MİNE KARA SHELF SPACE ALLOCATION 2022

BIG DATA ANALYTICS CAPSTONE PROJECT

SHELF SPACE ALLOCATION

A RECOMMENDATION SYSTEM INTEGRATING MACHINE LEARNING

MİNE KARA

312102003

MEF UNIVERSITY

DECEMBER 2022

MEF UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

MASTER’S IN BIG DATA ANALYTICS

CAPSTONE PROJECT

SHELF SPACE ALLOCATION

Mine KARA

Prof. Dr. Özgür ÖZLÜK

Asst. Prof. Tuna ÇAKAR

DECEMBER 2022

MEF UNIVERSITY

This is to certify that I have read the graduation project and it has been judged to be successful, in scope and in quality and is acceptable as a graduation project Master's Degree in Big Data Analytics.

Project Advisor Prof. Dr. Özgür ÖZLÜK

This study has been approved in partial fulfillment of the requirements for the Master’s Degree in Big Data Analytics by the MEF University Graduate School of Science and Engineering.

Date: Seal/Signature:

ACADEMIC HONESTY PLEDGE

This is to certify that I have read the graduation project and it has been judged to be successful, in scope and in quality and is acceptable as a graduation project Master's Degree in Big Data Analytics.

Name Surname: Mine KARA

Signature:

ABSTRACT

SHELF SPACE ALLOCATION

Mine KARA

M.Sc in Big Data Analytics

Project Advisor: Prof. Dr. Özgür ÖZLÜK

(if any) Co-Advisor: Asst. Prof. Tuna ÇAKAR

December 2022, 54 Pages

This study is a shelf space allocation study carried out at the request of the Volt Motor. Problems with allocating shelf space can take on many different forms. The reason for this is because each firm has its own unique long-term strategy, management style, product categories, competitive climate, retailer-vendor relationship, shop layout, store size, fixture structure, and so on. It is quite doubtful that we will be able to devise a mathematical model that could accurately depict each and every issue that arises while allocating shelf space in the actual world. As a result, for the sake of this research, this thesis will primarily concentrate on an abstracted problem that may encapsulate the primary characteristics of the shelf space allocation challenges that are present in the majority of retail businesses.

In this study, the motor types and features produced by Volt Motor company were examined. It is planned to improve the level of inventory management efficiently with the machine learning models. First, prioritization was made among the motors, considering the warehouse layout of the company and the historical locations of the motor. Finally, shelves were assigned to the motors by building the recommendation system.

Keywords : Warehouse, retail, shelf space, optimization, machine learning, planning, recommendation systems

Numeric Code of the Field : 92431

ÖZET

RAF ALANI TAHSİSİ

Mine KARA

Büyük Veri Analitiği Tezsiz Yüksek Lisans Programı

Proje Danışmanı: Prof. Dr. Özgür ÖZLÜK

(Varsa) İkinci Proje Danışmanı: Asst. Prof. Tuna ÇAKAR

Aralık 2022, 54 Sayfa

Bu çalışma Volt Motor'un talebi üzerine gerçekleştirilen bir raf alanı tahsis çalışmasıdır. Raf alanı tahsisi ile ilgili problemler birçok farklı şekilde olabilir. Bunun nedeni, her firmanın kendine özgü uzun vadeli stratejisi, yönetim tarzı, ürün kategorileri, rekabet ortamı, satıcı-satıcı ilişkisi, mağaza düzeni, mağaza büyüklüğü vb. olmasıdır. Gerçek dünyada raf alanı tahsis edilirken ortaya çıkan her sorunu doğru bir şekilde tasvir edebilecek bir matematiksel model geliştirebilmemiz oldukça zordur. Sonuç olarak, bu araştırma öncelikli olarak, perakende işletmelerinin çoğunda mevcut olan raf alanı tahsis zorluklarının birincil özelliklerini özetleyebilecek bir soruna odaklanacaktır.

Bu çalışmada, Volt Motor firmasının ürettiği motor çeşitleri ve özellikleri incelenmiştir. Makine öğrenimi modelleri ile envanter yönetimi seviyesinin verimli bir şekilde iyileştirilmesi planlanmaktadır. Öncelikle firmanın depo yerleşimi ve motorun eski lokasyonları dikkate alınarak motorlar arasında önceliklendirme yapılmıştır. Son olarak öneri sistemi geliştirilerek motorlara raflar atanmıştır.

Anahtar Kelimeler : Depo, perakende, raf alanı, optimizasyon, makine öğrenimi, planlama, öneri sistemleri

Bilim Dalı Sayısal Kodu: 92431

CONTENTS

FIGURE TABLE

1. INTRODUCTION

Space is one of the most difficult resources to manage in retail. Retailers must allocate limited store space to a rising number of product categories in order to maximize sales, minimize the distance of best-selling products from the warehouse to the exit and any other related metrics. This appears to be a perfect task for a data mining approach; however, the representativeness of the available data is a barrier. Because changes in warehouse layout are rare and many different categories of stored products, generalization of the data is quite difficult. In this paper, we describe a recommendation system with machine learning algorithms to assign space to products. This study prioritizes products by first considering their sales data and other characteristics. It then creates a recommendation system, considering these values and the distances of the shelves to the warehouse exit.

The data in this study was provided by Volt Motor. Originally founded in 1966 as a winding manufacturer, Volt Motor expanded into the production of electric motors in 1987, first with single-phase motors and then with three-phase motors. Volt Motor is now one of Turkey's leading producers of electric motors. Since its founding, Volt Motor has made a positive impact on the electric motor industry through its groundbreaking initiatives and the incorporation of those ideas into the company's operations.

Volt Motor introduced electric motors in the IE3 efficiency class, which is the highest energy level, as a result of its innovative research and development efforts to lower our nation's electrical energy consumption and boost energy efficiency. With its own R&D and patent, it also manufactures IE4 and IE5 engines with the Super Premium energy level. Concurrently, Volt Motor, which became an R&D Center in 2021, maintains its international market collaboration.

Modern logistic warehouses and distribution centers are designed based on dozens of optimization studies, as they involve a lot of transaction processes during the day. Almost all businesses and organizations maintain an inventory of their tangible, physical assets as well as their corresponding intangible ones, such as documents, or data. Effective inventory management is crucial for supplying businesses to maintain predictability, control demand variances, and protect against unreliable supply. [1] The software market offers a wide range of solutions with various system requirements and capabilities. Choosing the right system for each company is not an easy task because it depends on many factors that must be considered.

Many ideas and methods used in warehouse management systems have been around for a while, but a lot has changed recently. Nowadays, computers can perform many tasks that were formerly carried out by humans constantly, more accurately, consistently, and reliably. As old methods have become less effective, computer models have grown in importance. Machine learning methodologies are one of these models. Machine learning is a technique that enables computer algorithms to recognize regularities, patterns, and structures in data without needing human participation to create models. Then, new data may be analyzed, and predictions can be made using these models. One of the most important objectives to meet client needs is this demand structure forecast. [2]

Literally, ML algorithms are used in many stages of the supply chain, upstream and downstream, including the following operations: scheduling, supply management, production, inventory management, and storage, followed by transportation and distribution. For instance, demonstrates how an inductive learning-based tool can dynamically identify the best supplier for the various nodes depending on the lead times and the quantity of orders, emphasizes how recurrent neural networks and support vector machines can provide extremely accurate forecasts for real-world datasets, leading to improved inventory control, and suggests a predictive methodology for forecasting near-real-time e-commerce order arrivals in distribution centers. [3]

The use of simulation tools and machine learning is crucial to business operations. Potential digitalization of business activities could boost productivity and effectiveness. Every company has found big data to be an asset, and managers are attempting to use this asset to improve performance in their organizations. The use of machine learning methods like support vector machines and random forests has increased profitability. In the era of internet commerce, it is crucial, especially for the sales team, to cognitively regress and evaluate their historical sales data with the use of non-human or instinctive prediction to validate their marketing hypothesis with supporting statistics. [4]

For a warehouse management system to be properly deployed, several things must be examined. This is particularly true for warehouses where many tasks and issues were performed by hand and where there was no prior organized workflow. Adding a warehouse management system to an existing warehouse demands more attention than establishing a planned, structured warehouse with a detailed integrated and operating workflow. There are various causes for this matter. The largest problem is the resistance from the staff because the installation of any new system ultimately signifies a huge paradigm shift in their daily routines. The effective introduction of the new model to the workforce is of utmost importance because it will determine the system's ultimate success. The next thing to keep in mind is where things are currently placed in the warehouse. This is a direct result of the workflow management that came before it. The system may have serious problems due to the variability of the resulting item placement. Other problems include inaccurate data inputs—if there is any data gathering at all—ineffective space usage, the replacement of warehouse racks, which raises operational expenses, inconsistent worker conduct, etc.

