stores based on the performance of products sold in the stores. In this network, the stores are the users, and products sold at these stores are the items. While a similarity coefficient is used in CF, we use a distance metric as the dissimilarity coefficient. Using dissimilarities in sales performance of products in the different stores, we calculate the distance of performance between the stores. The greater the similarity in performance between stores, the smaller the distance.

By using the performance of common products in the stores, a distance metric is calculated between stores. That distance metric is like dissimilarity of the performance of the products sold in the different stores, rather than spatial distance. If a product has a high sales performance in one store and low performance in another, the dissimilarity, and thus the distance, between the two stores will be high. Conversely, if a product performs high or low in both stores, the distance between the two stores will be small.

That distance metric provides us to set a network whose nodes are stores. Then, we cluster the stores using the store network with the distances. We study a clustering method like k-means clustering. The main difference between our method and k-means clustering is that we use a distance metric determined from store to store, not spatial distance. In the k-means clustering algorithm, when choosing a cluster for a store, it enters the cluster with the closest cluster center to it. We do not calculate the centroid for each cluster. In our method, when choosing a cluster for a store, the selection is made according to the average of the distances between the stores in that cluster. As in the k-means method, the number of clusters, k, is given as a parameter in our method. The method works according to a given number of clusters, k.

5.3 Methodology

The goal of the Store Network method is to create a network of stores using sales data and to cluster the stores according to that network. The nodes of the network represent the stores and arcs are associated with the relation coefficient or distance between the stores. By using the performance of common products in the stores, a distance metric is calculated between stores. This distance metric is a measure of dissimilarity of the performance of the products sold in different stores, rather than spatial distance. If a product has a high sales performance in one store and low performance in another, the dissimilarity, and thus the distance, between the two stores will be high. Conversely, if a product performs high and low in two stores, the distance between those stores will be small.

In the Figure 5.1, the flowchart of the method is shown. The method starts with creating the store distance table and ends with clustering the stores. Store distance table contains the store-to-store distance values. It is also the source of the store network. After distance table between stores is obtained, we sort the stores by total distance to other stores in descending order. Our clustering method is similar to k-means clustering. The main difference between our method and k-means clustering is that we use a distance metric for store dissimilarity instead of a spatial distance. In the k-means clustering algorithm, when choosing a cluster for a store, it enters the cluster with the closest cluster center to it. We do not calculate the centroid for each cluster. In our method, when choosing a cluster for a store, the selection is made according to the average of the distances between the stores in that cluster. As in the k-means method, the number of clusters, k, is given as a parameter in our method.

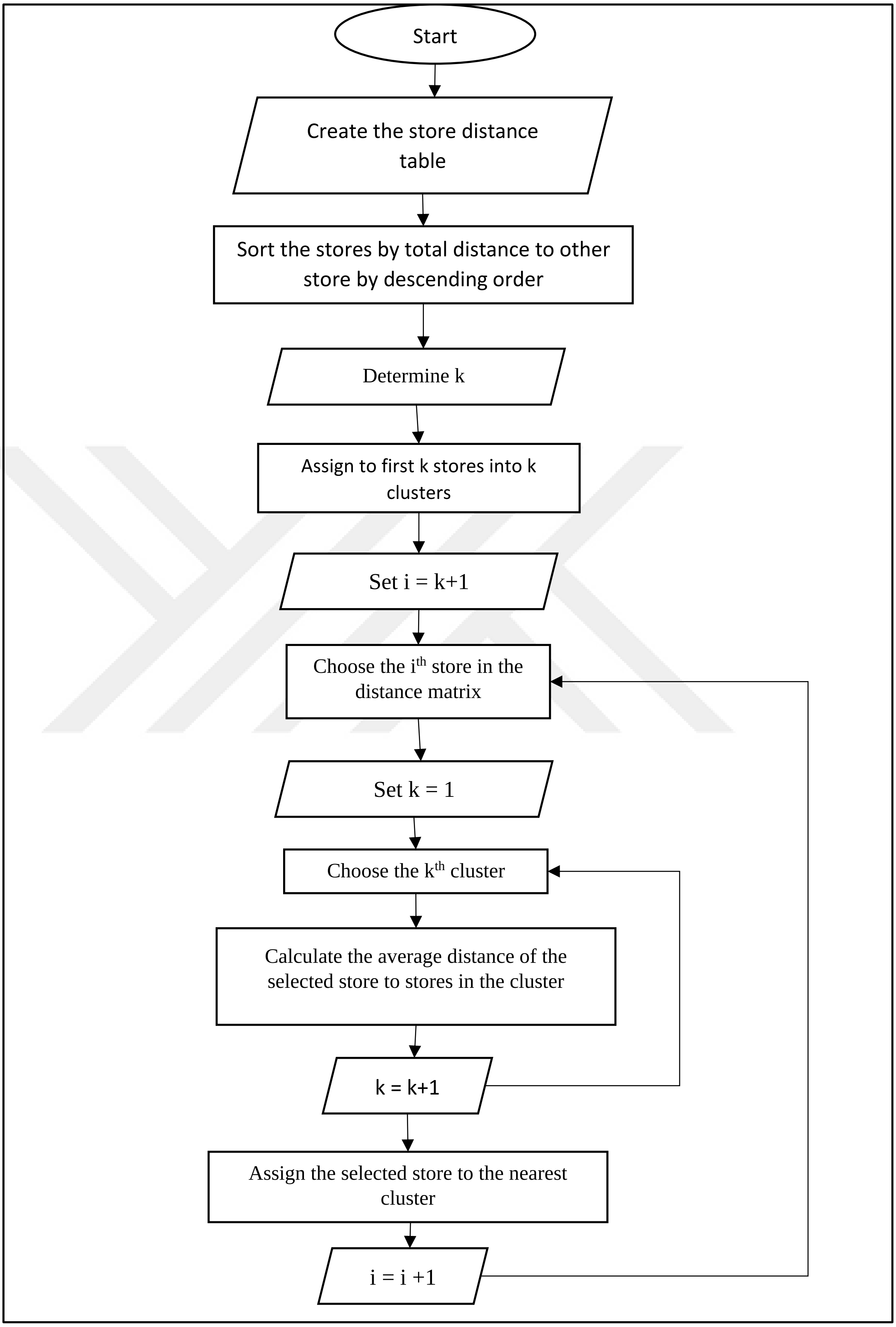

Figure 5. 1: Flowchart of the method

5.3.1 Creating store distance matrix

The method starts with the sales data of a category in a single period. For example, we use the sales data of t-shirt category in Men section in the summer period (June, July and August). An example of the sales data is shown in the Table 5.1. There are three columns in the sales data. The first column has stores, the second column has products sold in the relative store and the third column has sales quantities of related products in the related stores.

Table 5. 1: An example of sales data

Store	Product	Sales Quantity
Store 1	Product 1	6
Store 2	Product 3	2
Store 3	Product 5	8
Store 4	Product 5	4
Store 5	Product 1	6
Store 6	Product 2	7
Store 7	Product 4	8
…	…	…
…	…	…

We use the sales data and set the store-to-store distance matrix.

Notations:

$M = \{1,2,\dots,m\}$: the set of stores

$N = \{1,2,\dots,n\}$: the set of products

N_{ij} is the set of common products between store i and store j.

T: The $t \times 3$ matrix of store features, where t is the number of combinations of stores and products. Each store may have different capacities for displaying the products in the store. Because of that, not all products may have been sold in every store. Therefore,

$\max(m, n) \le t \le m * n.$

If each of n products are sold in each of m stores, t will be equal to mn.

To determine the distance between stores, we quantify the sales performance of products in the stores. For each store, we calculate the z-value of the products sold in that store, using the sales quantities.

x_{pi}: The sales quantity of product p in store i.

μ_i: The average sales quantity in store i.

σ_i: The standard deviation of sales quantities in store i.

z_{pi}: The z-value of product p in store i.

$$z_{pi} = \frac{x_{pi} - \mu_i}{\sigma_i} \tag{5.1}$$

The z-value shows how many standard deviations a product's sales are away from the average sales in that store. The positive z value indicates that the related product has above-average sales in the related store, and vice versa for the negative z value. For a common product in two stores, the absolute difference of z values is the distance between those stores in terms of that common product.

The distance between store i and store j, d_{ij}, is calculated as shown below.

$$d_{ij} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{p \in N_{ij}} |z_{pi} - z_{pj}|}{|N_{ij}|}, i \neq j \tag{5.2}$$

After calculating the distances between all of the stores, we obtain the distance matrix for stores as shown Table 5.2.

Table 5. 2: Distance matrix for stores

Store	Store 1	Store 2	Store 3	Store 4	Store 5	...	Store m
Store 1	0	d_{12}	d_{13}	d_{14}	d_{15}	...	d_{1m}
Store 2	d_{21}	0	d_{23}	d_{24}	d_{25}	...	d_{2m}
Store 3	d_{31}	d_{32}	0	d_{34}	d_{35}	...	d_{3m}
Store 4	d_{41}	d_{42}	d_{43}	0	d_{45}	...	d_{4m}
Store 5	d_{51}	d_{52}	d_{53}	d_{54}	0	...	d_{5m}
Store 6	d_{61}	d_{62}	d_{63}	d_{64}	d_{65}	...	d_{6m}
Store 7	d_{71}	d_{72}	d_{73}	d_{74}	d_{75}	...	d_{7m}
Store 8	d_{81}	d_{82}	d_{83}	d_{84}	d_{85}	...	d_{8m}
Store 9	d_{91}	d_{92}	d_{93}	d_{94}	d_{95}	...	d_{9m}

...	...	...	...	...	...	...	...
Store m	d_{m1}	d_{m2}	d_{m3}	d_{m4}	d_{m5}	...	0

In Figure 5.2, the sales of the products in the stores are displayed. There are different products in the different stores and we use the common products between two stores to calculate their distance.

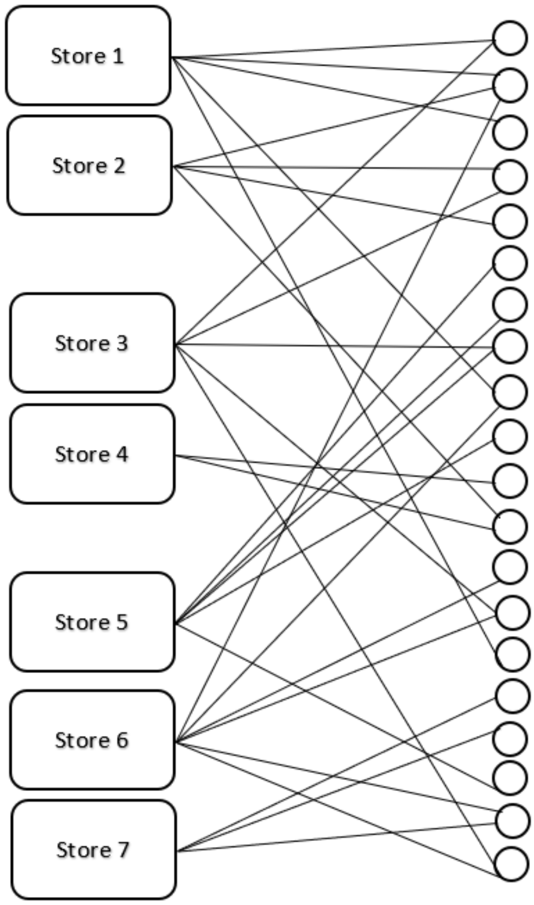

Figure 5. 2: Sales network

After calculating the distance between stores and setting the store distance table, we obtain such a network, an example of which is shown in Figure 5.3. The distance is less between the stores that are similar in terms of the sales performance of the products.

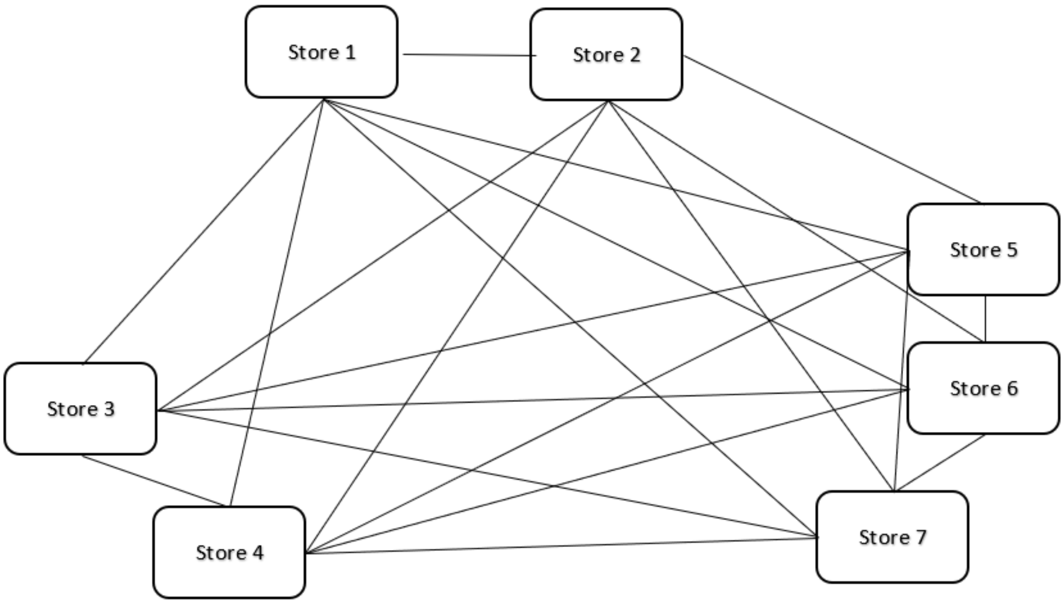

Figure 5. 3: An example of store network obtained from store distance table

5.3.2 Clustering the stores

After the creating of the distance matrix for stores, we cluster the stores by using the distance metric in the distance matrix. We use the steps in the K-means clustering. K-means algorithm is one of the basic unsupervised clustering algorithms developed in 1967 (MacQueen, 1967). It works for a predetermined number of clusters. The algorithm defines k centroids, one for each cluster. Those centroids should be as far as possible from each other in terms of Euclidean distances. Then, each point in the data is evaluated and assigned to the nearest cluster. After each point is placed in a cluster, the center of that cluster is recalculated. This step is repeated until all the points are considered. When the clusters of points no longer change, the algorithm is terminated.

In our study, we have the distance metric that is not spatial, but similar to the dissimilarity between stores. Our distance metric is defined store-to-store. Because of that, we cannot calculate the centroid of clusters. Instead of that, we calculate the average of distance between stores in cluster.

Firstly, we calculate the total distances of each store from other stores. D_i is the total distance of store i to the other stores.

$$D_i = \sum_{j=1}^{m} d_{ij}, \ \forall i = 1,2,\dots,m \ and \ i \neq j \tag{5.3}$$

We start the clustering with the number of clusters, k. Then, k stores with the highest D_i are assigned the k clusters separately. After first k stores are assigned to the clusters, $(k+1)^{th}$ store is assigned to the nearest cluster. While we are determining the nearest cluster to the next store, we do not calculate the distance to the centers of clusters, because our distance metric is not spatial. We calculated the average distance between stores in the cluster, if the next store enters that cluster. Thus, next store will enter the cluster with the minimum average distance calculated. This is a little different from the k-means clustering, but very similar in selecting the nearest cluster for the stores. After each store enters a cluster, average distance between stores is calculated again.

After all of the stores are assigned to a cluster, we go back to the first store and apply the steps above again. Algorithm works until clusters of stores no longer change.

5.3.3 Calculating the forecast error according to the clustering

After clusters are determined, we check the accuracy of the clustering. In fashion retail, not every product can be displayed in every store. We can make sales forecasting for the products for the stores where these products are not displayed. We consider the average z-value in stores where a product is sold in a cluster as the z-value in stores in the same cluster where it is not sold. In other words, our basic approach is that a product will have a similar z-value in different stores in the same cluster.

$C = \{1,2,\ldots,k\}$: the set of clusters

N_i is the set of products sold in store i.

z_{pi}: The z-value of product p in store i.

z_{pc}: The z-value of product p in cluster c.

F_{pi}: The sales forecast for product p in store i.

$$z_{pc} = \frac{\sum_{p \in N_i} z_{pi}}{|N_i|} \tag{5.4}$$

Then, we use the z_{pc} value and calculate the sales forecast for the product p in store i.

$$F_{pi} = \mu_i + z_{pc} \, x \, \sigma_i \tag{5.5}$$

We calculate the Weighted Mean Absolute Percentage Error (WMAPE) to measure the accuracy of the clustering.

M_c = the set of stores in the cluster c

$$WMAPE_c = \frac{\sum_{i \in M_c} |x_{pi} - F_{pi}|}{\sum_{i \in M_c} x_{pi}} \tag{5.6}$$

5.3.4 Application

In our data, there are 50 stores and 598 products. The sale of the product with the lowest sales in 50 stores is 2; the sale of the product with the highest sales in 50 stores is 2742. In the Table 5.3, there are summary of sales data. As shown in the table, the number of products sold in each store is different. Moreover, the range of the sales quantities is different from each other.

Table 5. 3: Summary of sales data

Stores	Number of Products	Min of Sales Quantity	Average Sales Quantity	Max of Sales Quantity
Store 1	271	9	114	241
Store 2	387	2	120	351
Store 3	108	2	94	189
Store 4	205	8	145	351
Store 5	119	16	102	178
Store 6	72	4	101	300
Store 7	26	8	19	28
Store 8	140	5	137	287
Store 9	350	4	281	736
Store 10	186	4	96	322
Store 11	171	11	81	219
Store 12	188	13	150	357
Store 13	253	12	135	407
Store 14	251	11	117	280
Store 15	126	14	103	178
Store 16	43	117	137	158
Store 17	70	18	131	209
Store 18	288	9	127	407
Store 19	280	7	112	243
Store 20	111	10	66	228
Store 21	221	3	116	243
Store 22	50	14	151	315
Store 23	100	4	103	178
Store 24	109	14	126	266

Store 25	324	13	174	372
Store 26	91	4	143	281
Store 27	445	4	254	736
Store 28	318	2	163	556
Store 29	208	8	111	324
Store 30	95	2	117	249
Store 31	368	2	139	556
Store 32	118	4	116	249
Store 33	348	8	161	372
Store 34	327	3	152	426
Store 35	222	13	135	220
Store 36	81	8	50	202
Store 37	246	9	110	276
Store 38	187	3	134	300
Store 39	148	16	115	248
Store 40	187	13	90	236
Store 41	240	2	96	322
Store 42	158	8	97	200
Store 43	308	2	96	247
Store 44	104	11	119	310
Store 45	239	14	123	266
Store 46	111	6	137	196
Store 47	292	9	140	289
Store 48	250	13	106	249
Store 49	142	15	111	249
Store 50	348	13	408	2742

5.3.4.1 Creating store distance matrix

We used the method described above to create store distance matrix. We obtained $50x50$ symmetric matrix with diagonal elements 0 as seen in Table 5.4.