There are many ways to categorize warehouse activities, but one of the simplest is that there are four core activities: receiving, storing, putting away, picking, and shipping. The delivery of items to storage locations, identification, assignment of a storage bin, and put-away make up the storing processes. The process of "putting away" involves choosing a storage container based on the items' physical characteristics and weight, which necessitates a precisely chosen storage site. It is also stated in that those two operations take up to 15% of all the operating costs. Because of this, these procedures may be seen as a niche for enhancing the effectiveness of a future system. [5]

Shelf space is one of the most expensive retail resources. Allocation of products' limited storage space is always a problem for retailers to solve and they frequently make decisions about how to optimize. Optimized shelf management can increase profits by attracting customers, and reduce time spent to find and prepare the product for shipping. Defining enough shelf space for a product and determining the best product placement on shelves are critical issues in shelf space allocation. One of the most essential considerations in getting an edge in the cutthroat retail market is the strategic use of shelf space. As their selection of goods grows, retailers face a formidable obstacle. [6]

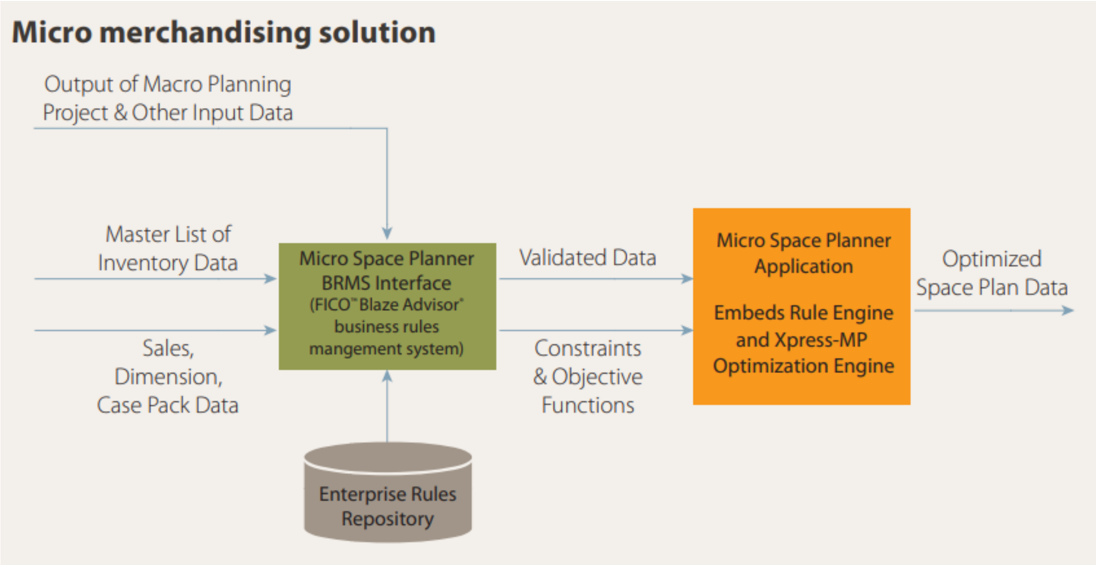


Figure 1 - Retail Space Planning and Optimization Workflow (FICO) ***[7]***

Within the restrictions of available shelf space, shelf space planning determines the optimal placement of items and the number of facings for each option. The amount and placement of the product on the shelf may affect its actual sales. The shelf space allocation problem (SSAP) is a decision issue with the objective of maximizing profit while satisfying some store-specific operational constraints. It is considered a natural continuation of the knapsack issue. The SSAP's main goal is to figure out what kind of product displays would bring in the most money.

Considering just length limit as shelf capacity parameter, only fundamental restrictions, and not including capping and nested allocation parameters in all models are the key omissions in the SSAP literature.[8]

Due to the nonlinear nature of SSAP and the numerous practical aspects, it must take into account, the solution technique becomes computationally difficult. Since this is the case, various effective strategies for tackling this problem have been presented since some models are unsolvable using commercially available software.

Normal stores, especially those that sell clothing, have both goods that sell in steady numbers year-round and seasonal favorites. As more information becomes readily available, it becomes more critical than ever to reliably gauge demand and promptly address fluctuations in consumer preferences. [9]

When examined solely through the lens of their definitions, the storing and put-away actions constitute challenging mathematical problems that can be difficult to solve. When the volume and capacity of the warehouse racks as well as the capacity of the things that are being received are taken into consideration, a form of knapsack problem emerges. This problem has to be eased, and the capacities of things and storage space should no longer be taken into consideration. Given how difficult it is to put such a system into practice and the lack of volume of information, this problem needs to be addressed. If the warehouse is originally empty, this problem may be simplified into a selling frequency-based ordering problem. However, if the warehouse is already operating at full capacity and a new system has to be deployed, the problem becomes more difficult. [10]

2. LITERATURE REVIEW

Both the field of marketing research and the field of operations research have shown interest in the topic of shelf space allocation for their respective research projects. This study presents empirical studies that illustrate and test theoretical relationships of the factors that determine brand space allocation. Additionally, mathematical models are developed for calculating optimal brand space for competing companies based on the findings of these empirical studies. This paper provides a review of the previous research that attempts to present empirical and quantitative models on the decisions regarding shelf space allocation problem (SSAP).

In retail layout, you have to make decisions about both the physical structure of the store (aisles, shelves, etc.) and the way shelf space is used. The shelf space allocation problem is a hierarchy of decisions that begins with placing products into departments, then into racks within those departments, and finally allocating rack shelf space (defined by number of facings of a specific product) to specific products and onto the racks themselves. Several academic works (Corstjens and Doyle, 1981; Botsali and Peters, 2005; Irion et al., 2012; Flamand et al., 2016) have examined this decision-making issue in the retail and marketing context over the years. Even though the store's layout is planned before shelf space is allocated, the latter is based on estimates of how likely it is that a consumer will see a certain section of racks. In other words, the problem with the physical layout is followed by the problem of how to divide up shelf space.

The spatial elasticity concept is first introduced in the literature by Hansen and Heinsbroek [1979], Corstjens and Doyle [1981], and Zufryden [1986]. In the years that followed, researchers improved upon these models by doing things like including assortment decisions and stock-out costs into one framework (Borin et al., 1994). Considering a stockroom for restocking merchandise was an extension of preexisting models, as noted by Urban [1998]. Irion et al. [2012] give a comprehensive model with a linearization strategy, complete with a full cost function and cross-space elasticity.

Hansen and Heinsbroek modeled demand based on the main shelf space impact as part of their model for allocating shelf space. However, they ignored the cross effects that occurred across different types of products. To modify the demand function, they are multiplicatively instead of additively. They reasoned that since sales will be directly proportionate to the amount of shelf space a product is given, there must be a linear relationship between the two. They accounted for assortment homogeneity and assumed that the space elasticity component is constant by including it in the demand function.

Corstjens and Doyle (1981) were the first to advance a model for optimizing shelf space that considered the inter-cross elasticity impact between products in the shop as well as the product-space elasticity effect. Much other researchers have made use of their findings by either adopting the original demand model verbatim or making just minor adjustments to it. They acknowledged that elements beyond the confines of physical space, such as advertising, special offers, and pricing, all have a role in influencing demand, but they believed these elements would remain constant in their demand forecast.

According to Zufryden's (1986) hypothesis, product demand is a scalar function of a vector that includes all aspects that may affect demand, such as but not limited to advertising, promotions, shelf space, and retail pricing. Zufryden made the space impact the dominant effect, assumed the other non-space demand factors were constant and disregarded the cross-elasticities between products in the shop so that his analysis could be easily implemented. The resulting demand model was the first of its kind to consider shelf depth in addition to shelf width.

Also considered by Yang and Chen (1999) was the impact of a product's placement on shelves of varying sizes and orientations. Their model assumed that all other marketing variables, other from space, are fixed, and it included cross-elasticity between items. They developed a demand model that accounts for the orientation of the shelf display.

For their demand model, Irion et al. (2004) consider both the number of facings and the width-lengths of the products in question. They explain the relationship between the product and its dimensions, including the width-length diversity of assortments and hinting that the width-length of a display front would influence demand. Therefore, they model the demand by considering the entire width-length area rather than the number of facings. The demand is modeled with the overall width-length of a product rather than the total quantity of items, making this model comparable to that of Hansen and Heinsbroek (1979). However, the cross-space effect between products in the assortment was not considered by Hansen and Heinsbroek. Demand is simulated by stacking products in ascending order of their one-dimensional regions until no more space is available. The authors argue that the demand is determined by a one-dimensional area allocation (total width x length of the product allocation), even though it may appear that the demand model can be expressed in a two-dimensional space due to the stacking process. This is because the area remaining between the stacked products and the heaviest product is what matters most. For this reason, their demand function was based solely on the area of the product in a single dimension, rather than the area of the entire product. In a category where items are otherwise similar in height, we think this could be a safe assumption to make. Because of this, the overall height of the products would be relatively consistent because the same number of display facings could be stacked in the available shelf space. Consumers' space allocation perception depends more on the allocated space of the total width-length of the products (consistent with the shelf space allocation literature), which suggests that demand might not be affected by the deviations of the uncovered area on top of the shelf caused by different (but close) product heights.

Peters et al. (2004) were the first to examine the retail layout issue. They examined three distinct store layouts: aisle, hub-and-spoke, and serpentine. They devised a strategy for assigning departments (groups of products) to locations in order to maximize revenue from impulse purchases. They used "visits" to figure out how many impulse purchases were made. This was based on the idea that a product could be bought on the spot if it was seen by the shopper while he or she was making other purchases. Peters et al. considered that a shopper's vision is perpendicular, indicating that a product must be visible if it is adjacent to the consumer. Although this is a decent approximation of visibility, the human eye often only perceives what is immediately in front of it.

Yapicioglu and Smith (2012) looked at the best way to set up a departmental block layout (which includes departments and aisle space) to make the most money for the store. This meant figuring out the size and location of each department within a racetrack structure. They also thought that shoppers were more likely to buy something on the spot in departments that were busy. To make up for this, a way was found to put departments with high impulse rates in places where there were a lot of shoppers. Yapicioglu and Smith (2012) changed this problem so that it had two goals. The first goal was to maximize store revenue based on the layout of the departments, and the second goal was to maximize the satisfaction 5 of departmental adjacencies. Visibility was generally defined as the area around a shopper. High traffic areas mean that many shoppers will be there, so products in those areas will be seen.

Pinto and Soares (2013) made a space allocation system called a Decision Support System (DSS) that uses a machine learning method and a meta-heuristic optimization method. This DSS is intended to help with the process of allocating space. The meta-heuristic looks through the space and finds all the space allocations that are acceptable. The individual estimates of sales for each product category are derived using sales forecasting models, and then the associated total sales for each potential solution are predicted based on those individual forecasts. These models take into account a wide variety of parameters about the shop and the type of goods being modeled, not only the quantity of space allocated to the latter. The models are discovered by the application of methods of machine learning to historical data.

This method presents several difficulties, the most significant of which are the evaluation of the individual models, the evaluation of the complete system, and the representativeness of the data. Because there are only occasional shifts in the amount of shelf space dedicated to a given product category, the data collected by retail companies only represent a minute portion of their total domain. As a result, it is challenging to create models that have a high capacity for generalization. They devised a method of measuring the spatial volatility of different product categories so that they could address this issue. Surprisingly, however, the best models were not achieved on the product categories that, according to our criteria, had more space volatility.

There are two ways to determine whether the data for a certain product category is truly representative. However, empirical research showed that the metrics were poor predictors of the models' accuracy in making predictions. Data from a recently implemented, store-wide layout modification was used to assess the effectiveness of the system. According to the findings, the system's suggestions for retail space allocations are highly consistent with those of domain experts. Moreover, from a business standpoint, some of the recommendations' discrepancies were intriguing. They've come up with their own unique models that don't rely on anyone else's. Although it's possible that some product categories' sales are dependent on one another. To solve this issue, they want to try out some multi-target regression techniques.

3. PROJECT DESCRIPTION

Problems with allocating shelf space can take on many different forms. The reason for this is because each firm has its own unique long-term strategy, management style, product categories, competitive climate, retailer-vendor relationship, shop layout, store size, fixture structure, and so on. It is quite doubtful that we will be able to devise a mathematical model that could accurately depict each issue that arises while allocating shelf space in the actual world. As a result, for the sake of this research, this thesis will primarily concentrate on an abstracted problem that may encapsulate the primary characteristics of the shelf space allocation challenges that are present in most retail businesses.

This study is a shelf space allocation study carried out at the request of the Volt Motor. The data used in this project is provided by the Volt motor. In general, it is related to their products and sales. It is planned to improve the level of inventory management efficiently with the ML models. Necessary data was retrieved from SSMS with SQL codes and models are generated. In general Volt Motor’s data is grouped as location structure, master data-material, master data-partner, grouping, material movements, operation, packaging, serial numbering, delivery, and production. The data obtained for this study includes the following information:

Motor Features I dataset: (4141 rows, 8 columns)

This dataset contains motor properties (power, yield, etc.)

- **Material ID:** Motor unique material ID
- **Material No:** Unique ID for the material type of the motor
- **Definition:** Description of motors
- **Power (kW):** Motor power
- **Pole Quantity:** Motor Pole Count (The number of permanent magnetic poles, both north and south, that are located on the rotor of a motor is referred to as the pole count. On the rotor, the number of north poles and south poles will always be equal.)
- **Yield Class:**
  Efficiency classes developed in 2008 according to IEC (International Electrotechnical Committee) 60034-30:2008 standard have survived until today. These classes are IE1, IE2, IE3 and IE4. The most efficient class is IE4, but the least efficient is IE1.
- **Body:** Body size of motor (80S/M/L, 90S/M/L, etc.)
- **Structure Shape:** Shape of motor

Motor Features II dataset (9391 rows, 2 columns)

This dataset is a dataset that includes the weights of the motors, among other features. By combining this dataset with Motor Features I, all features of motors are collected later in one dataset.

- **Material ID:** Motor unique material ID
- **Weight:** Motor weight
- **Total Weight:** Total weight of the unique motors

Volt Motor dataset: (2209489 rows, 11 columns)

This dataset contains the information about the production, supply, and delivery of the motors. The shipping speed of the motors will be obtained from this data set.

- **Serial ID:** Motor unique ID
- **Serial No:** Motor unique number
- **Material ID:** Motor unique material ID
- **Material code:** Motor unique material number
- **Production Date:** Production Date of Motor
- **Supply ID:** Order ID
- **Exit Date:** Exit date from warehouse, shipment date
- **Delivery Document No:** Delivery document to the customer
- **Delivery Date:** Delivery date to the customer
- **Stock:** Amount of availability

Historical Shelf Placement dataset: (2320709 rows, 5 columns)

This dataset contains the shelves where the motors were previously placed in the warehouse.