Table 5. 4: Some part of the store distance matrix

Store	Store_1	Store_2	Store_3	Store_4	Store_5	Store_6	Store_7	Store_8	…	…	…	**Store_50**
Store_1	0.00	0.41	0.28	0.43	0.89	0.34	1.02	0.55	…	…	…	0.40
Store_2	0.41	0.00	0.15	0.33	1.13	0.57	0.82	0.72	…	…	…	0.41
Store_3	0.28	0.15	0.00	0.25	0.87	0.67	1.24	0.51	…	…	…	0.38
Store_4	0.43	0.33	0.25	0.00	1.01	0.53	1.43	0.50	…	…	…	0.59
Store_5	0.89	1.13	0.87	1.01	0.00	0.97	2.85	0.98	…	…	…	1.04

Store_6	0.34	0.57	0.67	0.53	0.97	0.00	1.79	0.62	…	…	…	0.43
Store_7	1.02	0.82	1.24	1.43	2.85	1.79	0.00	2.85	…	…	…	0.91
Store_8	0.55	0.72	0.51	0.50	0.98	0.62	2.85	0.00	…	…	…	0.52
…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…
Store_50	0.40	0.41	0.38	0.59	1.04	0.43	0.91	0.52	…	…	…	0.00

We calculate the total distances of each store from other stores. D_i is the total distance of store i to the other stores.

$$D_i = \sum_{j=1}^{m} d_{ij}, \ \forall i = 1,2,\dots,m \ and \ i \neq j \tag{5.7}$$

As seen in the Table 5.5, we listed the stores according to the sum of the distances from other stores, D_i.

Table 5. 5: Some part of the store distance matrix

Store	Total Distance
Store 7	80.26
Store 5	50.66
Store 50	50.07
Store 36	42.55
Store 16	55.52
Store 22	43.22
Store 44	43.21
Store 46	39.18
Store 31	35.31
Store 34	34.74
Store 28	33.16
Store 9	33.05
Store 27	33.00
Store 26	31.97
Store 25	32.65
Store 13	29.66
Store 20	32.33
Store 17	28.89
Store 33	28.78

Store 8	31.23
Store 41	28.37
Store 39	28.30
Store 18	28.29
Store 43	28.22
Store 11	28.19
Store 29	30.46
Store 10	27.51
Store 35	27.39
Store 6	32.90
Store 47	30.02
Store 24	26.77
Store 2	26.56
Store 4	26.42
Store 37	26.19
Store 45	25.99
Store 38	25.78
Store 15	25.16
Store 32	24.40
Store 48	24.31
Store 49	24.04
Store 40	23.85
Store 12	23.84
Store 1	23.81
Store 14	23.71
Store 30	26.42
Store 19	23.38
Store 42	23.26
Store 23	23.16
Store 3	22.93
Store 21	22.86

5.3.4.2 Clustering the stores

We use the clustering method similar to k-means clustering. We applied the method for different k values, from 1 to 10. We calculated the average distance between stores in each cluster. Moreover, we controlled the accuracy of the clustering by making forecast for the products and calculating the WMAPE scores for these forecasts. As seen in Table 3, each store has different number of products to be sold. Some of the products are not displayed in some of the stores. We made forecast for the products for each store by using the clustering results. We considered the average z-value in stores where a product is sold in a cluster as the z-value in stores in the same cluster where it is not sold. In other words, our basic approach is that a product will have a similar z-value in different stores in the same cluster. In Table 5.6, there are average distance and WMAPE values in terms of different number of clusters. As seen in the table, as the number of clusters increases, the distance and WMAPE values decrease.

Table 5. 6: Average distance and WMAPE values according to the different number of clusters

Number of Clusters	AVG DISTANCE	WMAPE
1	0.6416	26.2%
2	0.6010	24.9%
3	0.5596	23.0%
4	0.5305	22.6%
5	0.4606	21.5%
6	0.3795	19.8%
7	0.3950	18.3%
8	0.3808	17.5%
9	0.3584	16.0%
10	0.3570	15.5%

5.3.4.3 Controlling the WMAPE for each cluster

WMAPE values decrease while number of clusters increases. Although the WMAPE values decreases, it may be deviation in WMAPE scores between different clusters. We controlled the WMAPEs for different clusters in the case of number of clusters is equal to 4 and 5.

As seen Figure 5.4, after clustering was completed, WMAPE for all data is 22.6%. However, WMAPE for cluster 4 is larger than WMAPE for all data and it is 39.1%. WMAPE for cluster 1 is 15.5%. There are deviations for cluster 1 and cluster 4. In this case, we do some iterations to decrease the deviations in the WMAPE for clusters. Firstly, we select the store with highest total distance, D_i, in the cluster 4 and put this store to the cluster 1. Here, we aim to put the store with highest total distance from the cluster with highest WMAPE to the cluster with lowest WMAPE. After this change, we see the case 2. WMAPE for Cluster 4 approached to the WMAPE for total, but WMAPE for Cluster 1 increased. In this step, we select the store with highest total distance, D_i, in the cluster 1 and put this store to the cluster 3. After step 2, we encountered the Case 3. We select the store with highest total distance, D_i, in the cluster 3 and put this store to the cluster 1.

Finally, after three steps, we see the case 4. Deviation in WMAPE for clusters decreased and WMAPEs approached to the average.

Case 1		
Clusters when k=4	WMAPE	WMAPE Difference From WMAPE of Total
1	15.5%	7.1%
2	21.5%	1.2%
3	19.9%	2.7%
4	39.1%	16.5%
Total	**22.6%**	0.0%

Case 2		
Clusters when k=4	WMAPE	WMAPE Difference From WMAPE of Total
1	30.8%	8.4%
2	21.5%	1.0%
3	19.9%	2.5%
4	21.3%	1.2%
Total	**22.4%**	0.0%

Case 3		
Clusters when k=4	WMAPE	WMAPE Difference From WMAPE of Total
1	15.5%	7.0%
2	21.5%	1.1%
3	26.6%	4.1%
4	21.3%	1.2%
Total	**22.5%**	0.0%

Case 4		
Clusters when k=4	WMAPE	WMAPE Difference From WMAPE of Total
1	18.5%	4.1%
2	21.5%	1.1%
3	26.2%	3.6%
4	21.3%	1.3%
Total	**22.6%**	0.0%

Figure 5. 4: Situation of WMAPE for Clusters for the number of Clusters, k=4

We applied this method for the number of clusters 5. The results of these are in the Figure 5.5.

Case 1		
Clusters when k=4	WMAPE	WMAPE Difference From WMAPE of Total
1	4.7%	16.8%
2	18.2%	3.3%
3	8.2%	13.3%
4	26.5%	5.0%
5	23.0%	1.5%
Total	**21.5%**	0.0%

Case 2		
Clusters when k=4	WMAPE	WMAPE Difference From WMAPE of Total
1	11.9%	9.5%
2	18.2%	3.2%
3	8.2%	13.2%
4	25.1%	3.7%
5	22.9%	1.5%
Total	**21.4%**	0.0%

Case 3		
Clusters when k=4	WMAPE	WMAPE Difference From WMAPE of Total
1	11.9%	9.4%
2	18.2%	3.2%
3	18.0%	3.4%
4	24.7%	3.4%
5	22.9%	1.6%
Total	**21.4%**	0.0%

Case 4		
Clusters when k=4	WMAPE	WMAPE Difference From WMAPE of Total
1	18.7%	2.2%
2	18.2%	2.8%
3	18.0%	3.0%
4	22.6%	1.6%
5	22.9%	2.0%
Total	**20.9%**	0.0%

Figure 5. 5: Situation of WMAPE for Clusters for the number of Clusters, k=5

To sum up, we set a store networks in terms of the sales performance of the products sold in these stores. We calculated a distance metric over the performance of products, different from spatial distance. We created a store distance matrix and clustered the store using this matrix. The aim is to cluster the stores with the minimum average distance between stores in each cluster. We used a clustering method similar to k-means clustering with different k's from 1 to 10. After clusters completed, we controlled the accuracy of the clustering. We calculated WMAPE scores for all of the products. We have assumed that the z-value of the sales of a product in a store is close to the mean z-value of the sales of that product in the stores in the same cluster. After the forecast was made for the products, we calculated WMAPE scores for all of the products. While the number of clusters is increasing, WMAPE value decreases.

5.3.4.4 Discussion and conclusions

In some examples, there may be two stores, which do not have common products. We can use similar products to calculate the distance between those two stores in such a case.