- **Serial ID:** Motor unique ID
- **Serial No:** Motor unique number
- **Material ID:** Motor unique material ID
- **Material code:** Motor unique material number
- **Shelf Name:** Shelf names where the motors are placed

Warehouse Plan & Shelf dataset:

The warehouse plan obtained from the Volt Motor is in excel format as seen in figure 2. The yellow part shows the warehouse exit. The left part of the warehouse is the preparation area for the shipment. The right part shows the blocks (A1, A2, B1, C1… etc.) and shows shelves (A1-01, A1-02…, B1-01... etc.). Also, much other information is given in the excel file such as the dimensions of corridors, shelves, and preparation areas (Appendix A).

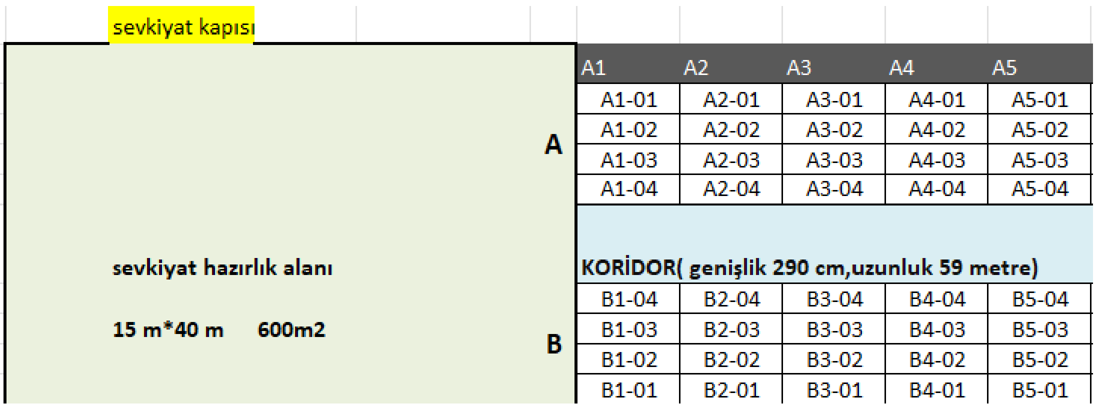


Figure 2 - Warehouse Layout in excel format (a part of)

This warehouse plan in excel format has been converted into a pandas dataframe (Figure 3). The sections surrounded by the yellow area seen below are the widths, the expression “4.6” represents the exit door. The red area is the preparation area for the shipment and the green zones indicate shelves. (Appendix B)

	0	1	2	3	4	5	6	7	8	9	10
0	2.8	2.8	4.6	2.0	2.8	2.8	2.8	2.8	2.8	2.8	2.8
1	1.05	0.0	0.0	0.0	0.0	0.0	A1	A2	A3	A4	A5
2	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0
3	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0
4	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0
5	1.05	0.0	0.0	0.0	0.0	0.0	B1	B2	B3	B4	B5
6	1.05	0.0	0.0	0.0	0.0	0.0	C1	C2	C3	C4	C5
7	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0

Figure 3 - Warehouse Plan in the form of pandas dataframe

4. STUDY

This section will focus on finding a solution for the defined space allocation problem. The whole process is split into different stages: analysis and prioritization of products (scoring) and shelf recommendation for products according to historical placement and distances of the shelf to the exit.

4.1 Analysis of Products and Scoring

Product scores are calculated from the formula below according to the total weight of the products and their shipping speed. Also, shipping speed is found from the difference between the production date and the exit date.

Score Formula:

$$\text{Score} = \log\left(\left(\text{Total Weight} / \text{Shipping Speed}\right) + 1\right)$$

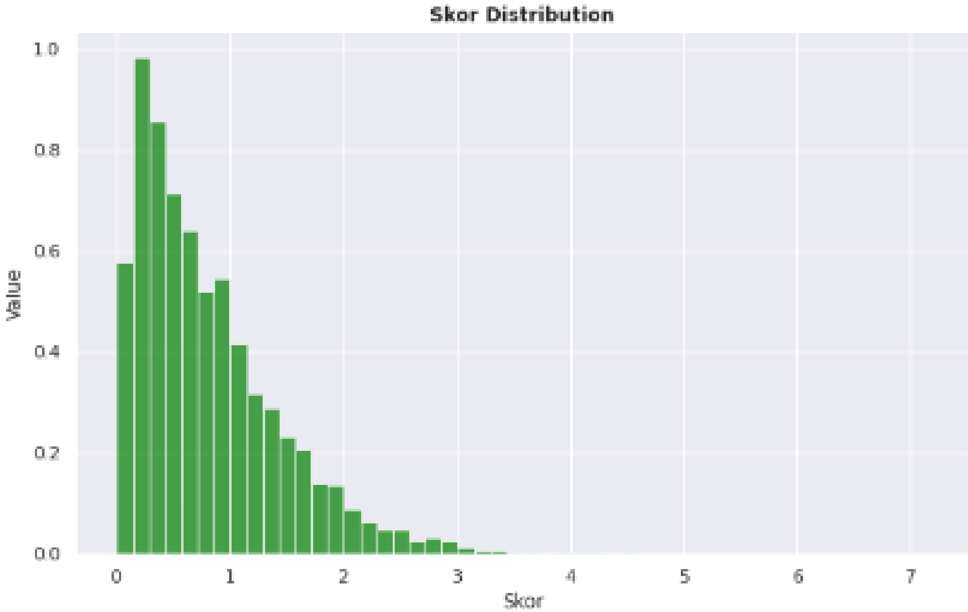


Figure 4 - Score Distribution

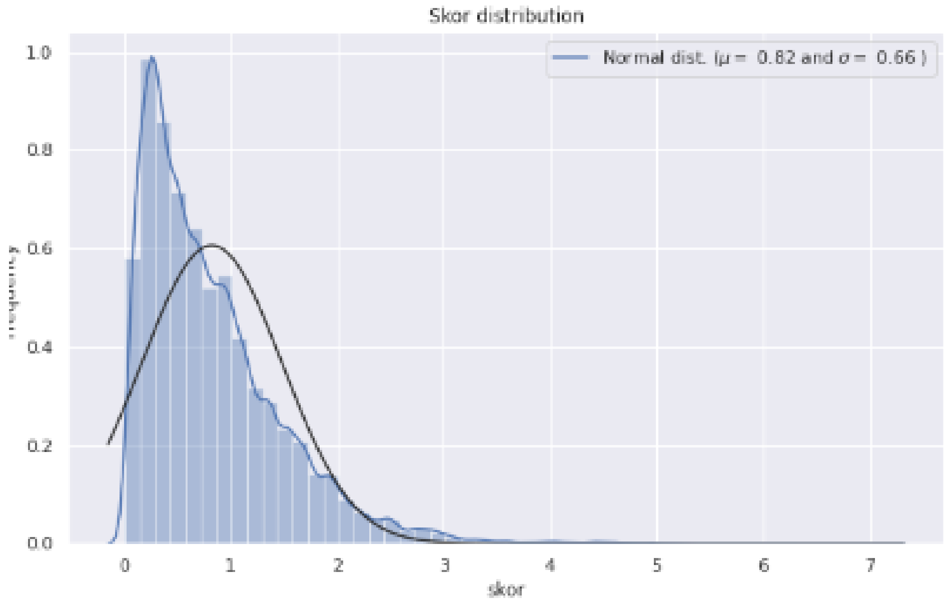


Figure 5 - Normal Distribution

Initially, exploratory data analysis was carried out. First, motor features I, motor features II and volt motor dataset were combined, and all motor features were collected in a dataset. Then invalid or NAN rows are eliminated.

```python
# Drop all row with "GEÇERSİZ"
final_df = final_df[(final_df != 'GEÇERSİZ').all(axis=1)]
```

```python
df = df1[~(df1[['Gövde','Güç [kW]']].isna()).any(axis=1)]
```