Firstly, we need to create an item-based similarity matrix. By the adjusted cosine similarity coefficient formula below, we calculate the similarity between products. We consider the sales in Common stores where two products are sold.

$$sim(u,v) = \frac{\sum_{i \in C_{u,v}}(R_{u,i}-R_u)(R_{v,i}-R_v)}{\sqrt{\sum_{i \in C_{u,v}}(R_{u,i}-R_u)^2 \sum_{i \in C_{u,v}}(R_{v,i}-R_v)^2}} \tag{5.8}$$

Where $R_{u,i}$ is the rating of user u on the item i; $R_{v,i}$ is the rating of user v on the item i; $C_{u,v}$ is the set of common items rated by the user u and v. R_u is the average rating given by user u, and R_v is the average rating given by user v.

For instance, we created a similarity matrix using ten products' sales data. In the case that there is not any common product between two stores, we can use top N similar products' z value and calculate the distance between those two stores.

Table 5.7 contains the example of an item-based similarity matrix.

Table 5. 7: Example of Item-Based Similarity Matrix

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1	1.00	0.68	0.87	0.37	0.72	0.22	0.18	0.56	0.81	0.86
P2	0.68	1.00	0.68	0.66	0.28	0.80	0.73	0.89	0.43	0.72
P3	0.87	0.68	1.00	0.38	0.74	0.31	0.26	0.65	0.83	0.88
P4	0.37	0.66	0.38	1.00	-0.03	0.71	0.74	0.67	0.14	0.29
P5	0.72	0.28	0.74	-0.03	1.00	-0.12	-0.14	0.21	0.87	0.64
P6	0.22	0.80	0.31	0.71	-0.12	1.00	0.97	0.76	0.08	0.36
P7	0.18	0.73	0.26	0.74	-0.14	0.97	1.00	0.66	0.08	0.26
P8	0.56	0.89	0.65	0.67	0.21	0.76	0.66	1.00	0.36	0.66
P9	0.81	0.43	0.83	0.14	0.87	0.08	0.08	0.36	1.00	0.74
P10	0.86	0.72	0.88	0.29	0.64	0.36	0.26	0.66	0.74	1.00

6. DISTRIBUTION OF COLLECTION INTO STORES USING OPTIMIZATION

6.1 Current State in Apparel Retail Industry

In fashion retail companies, collections are prepared for seasons, phases, months, or weeks. While preparing a collection, products are selected according to the features of the period and the product attributes by category managers. Fundamental product attributes are fabric, color, pattern, fit, silhouette, neck, and trend level. It is necessary to consider the proportions of those product attributes in the collection. According to the features of the period, company should put up the products having each color, fabric, pattern, and any other attributes for sale in the right ratios. The goal is to prepare and distribute a collection such that each customer who arrives at a store is able to find the products they need. To prevent loss of sales and to provide customer satisfaction, it is needed to create the integrity of color, fabric, pattern, fit or any other attribute related to the category.

Each store may have different capacities of presenting products. The term capacity here is the maximum number of items that can be displayed in the store at anytime. Even if two stores are the same size, the space allocated to the categories may vary. For example, capacity of swimwear category may differ between a store in a continental climate and a store in a seaside area. Since the capacities of categories in stores may vary, it is difficult to manage the collections in the all of the stores. Category managers prepare a main collection for global and they version it for different sales region. In the case that stores are categorized by the capacities of the categories and there are few types of stores with different capacity, it would be easier to manage the collection in store detail. In reality, there are a lot of different number of capacities of categories and category managers cannot find enough time to plan collection for each store types. There is a need for a smart algorithm that will enable category managers to prepare a large collection and distribute this collection to stores with various product capacities.

6.2 Objective, Outcomes, and Deliverables

In our study, we try to distribute the main collection to each store in terms of their capacities. While doing this, we aim to provide the distribution of product attributes in main collection for each store. For example, if 20% of the products in the main collection are white color, we aim to make nearly 20% of the collections white color in each store. In addition, we consider this rule for several product attributes.

Moreover, while our method is selecting the products for stores, it tries to select the products with the biggest potential in the related store.

The large collection prepared for the largest store will be distributed by integer programming algorithm to smaller stores. The algorithm will use product attribute rates taken from the large collection as constraints for integer programming.

All products in the large collection may not be selected for smaller stores, because of the capacity constraint. Thus, the algorithm should eliminate some of products. This elimination will be according to product attribute constraint. To eliminate the products, we can use products' rating scores given by category managers or sales forecasting. In integer programming, the objective function will be to maximize the total product rating of the collection. Therefore, if the algorithm should select 4 of 6 blue products, those 4 will be with maximum product rating. Product ratings can be global or change by store.

6.3 Methodology

Distribution algorithm will be a binary integer-programming algorithm. The objective function will be to maximize total rating score of the collection. Binary integer programming we use for distribution of collection is similar to Test Store Selection part in the Merchandise Testing. While we use store features and their values in Merchandise Testing topic. Here, we use product attributes. There will be product attribute constraints and capacity constraints. For example, if there are 5 attributes and 4 values under those, we will have 20 constraints and an additional capacity constraint.

Notations:

$M = \{1,2,\dots,m\}$: the set of products

$N = \{1,2,\dots,n\}$: the set of product attributes

S: The $m \times n$ matrix of product attributes, where m is the number of products, n is the number of attributes.

x_i: 1, if product i selected for the store; 0 otherwise.

r_i: rating score of product i.

$a_j = S(:,j)$ is the $m \times 1$ column vector for product attribute j for all products, $j = 1,2,\dots,n$.

U_j: The set of unique values in column vector a_j.

$d_j = |U_j|$: The number of unique elements in column j of the S matrix such that $d_j \leq m$.

$D = \sum_{j=1}^{n} d_j$: The total number of unique values of all product attributes.

SE: an $m \times D$ matrix of zeros and ones obtained by converting the columns in matrix S by defining a separate column for each different value of each product attribute.

$U_j(k_j)$: the column of SE for attribute $j = 1,2,\dots,n$ and attribute value $k_j = 1,2,\dots,d_j$.

SE_{ik}: the binary value corresponding to product i attribute value k.

C: the capacity of store for the category, the maximum number of products that can be selected.

Based on the notation above, note that each row sum for the SE matrix is equal to n:

$$\sum_{k=1}^{D} SE_{ik} = n, \forall i \in M \tag{6.1}$$

Since, for a given feature, a product can be only in one of the attribute clusters, it can only take one of the unique attribute values, i.e.

$$\sum_{k=\underline{k}}^{\underline{k}+d_j} SE_{ik} = 1, \forall i \in M, j \in N \tag{6.2}$$

where $\underline{k} = \sum_{a=1}^{j-1} d_a + 1$.

In Table 6.1, an example of the binary form of the product attribute matrix SE is presented. For example, Product 1 takes on the first attribute's first value; therefore, there is an entry of 1 under the first column related to the first feature and entries of 0 under the other columns related to the first feature.

Table 6. 1: Example of binary form of product attributes matrix SE

	Binary values related to a_1				Binary values related to a_2					Binary values related to a_n		
Unique feature value	$U_1($	$U_1(2)$		$U_1(d_1)$	$U_2(1)$	$U_2(2)$		$U_2(d_2)$		$U_n(1)$		$U_n(d_n)$
Column index in matrix SE	1	2	…	d_1	$d_1 + 1$	$d_1 + 2$	…	$d_1 + d_2$	…	$\sum_{j=1}^{n-1} d_j + 1$	…	D
Product 1	1	0	…	0	0	1	…	0	…	0	…	1
Product 2	0	0	…	0	0	0	…	0	…	0	…	0
Product 3	0	1	…	0	0	0	…	0	…	0	…	1
Product 4	1	0	…	0	1	0	…	0	…	1	…	0
…												
Product m	0	0	…	1	0	0	…	1	…	0	…	1

Objective function:

$$\text{Maximize} \qquad z = \sum_{i=1}^{m} r_i x_i \tag{6.3}$$

Subject to

$$\sum_{i=1}^{m} (SE_{ik} * x_i) \leq C * \left(\frac{\sum_{i=1}^{m} SE_{ik}}{m}\right) \quad \forall k = 1,2, \dots, D \tag{6.4}$$

$$\sum_{i=1}^{m} x_i \leq C \tag{6.5}$$

$$x_i \in \{0,1\}, \forall\, i \in M \tag{6.6}$$

The objective function of this binary program is to maximize the total rating score of stores by selecting the products with highest rating. Constraint (1) ensures that the number of products that have a certain value k for an attribute does not exceed the existing number of such products in main collection, where the ratio of products with a certain attribute value is represented by $\left(\frac{\sum_{i=1}^{m} SE_{ik}}{m}\right)$. Constraint (2) limits the number of products by the capacity of the store for that category. Constraint (3) defines the binary decision variables of the model.

We use PuLP library in the Python program for the Distribution Algorithm.

Table 6.2 shows sample tables of product attributes planned in the large collection.