Figure 6 - Drop NAN Values

In general, the unique values of the columns were checked and the incorrectly entered ones were either eliminated or replaced with the real value.

```python
for i in final_df.columns:
  print(i,":",final_df[i].unique())
```

```
Serial_id : [4189197 2028896 4187902 ... 2948686 1692885 1711777]
Serial_no : ['MP02682307' 'MP01487385' 'MP02681149' ... 'MP01984722' 'MP01318877'
 'MP01328345']
Material_code : ['VSPA90S4AB348MIS' 'V2EA90S4CB3440BF' 'V2EA100L4JB3440AI' ...
 'V1EA71M2CB34KSB' 'V3EA132S2DB346KSB' 'V2EA132S4GB3566UL']
prod_date : ['2020-12-23 22:21:28.457' '2019-10-01 15:41:46.813'
 '2020-12-22 12:46:18.950' ... '2020-04-29 16:08:11.040'
 '2019-06-21 16:02:02.453' '2019-06-28 11:07:09.850']
sup_id : [4214401. 2029201. 4194204. ... 2954560. 1692908. 1713797.]
out_date : ['2020-12-30 07:57:45.033' '2019-10-04 08:00:15.067'
 '2020-12-31 12:32:44.090' ... '2021-06-21 09:21:08.617'
 '2020-05-04 07:48:09.520' '2019-06-25 08:25:30.163']
delivery_doc_no : ['0070084644' nan '0070084426' ... '0070072819' '0070092691' '0070073434']
dlv_date : ['2020-08-12 00:00:00.000' '2019-09-04 00:00:00.000'
 '2021-05-18 00:00:00.000' ... '2021-02-19 00:00:00.000'
 '2017-06-01 00:00:00.000' '2020-03-27 00:00:00.000']
Stok : [nan 'STOKTA']
material_id : [70895 64237 37837 ... 90274 72292 71904]
material no : ['VSPA90S4AB348MIS' 'V2EA90S4CB3440BF' 'V2EA100L4JB3440AI' ...
 'V1EA71M2CB34KSB' 'V3EA132S2DB346KSB' 'V2EA132S4GB3566UL']
definition : ['0,55KW 22X84 K.SUZ P.Ş SY SEZER S.BOYA' '1,1 KW 4P TRF 90S 2/4V IE2 B34'
 '2,2KW 4P TRF 100 2/4V IE2 AI B34' ... 'X0,55KW 2P TRF 71 IE1 B34 KSB'
 '5,5 KW 2P TRF 132S 400V 60HZ IE3 KSB'
 '5,5KW 4P 132S 600V 60HZ IE2 BF UL B35']
Güç [kW] : [0.55 1.1 2.2 0.18 5.5 3 4 0.37 7.5 22 0.75 0.25 1.5 '1,0-1,7' '1,5-2,5'
 '0,3-0,44' 15 11 0.12 '1-1,7' '1,3-1,8' '1,1-1,5-1,85' '1,0-1,3' nan
 '0,60-0,90' '2,4-3,0' 18.5 '1,5-2,0-2,5' '1,8-2,2' 30 0.2 '7,5-11' 5
 '0,15-0,55' '0,35-0,55' 1 37 1.6 '0,70-0,85' 110 '3,7-4,5' '1-1,3'
 '0,25-0,37' '4,7-6' '0,75-1,1' 9.2 '160W' 90 55 '0,6-0,9' 3.7 '0,2-'
 '5,5-7,5' 12 '5,5-6,3']
```

```python
# Correcting format of 160W as other power values
final_df.loc[final_df['Güç [kW]'] == '160W', 'Güç [kW]'] = 0.16
```

Figure 7 - Correcting the wrong values

The data types of the columns were checked and brought to the appropriate format. Unusable columns have been dropped.

```
df["TOPLAM_AGIRLIK"] = df["TOPLAM_AGIRLIK"].replace(to_replace=",", value=".", regex=True)
df[['TOPLAM_AGIRLIK']] = df[['TOPLAM_AGIRLIK']].astype(float)
```

```
df.describe()
```

	Serial_id	sup_id	material_id	TOPLAM_AGIRLIK	Depo_Sevk_Hızı(Gün)
count	2.904160e+05	2.904160e+05	290416.000000	290416.000000	290416.000000
mean	2.745501e+06	2.753736e+06	59794.366767	485.185512	28.657343
std	1.916983e+06	1.920446e+06	30004.396506	170.208497	48.662749
min	1.134600e+04	1.197000e+03	16589.000000	5.000000	0.034336
25%	9.350065e+05	9.342100e+05	37833.000000	374.000000	5.990030
50%	2.539357e+06	2.544118e+06	64742.000000	480.606061	14.056175
75%	4.382693e+06	4.399051e+06	84415.000000	598.240458	32.652754
max	6.684390e+06	6.684401e+06	137018.000000	2600.000000	1326.019145

Figure 8 - Data Types Check

Columns required for the analysis such as shipping speed and score values were obtained. Shipping speed was found by subtracting the exit date from the production date. And the production date and exit date columns were also dropped from the dataset.

```
# Dispatch Rate is calculated by taking the difference between the product's entry into the warehouse and shipment dates.
df['Depo_Sevk_Hızı(Gün)'] = df['out_date'] - df['prod_date']
df['Depo_Sevk_Hızı(Gün)'] = df['Depo_Sevk_Hızı(Gün)'].apply(timedelta.total_seconds)/86400
```

	Serial_id	sup_id	material_id	AGIRLIK	TOPLAM_AGIRLIK	Depo_Sevk_Hızı(Gün)
count	2.904160e+05	2.904160e+05	290416.000000	290416.000000	290416.000000	290416.000000
mean	2.745501e+06	2.753736e+06	59794.366767	16.046560	485.185512	28.657343
std	1.916983e+06	1.920446e+06	30004.396506	14.142469	170.208497	48.662749
min	1.134600e+04	1.197000e+03	16589.000000	3.900000	5.000000	0.034336
25%	9.350065e+05	9.342100e+05	37833.000000	8.000000	374.000000	5.990030
50%	2.539357e+06	2.544118e+06	64742.000000	12.000000	480.606061	14.056175
75%	4.382693e+06	4.399051e+06	84415.000000	18.000000	598.240458	32.652754
max	6.684390e+06	6.684401e+06	137018.000000	679.200000	2600.000000	1326.019145

Figure 9 - Shipping Speed

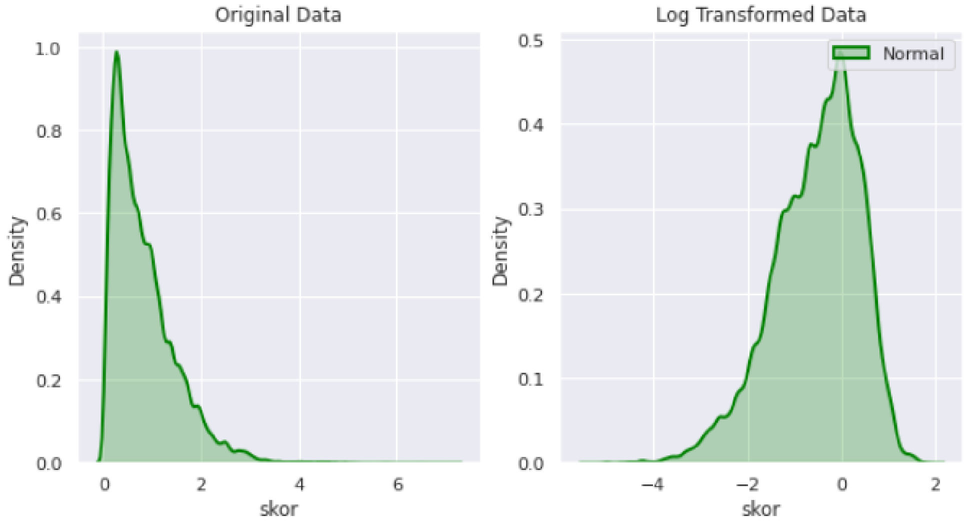
```
df["skor1"] = (df["TOPLAM_AGIRLIK"]/df["Depo_Sevk_Hızı(Gün)"])

df["skor"] = np.log((df["TOPLAM_AGIRLIK"]/df["Depo_Sevk_Hızı(Gün)"])+1)
```

Figure 10 - Score Values

There are outliers in the data. Log, boxcox, squareroot and reciprocal transformation were applied to eliminate these outliers and obtain a normal distribution.

```
Text(0.5, 1.0, 'Log Transformed Data')
```

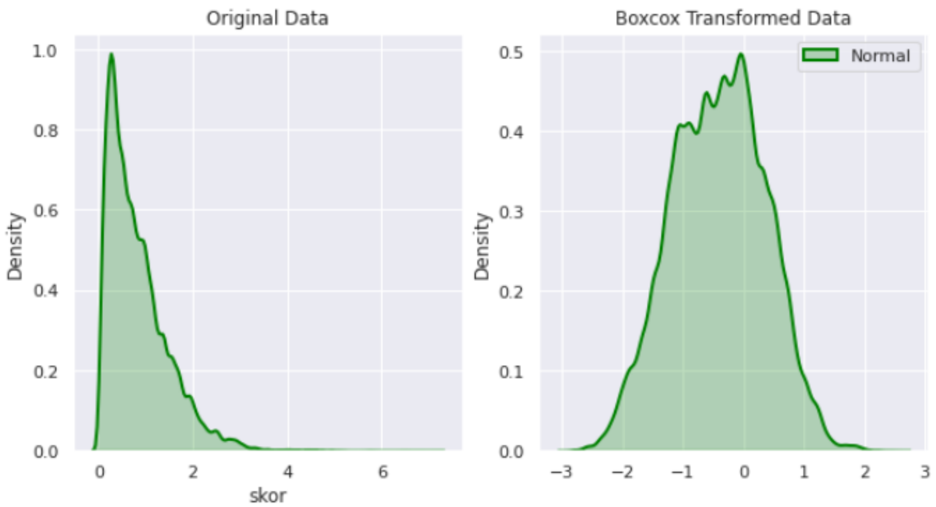



```
shapiro(data_log)
```

```
ShapiroResult(statistic=0.9735022783279419, pvalue=0.0)
```

Figure 11 - Log Transformation

```
Lambda value used for Transformation: 0.2607958613759189
```

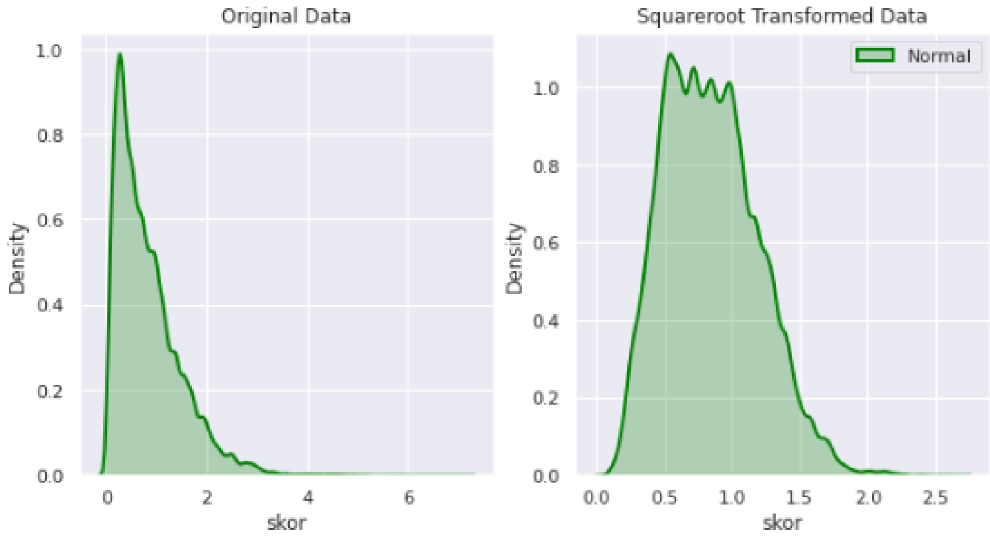



```
shapiro(data_bx)
```

```
ShapiroResult(statistic=0.9968959093093872, pvalue=0.0)
```

Figure 12 - Boxcox Transformation

```
Text(0.5, 1.0, 'Squareroot Transformed Data')
```

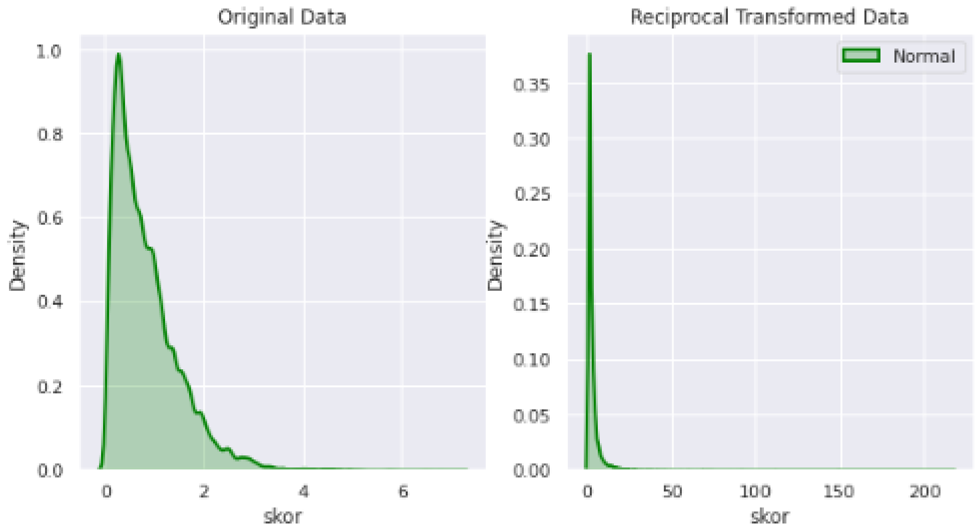



```
shapiro(data_sq)
```

```
ShapiroResult(statistic=0.9823850393295288, pvalue=0.0)
```

Figure 13 - Square root Transformation

```
Text(0.5, 1.0, 'Reciprocal Transformed Data')
```




```
shapiro(data_rp)
```

```
ShapiroResult(statistic=0.41192352771759033, pvalue=0.0)
```

Figure 14 - Reciprocal Transformation

Looking at these plots above, it seems that boxcox and squareroot transformations are closer to the normal distribution. However, to decide which one to use, the shapiro wilky test was used and continued with the one with the largest statistical value.

Before building the model, the correlation matrix was plotted to see the correlation rates of the features. In this correlation matrix, we see the relationship of total weight and shipping speed with the score values of the motors. While the shipping speed does not affect the score value of the motors, the total weight clearly contributes (with a ratio of 0.3) to the score value.

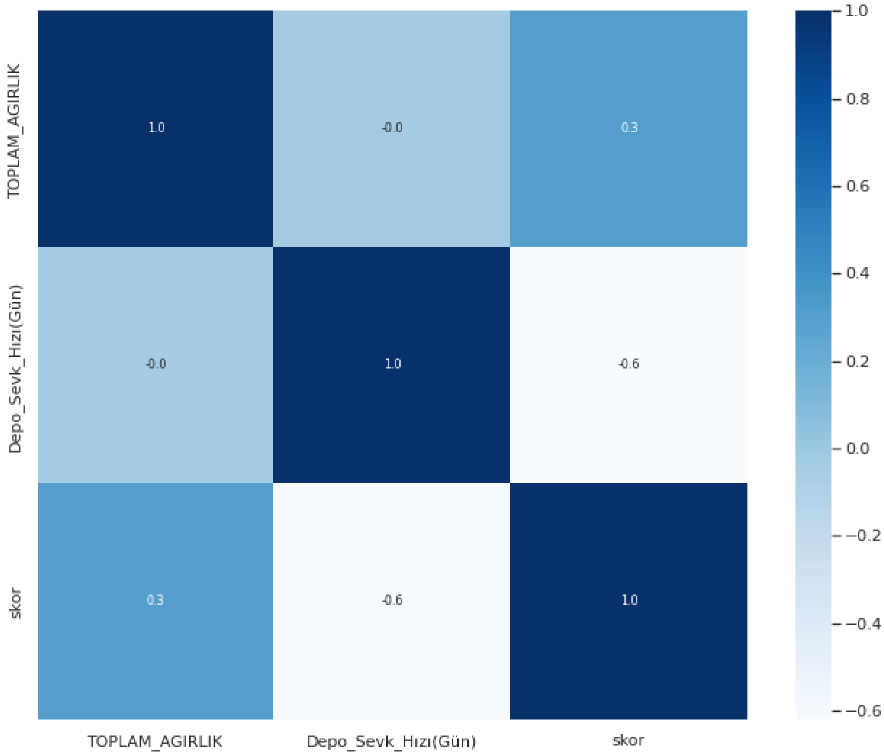


Figure 15 - Correlation Matrix

Several plots have been obtained to see the relationship of the score values more clearly with weight, total weight, and shipping speed. In Figure 16, we see the relationship between the total weight of the products and the score values. The total weight of the products is generally less than 1000 kg. On the other hand, score values vary.