Table 6. 2: Example of product attributes table

Product	**Color**	**Fabric**	**Neck**	**Graphic**
Product1	White	Supreme	Crew neck	Plain
Product2	Blue	Pique	Polo	Striped
Product3	Grey	Supreme	V-neck	Plain
Product4	Pink	Supreme	Crew neck	Printed

In Figure 6.1, there is an example of a t-shirt collection with product attributes. Collection contains 18 products with different colors, necks, fabrics and patterns. Before we distribute this collection to the stores with different capacities, we calculate the ratios of attribute values. In this example, 4 of t-shirts are white and this is 22% of the collection. Number of blue t-shirts are 3 (17% of collection). Moreover, there are 14 t-shirts with supreme fabric and the ratio is 78%. When we adapt the collection to a store with capacity of 14, we calculate the quantity of products according to ratios of attribute values in main collection. In our example, number of white t-shirts that should be in the collection of store with the capacity of 14 will be $14 * 22\% = 3.08 \cong 3$. This calculation should be done for the other attribute values. To achieve those calculations for other attribute values in the collection with 14 t-shirts at the same time, we use integer programming in Python.

Color: White Neck: Crew Fabric: Supreme Pattern: Plain	Color: Blue Neck: Polo Fabric: Pique Pattern: Striped	Color: Grey Neck: V-neck Fabric: Supreme Pattern: Plain	Color: Pink Neck: Crew Fabric: Supreme Pattern: Printed	Color: Yellow Neck: Polo Fabric: Pique Pattern: Striped	Color: Blue Neck: V-neck Fabric: Supreme Pattern: Plain
Color: White Neck: Crew Fabric: Supreme Pattern: Printed	Color: Green Neck: V-neck Fabric: Supreme Pattern: Striped	Color: Red Neck: V-neck Fabric: Supreme Pattern: Printed	Color: Grey Neck: Crew Fabric: Supreme Pattern: Plain	Color: White Neck: V-neck Fabric: Supreme Pattern: Striped	Color: Black Neck: Polo Fabric: Pique Pattern: Plain
Color: Green Neck: Crew Fabric: Supreme Pattern: Plain	Color: Black Neck: V-neck Fabric: Supreme Pattern: Plain	Color: Yellow Neck: Polo Fabric: Pique Pattern: Printed	Color: Blue Neck: Crew Fabric: Supreme Pattern: Plain	Color: Red Neck: Polo Fabric: Supreme Pattern: Plain	Color: White Neck: Crew Fabric: Supreme Pattern: Printed

Figure 6. 1: Example of a collection with product attributes in T-shirt category

We have a data of a collection of 100 products. Color, neck, fabric and pattern are the product attributes. The attribute values of the products in the collection as follow:

Color: Black, Ecru, Green, Grey, Navy, Red and White

Fabric: Melange, Nope and Single Jersey

Neck: Crew neck, polo neck and V-neck

Pattern: Animal, City, Death Head, Marine, Typography, Unprinted

We determine the ratios of attribute values in 100 product collection and form the collection for 50 and 85 capacity stores. Results are shown in Table 6.3. As seen, the ratios of attribute values in different stores are very close to each other.

Table 6. 3: Ratios of product attribute values in different capacity stores

	100 capacity store		**85 capacity store**		**50 capacity store**	
	Ratio	**Qauntity**	**Ratio**	**Qauntity**	**Ratio**	**Qauntity**
Color						
black	16%	16	16%	14	18%	9
ecru	21%	21	21%	18	22%	11

green	7%	7	7%	6	8%	4
grey	20%	20	20%	17	18%	9
navy	13%	13	12%	10	12%	6
red	10%	10	9%	8	10%	5
white	13%	13	14%	12	12%	6
Fabric						
melange	28%	28	28%	24	24%	12
nope	54%	54	53%	45	56%	28
single jersey	18%	18	19%	16	20%	10
Neck						
crew neck	38%	38	38%	32	40%	20
polo neck	37%	37	38%	32	38%	19
v neck	25%	25	25%	21	22%	11
Pattern						
animal	8%	8	8%	7	8%	4
city	5%	5	5%	4	2%	1
death head	10%	10	9%	8	10%	5
marine	14%	14	14%	12	14%	7
typography	37%	37	38%	32	38%	19
unprinted	26%	26	26%	22	28%	14

6.4 Contribution and Impact

‘Distribution of collection to stores’ provides managing all of the stores while planning collections. Normally, it is so difficult to prepare collection for each store. By this method, category managers may prepare just one large collection and manage the other stores. Large collection may be for the store with the largest capacity or for a virtual store with the capacity larger than the existing maximum capacity.

This method calculates the rate of the product attributes and distributes the large collection to stores with those rates. Moreover, category managers may determine the rates of attributes for stores as parameter. In this case, distribution algorithm uses those rates rather than calculating in the collection itself.

7. PREDICTION OF PRODUCTS' DEMAND BASED ON STORES

7.1 Current State

In Merchandise Testing topic, we aim to determine the potential demand of a product, which is new for the customer, company, or category.

In 'Distribution of collection to store' topic, we aim to distribute the large collection into stores with smaller capacity. We use integer programming and there is r_i, related to the rating or potential of products in the objective function. Category managers may give rating of products, or it may be calculated by a data science method. The determination of ratings by the category managers creates subjectivity.

Moreover, it is difficult to determine the ratings of products in terms of different stores for category managers. They may give a single rating to a product for all stores. To increase the usefulness of 'Distribution Method', we need to different rating or forecast of products for different stores. This provides us to select products for the stores with the most potential.

To predict the sales of products for different stores, we use last season's sales data. Since products in the collection may change over the seasons, this season's collection may not include the products from the last year data. Changing the products is the nature of fashion retail. It is easy to forecast the sales of basic models and models from last year. We use product attributes to forecast the potential of new products not included in last year's data. Moreover, we use the store features used in the Merchandise Testing topic to forecast by store. Therefore, we will have different r_i values that differ on a store basis.

7.2 Objective, Outcomes, and Deliverables

We aim to develop a prediction algorithm that works for any product in the research list of category managers. When category managers come accros a product in social media or in the e-commerce site of a brand, they should be able to know the potential of the product for each store.

We have the sales data for the previous seasons. Data is based on products and stores. We will use the product attribute and store features to analyze the past sales data and make predictions for the new season. Therefore, we will use the prediction results both for preparing a collection and for the Distribution Method.

We have sales data by store and product with columns for store features and product attributes. The sales column consists of sales quantities of a product in a store. Our prediction algorithm will predict the sales quantity for products and the stores. The result will be sales quantity.

We will use the result of the prediction algorithm in the Distribution Method to distribute the large collection into the stores with smaller capacity. In the Distribution Method, we will use the prediction score in the objection function. The objective function consists of the sum of the multiplication of the decision variables with the corresponding product rating score.

We will use the production rating scores for the store that the Distribution Method works for. That means, we will use customized rating/forecast in store detail. Therefore, those rating values will not be subjective and only one rating per product. They will be calculated by an algorithm and be store-based.

The prediction algorithm will work for products and stores seasonally. Thus, there will be one rating score for a product and a store for a season, and no weekly or monthly predictions. We will work only in the detail of product and store, not the detail of time.

7.3 Methodology

Sales data includes the features of product and store. In each row, there is a sales data for one store and one product. We have store features, product attribute, and sales quantities as demonstrated in Table 6.4.

Table 6. 4: Data consisting of Store Features, Product Attributes and Sales

Store	Product	Store Feature1	Store Feature2	Store Feature3	Product Attribute1	Product Attribute2	Product Attribute3	Sales
S1	P1	F1_1	F2_1	F3_1	A1_1	A2_1	A3_1	3
S1	P2	F1_1	F2_1	F3_1	A1_2	A2_2	A3_2	4
S1	P3	F1_1	F2_1	F3_1	A1_3	A2_3	A3_3	1
S2	P1	F1_2	F2_2	F3_2	A1_1	A2_1	A3_1	2
S2	P2	F1_2	F2_2	F3_2	A1_2	A2_2	A3_2	3

S2	P3	F1_2	F2_2	F3_2	A1_3	A2_3	A3_3	4

There is sales performance of 19 products in 346 stores with 6,574 rows. Products have 6 attributes. Trend level, pocket, price level, pattern, fabric and collar are the product attributes. In Table 6.5, there are the values of attributes in data. Store features are the same as Table 3 used in Merchandise testing.

Table 6. 5: Values of attributes in data

Trend Level	Pocket	Price level	Pattern	Fabric	Collar
Basic	with pocket	Lower	Printed	Dobby weave	Button
Commercial	pocketless	Middle	Striped	Poplin	Buttonless
Trendy		Upper	Unprinted		Mandarin
					Italian

We aim to develop a prediction algorithm that analyzes the sales quantities for each store feature and product attribute and make a prediction. For example, CM sees a yellow t-shirt in social media and want to include it in the collection for the next season. By the prediction model, they need to know in which stores the yellow t-shirt can be successful. The prediction model will know the performance of yellow t-shirts in the previous season's sales data. Also, the stores in which yellow t-shirts become successful or unsuccessful will be determined by the algorithm. We will focus on the features of the stores in which yellow t-shirts have data and make predictions for other stores using those stores' features.

In addition to the color data of the products, we also have other product attributes and we can analyze the features of the successful yellow t-shirts. Which fabric, neck, pattern, or any other features do the successful yellow t-shirts have? In addition, which features make the yellow t-shirt successful in which store? Which product attribute or store feature has more impact on the sales quantity?

We need to answer those questions and find a model to predict the store-based potential of products. Our data consists of 26 independent categorical variables (6 product attributes, 20 store features) and sales quantity as the dependent variable. Therefore, we need to work on regression issues. Decision Tree models are suitable for categorical data. In the prediction model, there are Regression Tree, Gradient Boosting and Extreme

Gradient Boosting. Gradient Boosting algorithms are advanced versions of Decision Tee method. The prediction model splits the data into test and train. It fits the train part and make prediction for test using 3 of the methods. According to the MAPE values, it selects the best method and use the predictions of it. The prediction model is created in Python and the library and parameters in Table 10 are used.

7.4 Application

Our data consists of 6,574 rows and 27 columns. 26 columns belong to independent categorical variables. There is a dependent sales quantity column. The prediction model splits the data as test and train. 75% of the data becomes train, and 25% of it becomes test. For each data science method, the model fit the train and make predictions for test part. Then, errors for predictions, MAPE, MAD and MSE values are calculated. The results of each method are shown in Table 6.6.

Table 6. 6: MAPE, MAD and MSE results for each data science method

Method	MAPE	MAD	MSE
Regression Tree	13.75%	0.1306	0.0373
Gradient Boosting	9.35%	0.0874	0.0154
Extreme Gradient Boosting	9.06%	0.0854	0.0154

Some parameters in Gradient Boosting and Extreme Gradient Boosting algorithms have impact on the predictions. 'n_parameters', 'max_depth' and 'learning rate' are the parameters that we try to find the best estimation using the different options of them.

We create the sets of different values for those parameters. Our model runs for each different parameter and save the results (predictions, errors, MAPE, MAD, MSE values).

- n_estimators: The number of trees in the ensemble, often increased until no further improvements are seen. In the model, we use 100 and 300.
- max_depth: The maximum depth of each tree, often values are between 1 and 10. We use 1, 5 and 10 in the model.
- learning_rate: The learning rate used to weight each model, often set to small values such as 0.3, 0.1, 0.01, or smaller. We use 0.1, 0.05, 0.03, 0.01 in the model.

We have 2 different values for number of estimators, 3 for max_depth and 4 for learning rate. The prediction model runs 2*3*4=24 times. It is possible to increase the number of parameters. Results for Gradient Boosting algorithm are shown in Table 6.7. The best result for MAPE comes from the case that n_estimator is 100, max_depth is 5 and learning_rate is 0.05. The best result for the MAD and MSE belongs to the case that n_estimator is 100, max_depth is 5 and learning_rate is 0.03. When the model runs for any other data, the best result comes from the different parameter values. The prediction model provides the best prediction using MAPE, MAD and MSE.

Table 6. 7: Results for gradient boosting with different parameters

n_estimator	max_depth	learning_rate	MAPE	MAD	MSE
100	1	0.1	12.75%	0.1181	0.0202
100	1	0.05	14.77%	0.1357	0.0248
100	1	0.03	16.80%	0.1524	0.0337
100	1	0.01	21.24%	0.1906	0.0611
100	5	0.1	9.66%	0.0916	0.0181
100	5	0.05	9.29%	0.0880	0.0167
100	5	0.03	9.35%	0.0874	0.0154
100	5	0.01	13.74%	0.1224	0.0263
100	10	0.1	13.16%	0.1255	0.0350
100	10	0.05	11.86%	0.1138	0.0304
100	10	0.03	11.28%	0.1076	0.0261
100	10	0.01	14.27%	0.1286	0.0300
300	1	0.1	11.88%	0.1096	0.0192
300	1	0.05	12.35%	0.1145	0.0197
300	1	0.03	13.06%	0.1209	0.0206
300	1	0.01	16.83%	0.1528	0.0339
300	5	0.1	11.13%	0.1048	0.0232
300	5	0.05	10.05%	0.0950	0.0195
300	5	0.03	9.55%	0.0907	0.0179
300	5	0.01	9.37%	0.0875	0.0154
300	10	0.1	13.77%	0.1311	0.0371
300	10	0.05	13.63%	0.1297	0.0365
300	10	0.03	13.04%	0.1243	0.0345
300	10	0.01	11.32%	0.1080	0.0261

Results for Extreme Gradient Boosting algorithm are shown in Table 6.8. The best result for MAPE and MAD comes from the case that n_estimator is 100, max_depth is 5 and learning_rate is 0.05. The best result for the MSE belongs to the case that n_estimator is 100, max_depth is 5 and learning_rate is 0.03.

Table 6. 8: Results for gradient boosting with different parameters

n_estimator	max_depth	learning_rate	MAPE	MAD	MSE
100	1	0.1	12.75%	0.1181	0.0202
100	1	0.05	14.73%	0.1359	0.0248
100	1	0.03	16.35%	0.1533	0.0344
100	1	0.01	22.90%	0.2544	0.1016
100	5	0.1	9.48%	0.0892	0.0171
100	5	0.05	9.06%	0.0854	0.0154
100	5	0.03	9.12%	0.0882	0.0148
100	5	0.01	19.46%	0.2163	0.0651
100	10	0.1	12.72%	0.1211	0.0326
100	10	0.05	11.14%	0.1059	0.0256
100	10	0.03	10.43%	0.1005	0.0204
100	10	0.01	19.76%	0.2178	0.0652
300	1	0.1	11.88%	0.1096	0.0192
300	1	0.05	12.36%	0.1145	0.0197
300	1	0.03	13.07%	0.1210	0.0206
300	1	0.01	16.36%	0.1537	0.0347
300	5	0.1	10.92%	0.1026	0.0223
300	5	0.05	9.75%	0.0917	0.0179
300	5	0.03	9.34%	0.0880	0.0166
300	5	0.01	9.13%	0.0885	0.0149
300	10	0.1	13.58%	0.1290	0.0359
300	10	0.05	13.39%	0.1272	0.0351
300	10	0.03	12.37%	0.1178	0.0313
300	10	0.01	10.47%	0.1009	0.0204

7.5 Contribution and Impact

We will have predicted store-based rating score for products. The rating scores taken from the algorithm will be numerical and continuous. We will use that data in the Distribution Algorithm to distribute the large collection to the stores with smaller capacity. Distribution algorithm will be able to select the best products which are suitable for attribute constraints and have the maximum possible sales potential. Moreover, category managers will consider the predicted sales data when they choose a product in the collection. Thereby, analytical results will be used in the collection preparation and constituting the Gantt plan for each store.

8. CONTRIBUTION AND IMPACT OF THESIS

Determining the place of trends in Boston Matrix provides category managers to focus on the Star and Cash Cow trends. Thus, they can consider certain upward trends. Then, they become able to test trends with the Merchandise Testing method. This method works for products with new trend to predict the high potential stores. Then, category managers will know the stores in which a trend has high potential. This information enables category managers to include the products with new trend into a collection. In addition, the Prediction Algorithm gives sales forecast of products used in collection preparation. Category managers can benefit from this method in preparing a collection by focusing on the products with high sales forecasts. There is a key point in the Prediction Algorithm in that it uses the last seasons' sales data, and it does not use trend analysis. Here, category managers should link both studies.

After preparing a collection for a region or for global sales, there is a need to distribute this collection into each store. We need to provide the attribute rates of the large collection in all stores. Distribution Algorithm provides category managers the attribute rates rather than determining these rates manually.

9. LIMITATIONS AND FUTURE DIRECTIONS

In this study, we studied predicting the potential of fashion trends and using them in collection management. Some further studies that can be done.

Boston matrix analysis can be done by analyzing data containing the frequency of use of trends in brands and product categories in the market. Market growth and relative market share are calculated and part of the fashion trends in the BCG matrix can be determined.

The forecasting part of the Merchandise Testing topic is very detailed, and the running time is average between 2 and 120 seconds. Some optimization steps for reducing the running time can be studied.

BIBLIOGRAHY

Aggarwal, C. C., Wolf, J. L., Wu, K.-L., & Yu, P. S. (1999). Horting hatches an egg: A New Graph-Theoretich Approach to Collaborative Filtering. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '99.*

AL-Bakri, N. F., & Hashim, S. H. (2019). Collaborative Filtering Recommendation Model Based on k-means Clustering. *Al-Nahrain Journal of Science, 22*(1), pp. 74–79. https://doi.org/10.22401/ANJS.22.1.10

Alon, I., Qi, M., & Sadowski, R. J. (2001). Forecasting aggregate retail sales: A comparison of artificial neural networks and traditional methods. *Journal of Retailing and Consumer Services, 8*(3), pp. 147-156.

Arora, G., & Kumar, A. (2014). Movie Recommendation System Based on Users'Similarity. *International Journal of Computer Science and Mobile Computing, 3*(4), pp. 765-770.

B.Thorat, P., M. Goudar, R., & Barve, S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications*, *110*(4), pp. 31-36. https://doi.org/10.5120/19308-0760

Baardman, L., Levin, I., Perakis, G., & Singhvi, D. (2018). Leveraging Comparables for New Product Sales Forecasting. *Production and Operations Management*, *27*(12). https://doi.org/10.1111/poms.12963

Benati, S., & García, S. (2014). A mixed integer linear model for clustering with variable selection. *Computers and Operations Research*, *43,* pp. 280-285. https://doi.org/10.1016/j.cor.2013.10.005

Brusco, M. J., Steinley, D., Cradit, J. D., & Singh, R. (2012). Emergent clustering methods for empirical OM research. *Journal of Operations Management*, *30*(6), pp. 454-466. https://doi.org/10.1016/j.jom.2012.06.001

Bölük, E. (2016). Köppen iklim sınıflandırmasına göre Türkiye iklimi. *TC Orman ve Su İşleri Bakanlığı Meteoroloji Genel Müdürlüğü, Araştırma Dairesi Başkanlığı, Klimatoloji Şube Müdürlüğü, Ankara*.