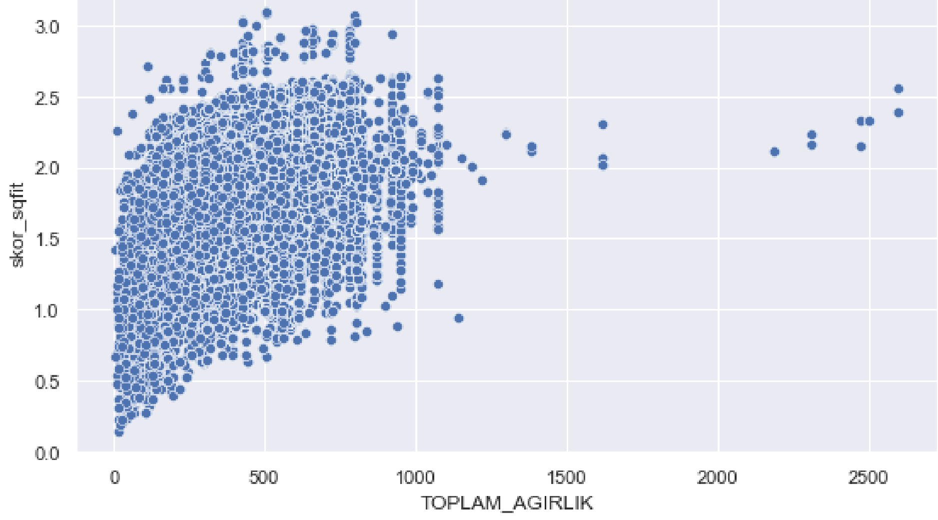


Figure 16 - Total Weight – Score Plot

In Figure 17, we see the relationship between the unit weight of the products and the score values. We can say that the products are generally divided into 3 groups by weight. Those weighing less than 200, between 200 and 300, and over 600. However, as seen in the graph below, there is no significant relationship between weight and score value.

```
sns.regplot(x="AGIRLIK", y="skor_sqfit", data=df1, fit_reg=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e45225450>
```

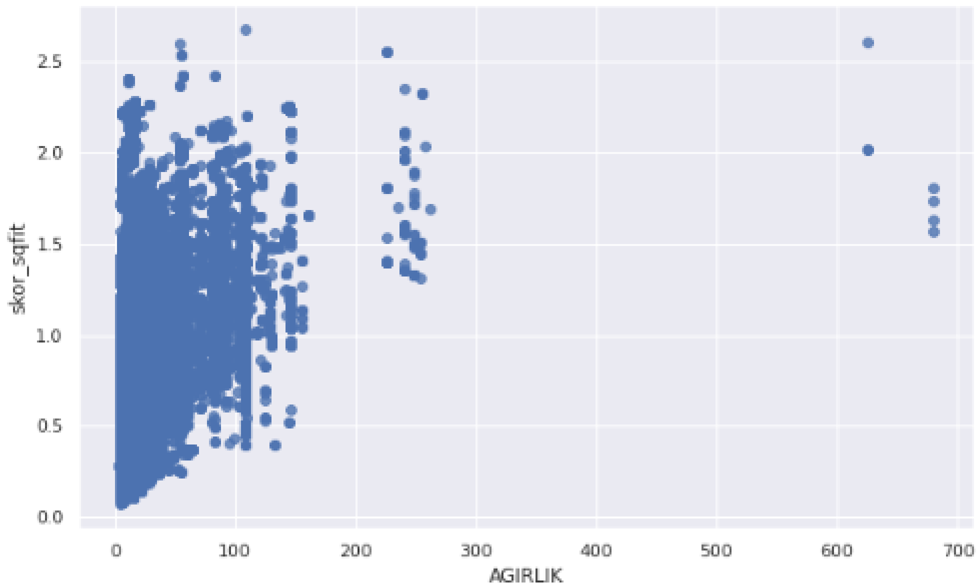


Figure 17 - Unique Weight of engines – Score Plot

In Figure 18, it can be seen the relationship between the shipping speeds of the products and the score values. It is seen in the graph below that as the shipping speed increases, the score value of the products decreases.

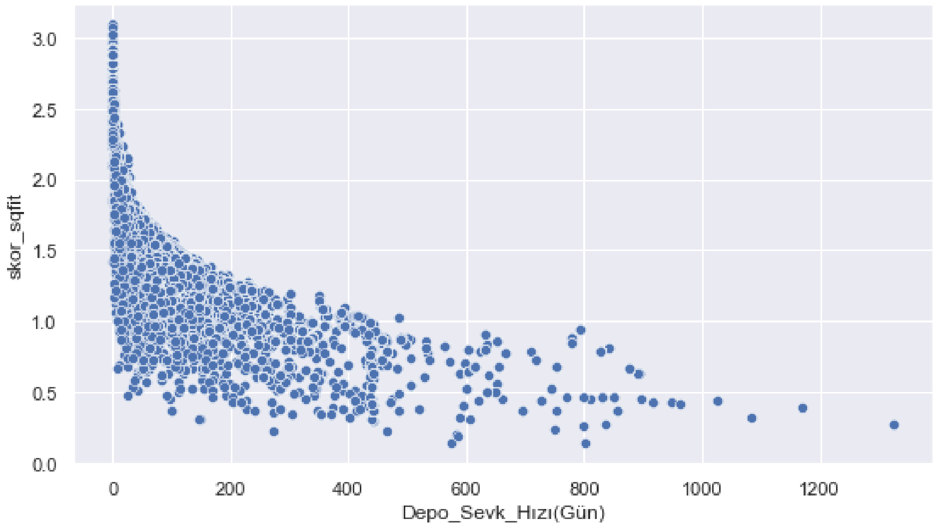


Figure 18 - Shipping Speed – Score Plot

Before moving on to the next step, one step has to be done. That step is normalization. Normalization is a method typically done as part of data preparation for machine learning. The purpose of normalization is to convert the numerical values of the dataset's numerical columns to a standard scale while preserving the original meaning of the data. In machine learning, the most common normalizing techniques are:

Mix-Max Scaling: To use this technique, take the difference between the lowest and highest values in each column and divide that result by the range. Each new column can take on the values between 0 and 1 inclusively.

Standardization Scaling: The word "standardization" is used to describe the procedure of setting a variable's mean to zero and its standard deviation to one. The method involves first subtracting the mean of each observation and then dividing it by the standard deviation.

4.1.1 Score Prediction based on total weight and shipping speed

4.1.1.1 Ridge and Lasso Regression

To avoid overfitting, polynomial ridge and lasso regression models were created by using shipping speed and total weight as features to estimate the score value. Prediction scores and error rates were obtained.