Bradlow, E. T., Gangwar, M., Kopalle, P., & Voleti, S. (2017). The Role of Big Data and Predictive Analytics in Retailing. *Journal of Retailing, 93*(1), pp. 79-95. https://doi.org/10.1016/j.jretai.2016.12.004

Cantador, I., & Castells, P. (2006). Multilayered semantic social network modeling by ontology-based user profiles clustering: Application to collaborative filtering. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4248 LNAI*. https://doi.org/10.1007/11891451_30

Chakraborty, S., Hoque, S. M. A., & Kabir, S. M. F. (2020). Predicting fashion trend using runway images: application of logistic regression in trend forecasting. *International Journal of Fashion Design, Technology and Education, 13*(3), pp. 376-386. https://doi.org/10.1080/17543266.2020.1829096

Chen, L., Mersereau, A. J., & Wang, Z. (2017). Optimal merchandise testing with limited inventory. *Operations Research, 65*(4), pp. 968–991. https://doi.org/10.1287/opre.2017.1596

Choi, T. M., Hui, C. L., Liu, N., Ng, S. F., & Yu, Y. (2014). Fast fashion sales forecasting with limited data and time. *Decision Support Systems, 59*(1), pp. 84-92. https://doi.org/10.1016/j.dss.2013.10.008

Cui, D., & Curry, D. (2005). Prediction in marketing using the support vector machine. *Marketing Science, 24*(4), pp. 595-615. https://doi.org/10.1287/mksc.1050.0123

Davidson, J., Liebald, B., Liu, J., Nandy, P., & Van Vleet, T. (2010). The YouTube video recommendation system. *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems*. https://doi.org/10.1145/1864708.1864770

Deshpande, M., & Karypis, G. (2004). Item-based top-N recommendation algorithms. *ACM Transactions on Information Systems*, *22*(1), pp. 143-177. https://doi.org/10.1145/963770.963776

Doyle, P., & Gidengil, B. Z. (1977). Review of In-Store Experiments. *Journal of Retailing*, *53*(2), pp. 47-62.

Eyjolfsdottir, E., Tilak, G., & Li, N. (2010). MovieGEN: A Movie Recommendation System. *Computer Science Department.*

Fallah Tehrani, A., & Ahrens, D. (2016). Enhanced predictive models for purchasing in the fashion field by using kernel machine regression equipped with ordinal logistic regression. *Journal of Retailing and Consumer Services, 32*, pp. 131-138. https://doi.org/10.1016/j.jretconser.2016.05.008

Fildes, R., Ma, S., & Kolassa, S. (2022). Retail forecasting: Research and practice. International *Journal of Forecasting, 38*(4), pp. 1283-1318. https://doi.org/10.1016/j.ijforecast.2019.06.004

Fisher, M., & Rajaram, K. (2000). Accurate retail testing of fashion merchandise: Methodology and application. *Marketing Science*, *19*(3), pp. 266-278.. https://doi.org/10.1287/mksc.19.3.266.11800

Fox, L. J. (1995). An integrated view of the assortment management process: The next frontier for leading retailers. *Chain Store Age Executive*, *71*(11), pp. 62-68.

Furukawa, T., Miura, C., Mori, K., Uchida, S., & Hasegawa, M. (2019). Visualisation for analysing evolutionary dynamics of fashion trends. *International Journal of Fashion Design, Technology and Education*, *12*(2), pp. 247-259. https://doi.org/10.1080/17543266.2019.1587789

Gallien, J., Mersereau, A. J., Garro, A., Mora, A. D., & Vidal, M. N. (2015). Initial shipment decisions for new products at Zara. *Operations Research*, *63*(2), pp. 269-286. https://doi.org/10.1287/opre.2014.1343

George, T., & Merugu, S. (2005). A scalable collaborative filtering framework based on co-clustering. *Proceedings - IEEE International Conference on Data Mining, ICDM*. https://doi.org/10.1109/ICDM.2005.14

Green, M., & Harrison, P. J. (1973). Fashion forecasting for a mail order company using a Bayesian approach. *Operational Research Quarterly*, *24*(2), pp. 193-205. https://doi.org/10.1057/jors.1973.37

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to Weave an Information tapestry. *Communications of the ACM*, *35*(12), pp. 61-70. https://doi.org/10.1145/138859.138867

Gomez-Uribe, C. A., & Hunt, N. (2015). The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*, *6*(4), pp. 1-19. https://doi.org/10.1145/2843948

Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. *Journal of Software*, *5*(7), pp. 745-752. https://doi.org/10.4304/jsw.5.7.745-752

Guță, A. J. (2017). The analysis of strategic alternatives using BCG matrix in a company. *Quality - Access to Success 18*, pp. 358-361.

Hollander, S. C. (1986). A rearview mirror might help us drive forward-a call for more historical studies in retailing. *Journal of Retailing*, *62*(1), pp. 7-10.

Huang, Z., Zeng, D., & Chen, H. (2007). A Comparative study of recommendation algorithms in e-commerce applications. *IEEE Intelligent Systems*, *22*(5), pp. 68-78.

Humaira, H., & Rasyidah, R. (2020). Determining The Appropiate Cluster Number Using Elbow Method for K-Means Algorithm. *In Proceedings of the 2nd Workshop on Multidisciplinary and Applications (WMA).* https://doi.org/10.4108/eai.24-1-2018.2292388

Jacobs, B. J. D., Donkers, B., & Fok, D. (2016). Model-based purchase predictions for large assortments. *Marketing Science, 35*(3), pp. 341-537. https://doi.org/10.1287/mksc.2016.0985

Kelleher, J., & Bridge, D. (2004). An accurate and scalable collaborative recommender. *Artificial Intelligence Review*, *21*(3), pp. 193-213. https://doi.org/10.1023/B:AIRE.0000036255.53433.26

Kharita, M. K., Kumar, A., & Singh, P. (2018). Item-Based Collaborative Filtering in Movie Recommendation in Real time. *ICSCCC 2018 - 1st International Conference on Secure Cyber Computing and Communications*. https://doi.org/10.1109/ICSCCC.2018.8703362

Kim, K. jae, & Ahn, H. (2008). A recommender system using GA K-means clustering in an online shopping market. *Expert Systems with Applications*, *34*(2), pp. 1200-1209. https://doi.org/10.1016/j.eswa.2006.12.025

Kohrs, A., Merialdo, B., & Merialdo, B. (1999). Clustering for collaborative filtering applications. *Computational Intelligence for Modelling, Control & Automation. IOS Press*, pp. 1-6. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.4809

Köppen, W. (1936): *Das geographische System der Klimate (Handbuch der Klimatologie, Bd. 1, Teil C).*

Lee, T. Y., & Bradlow, E. T. (2011). Automated marketing research using online customer reviews. *In Journal of Marketing Research, 48*(5), pp. 881-894. https://doi.org/10.1509/jmkr.48.5.881

Li, C., & He, K. (2017). CBMR: An optimized MapReduce for item-based collaborative filtering recommendation algorithm with empirical analysis. *Concurrency and Computation: Practice and Experience*, *29*(10). https://doi.org/10.1002/cpe.4092

Li, X. (2014). Application of decision tree classification method based on information entropy to web marketing. *Proceedings - 2014 6th International Conference on*

Measuring Technology and Mechatronics Automation, ICMTMA 2014. https://doi.org/10.1109/ICMTMA.2014.34

Liao, C. L., & Lee, S. J. (2016). A clustering based approach to improving the efficiency of collaborative filtering recommendation. *Electronic Commerce Research and Applications, 18,* pp. 1-9. https://doi.org/10.1016/j.elerap.2016.05.001

Lokhande, A., & Jain, P. (2019). Hybrid Collaborative Filtering Model Using Hierarchical Clustering and PCA. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3365525

Ma, S., & Fildes, R. (2021). Retail sales forecasting with meta-learning. *European Journal of Operational Research, 288*(1), pp. 111-128. https://doi.org/10.1016/j.ejor.2020.05.038

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1*.

Mendes, A. B., & Cardoso, M. G. M. S. (2006). Clustering supermarkets: the role of experts. *Journal of Retailing and Consumer Services*, *13*(4), pp. 231-247. https://doi.org/10.1016/j.jretconser.2004.11.005

Mitchell, S. (2009). An introduction to PuLP for Python Programmers. *The Python Papers Monograph, 1*(1).

Mitchell, S., O'Sullivan, M., & Dunning, I. (2011). *PuLP: A Linear Programming Toolkit for Python.* The University of Auckland, Auckland, New Zealand.

Mohana, H., & Suriakala, D. M. (2017). A Study on Ontology Based Collaborative Filtering Recommendation Algorithms in E-Commerce Applications. *IOSR Journal of Computer Engineering*, *19*(4), pp. 14-19. https://doi.org/10.9790/0661-1904011419

Mostard, J., Teunter, R., & de Koster, R. (2011). Forecasting demand for single-period products: A case study in the apparel industry. *European Journal of Operational Research, 211*(1), pp. 139-147. https://doi.org/10.1016/j.ejor.2010.11.001

Najafabadi, M. K., Mohamed, A., Balan Nair, M. A. L., & Tabibian, S. M. (2021). An effective collaborative user model using hybrid clustering recommendation methods. *Ingenierie Des Systemes d'Information*, *26*(2), pp. 151-158. https://doi.org/10.18280/isi.260202

Nguyen, L. V., Hong, M. S., Jung, J. J., & Sohn, B. S. (2020). Cognitive similarity-based collaborative filtering recommendation system. *Applied Sciences (Switzerland), 10*(12), pp. 1-14. https://doi.org/10.3390/APP10124183

Nilashi, M., Dalvi-Esfahani, M., Roudbaraki, M. Z., Ramayah, T., & Ibrahim, O. (2016). A Multi-Criteria Collaborative Filtering Recommender System Using Clustering and Regression Techniques. *Journal of Soft Computing and Decision Support Systems, 3*(5), pp. 24-30.

Phorasim, P., & Yu, L. (2017). Movies recommendation system using collaborative filtering and k-means. *International Journal of Advanced Computer Research, 7*(29), pp. 52-59. https://doi.org/10.19101/IJACR.2017.729004

Pollack, E. (1994). *Raising the Bar: Keys to High-performance Retailing'. Chain Store Age Executive, 70, 2MH-5MH.*

Ponnam, L. T., Deepak Punyasamudram, S., Nallagulla, S. N., & Yellamati, S. (2016). Movie recommender system using item based collaborative filtering technique. *1st International Conference on Emerging Trends in Engineering, Technology and Science, ICETETS 2016 - Proceedings*. https://doi.org/10.1109/ICETETS.2016.7602983

Purnima, B., & Arvind, K. (2014). EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. *International Journal of Computer Applications, 105*(9).

Qing, Y. X. (2015). An intelligent E-commerce recommendation algorithm based on collaborative filtering technology. *Proceedings - 7th International Conference on Intelligent Computation Technology and Automation, ICICTA 2014*. https://doi.org/10.1109/ICICTA.2014.27

Rajaram, K. (2008). Merchandise planning models for fashion retailing. *In International Series in Operations Research and Management Science*, *119*, pp.105-145. https://doi.org/10.1007/978-0-387-75240-2_5

Ranjan, A. A., Rai, A., Haque, S., & Pradeep K. Kushwaha, B. P. L. and. (2019). An Approach for Netflix Recommendation System using Singular Value Decomposition. *Journal of Computer and Mathematical Sciences*, *10*(4), pp. 774-779. https://doi.org/10.29055/jcms/1063

Renaud-Deputter, S., Xiong, T., & Wang, S. (2013). Combining collaborative filtering and clustering for implicit recommender system. *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*. https://doi.org/10.1109/AINA.2013.65

Rousso, C. (2012). *Fashion forward: A guide to fashion forecasting. New York: Fairchild Books.*

Saklecha, A., & Raikwal, J. (2017). Enhanced K-Means Clustering Algorithm Using Collaborative Filtering Approach. *Oriental Journal of Computer Science and Technology*, *10*(2), pp. 474-479. https://doi.org/10.13005/ojcst/10.02.31

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. *In Proceedings of the 2nd ACM Conference on Electronic Commerce*, pp. 158-167. https://doi.org/10.1145/352871.352887

Schafer, J. Ben, Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, *5*, pp. 1-2. https://doi.org/10.1007/978-1-4615-1627-9_6

Schwartz, E. M., Bradlow, E. T., & Fader, P. S. (2014). Model selection using database characteristics: Developing a classification tree for longitudinal incidence data. *Marketing Science, 33*(2), pp. 188-205. https://doi.org/10.1287/mksc.2013.0825

Sichel, B. (2008). Forecasting Demand with Point of Sales Data-A Case Study of Fashion Products. *The Journal of Business Forecasting*, *27*(4), pp. 15-16.

Smith, B., & Linden, G. (2017). Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Computing*, *21*(3). https://doi.org/10.1109/MIC.2017.72

Steenkamp, J. B. E. M., ter Hofstede, F., & Wedel, M. (1999). A cross-national investigation into the individual and national cultural antecedents of consumer innovativeness. *Journal of Marketing*, *63*(2), pp. 55-69. https://doi.org/10.2307/1251945

Stefani, M. A., Stefanis, V., & Garofalakis, J. (2019). CFRS: A Trends-Driven Collaborative Fashion Recommendation System. *10th International Conference on Information, Intelligence, Systems and Applications, IISA 2019*. https://doi.org/10.1109/IISA.2019.8900681

Stremersch, S., & Tellis, G. J. (2004). Understanding and managing international growth of new products. *International Journal of Research in Marketing, 21*(4), pp. 421-438. https://doi.org/10.1016/j.ijresmar.2004.07.001

Sun, H., Wu, T., Yan, M., & Wu, Y. (2012). A new item clustering-based collaborative filtering approach. *Proceedings - 9th Web Information Systems and Applications Conference, WISA 2012*, pp. 91-94. https://doi.org/10.1109/WISA.2012.30

Sun, Z. L., Choi, T. M., Au, K. F., & Yu, Y. (2008). Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems, 46*(1), pp. 411-419. https://doi.org/10.1016/j.dss.2008.07.009

Syntetos, A. A., Babai, Z., Boylan, J. E., Kolassa, S., & Nikolopoulos, K. (2016). Supply chain forecasting: Theory, practice, their gap and the future. *In European Journal of Operational Research*, *252*(1), pp. 1-26. https://doi.org/10.1016/j.ejor.2015.11.010

Tan, H., Guo, J., & Li, Y. (2008). E-learning Recommendation System. *In 2008 International conference on computer science and software engineering ,5,* pp. 430-433). IEEE. https://doi.org/10.1109/csse.2008.305

van den Bulte, C., & Stremersch, S. (2004). Social contagion and income heterogeneity in new product diffusion: A meta-analytic test. *Marketing Science, 23*(4), pp. 530-544. https://doi.org/10.1287/mksc.1040.0054

Vucetic, S., & Obradovic, Z. (2005). Collaborative filtering using a regression-based approach. *Knowledge and Information Systems*, *7*(1), pp. 1-22. https://doi.org/10.1007/s10115-003-0123-8

Wang, J., Robertson, S., De Vries, A. P., & Reinders, M. J. T. (2008). Probabilistic relevance ranking for collaborative filtering. *Information Retrieval, 11*(6), pp. 477-497. https://doi.org/10.1007/s10791-008-9060-1

Wang, X. (Shane), Ryoo, J. H. (Joseph), Bendle, N., & Kopalle, P. K. (2020). The role of machine learning analytics and metrics in retailing research. *Journal of Retailing 97*(4), pp. 658-675. https://doi.org/10.1016/j.jretai.2020.12.001

Wang, Z., Yu, X., Feng, N., & Wang, Z. (2014). An improved collaborative movie recommendation system using computational intelligence. *Journal of Visual Languages and Computing*, *25*(6), pp. 667-675. https://doi.org/10.1016/j.jvlc.2014.09.011

Werner, A. (2020). Organizing music, organizing gender: algorithmic culture and Spotify recommendations. *Popular Communication*, *18*(1), pp. 78-90. https://doi.org/10.1080/15405702.2020.1715980

Wilson, B. L. M. Kingdom, T. Reeve. 1995. Quick hits in store level merchandise and inventory management. *Chain Store Age Executive, 11*, pp. 103-106.

Wu, Z., Chen, Y., & Li, T. (2014). Personalized recommendation based on the improved similarity and fuzzy clustering. *Proceedings - 2014 International Conference on*

Information Science, Electronics and Electrical Engineering, ISEEE 2014, *2*, pp.1353-1357. https://doi.org/10.1109/InfoSEEE.2014.6947895

Xiaojun, L. (2017). An improved clustering-based collaborative filtering recommendation algorithm. *Cluster Computing*, *20*(2), pp. 1281-1288. https://doi.org/10.1007/s10586-017-0807-6

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999,* pp. 42-49. https://doi.org/10.1145/312624.312647

Yun, Y., Hooshyar, D., Jo, J., & Lim, H. (2018). Developing a hybrid collaborative filtering recommendation system with opinion mining on purchase review. *Journal of Information Science*, *44*(3), pp. 331-344. https://doi.org/10.1177/0165551517692955

ZENG, D., & CHEN, H. (2007). A comparison of collaborative-filtering recommendation algorithms for E-commerce. *IEEE Intelligent Systems*, *22*(5), pp. 68-78. https://doi.org/10.1109/MIS.2007.80

Zhang, J., Lin, Z., Xiao, B., & Zhang, C. (2009). An optimized item-based collaborative filtering recommendation algorithm. *Proceedings of 2009 IEEE International Conference on Network Infrastructure and Digital Content, IEEE IC-NIDC2009,* pp. 414-419. https://doi.org/10.1109/ICNIDC.2009.5360986

C.V

Personal Information

Name, Surname : Ramazan Arkan

Education

Bachelor's Degree	: Industrial Engineering, (Sabanci University) - 2008-2013
Masters' Degree	: Management of Business Administration, (Istanbul Bilgi University) – 2015-2017
Foreign Languages	: English

Job Experiences

Corporation Name and Date: LC Waikiki Mağazacılık (2014-…)