- **Ridge Regression**

```
model_ridge = make_pipeline(PolynomialFeatures(3), Ridge(alpha=10))
model_ridge.fit(X_train, y_train)
y_pred = model_ridge.predict(X_test)
model_ridge.score(X_test, y_test)
```

```
0.7979635119960549
```

```
[ ] mean_squared_error(y_test, y_pred)
```

```
0.20203648800394516
```

Figure 19 - Ridge Score

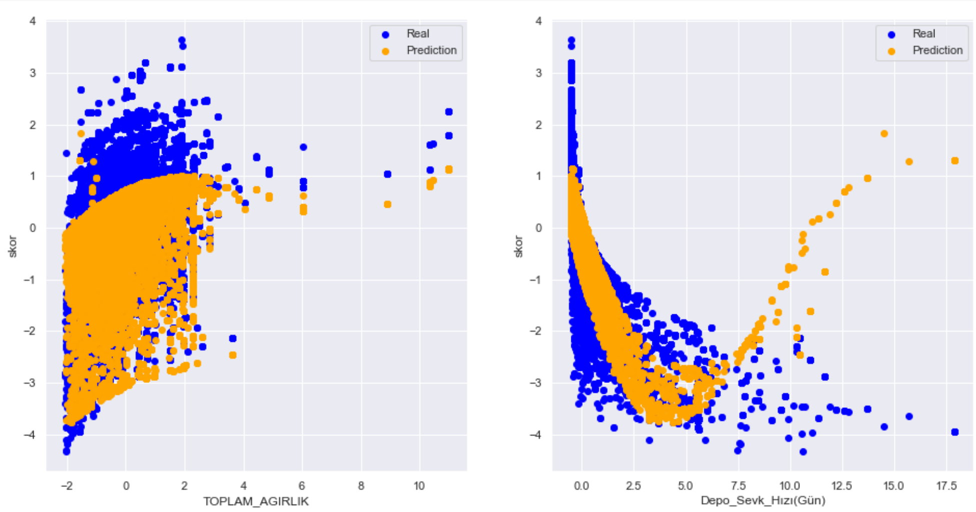


Figure 20 - True Values and Prediction Values

- **Lasso Regression**

```
model_lasso = make_pipeline(PolynomialFeatures(3), Lasso(alpha = 0.01))

model_lasso.fit(X_train, y_train)

y_pred1 = model_lasso.predict(X_test)

model_lasso.score(X_test, y_test)
```

```
0.7942789336675753
```

```
[ ] mean_squared_error(y_test, y_pred)
```

```
0.20203648800394516
```

Figure 21 - Lasso Regression

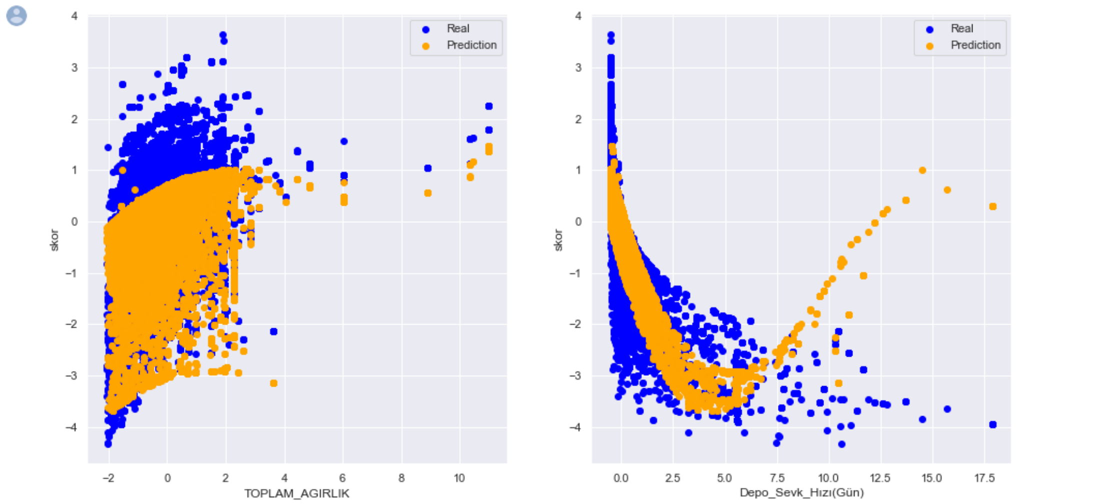


Figure 22 - True Values and Prediction Values

4.1.1.2 LightGBM

A model was created using the LightGBM algorithm alongside linear regression. Cross-validation was performed using KFold and repeated KFold. Then the best parameters were found with a Randomized search.

```
model_latest = lgb.LGBMRegressor(**best_params)

# Evaluate model with test set
model_latest.fit(X_train, y_train)
y_pred = model_latest.predict(X_test)

print(calc_results(y_test, y_pred, "validation", n_sample=0, n_features=0))
```

```
{'experiment': 'validation', 'MSE': 0.1283215692372777, 'MAE': 0.24590768422130285, 'RMSE': 0.3582200011686641, 'R2': 0.8716784307627223}
```

Figure 23 - LightGBM Scores

The R2 score of the LightGBM model is 0.87. It states that this model makes 87 percent accurate predictions, and the fact that our mean square error value is small indicates that the current state of our model is good.

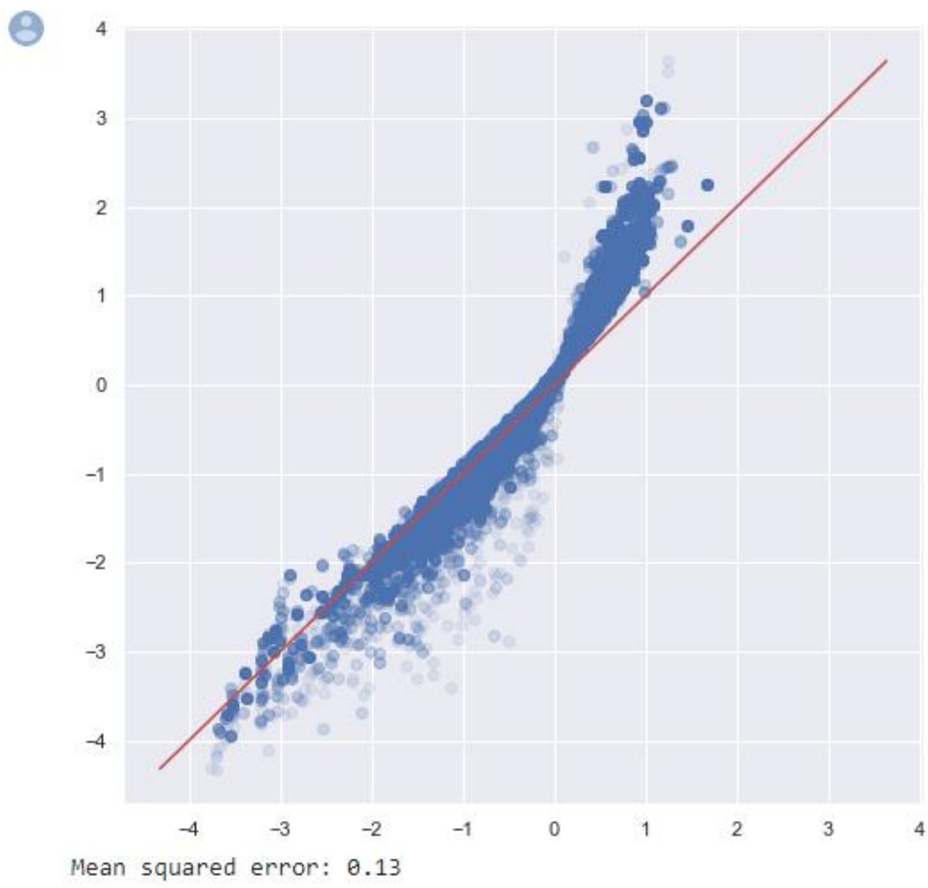


Figure 24 - LightGBM Plot

4.1.2 Score Prediction According to all features

A dataset with all the features of engines was obtained. Numeric values data types were checked, and non-numeric values were converted to category data types and one hot encoding was done. Also, here, ridge regression was used to predict the label.

```
[ ] all_df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 290416 entries, 0 to 324452
Data columns (total 10 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   Güç [kW]            290416 non-null  object
 1   Verim Sınıfı        290416 non-null  object
 2   Gövde               290416 non-null  object
 3   Yapı Şekli          290416 non-null  object
 4   Kutup Sayısı        290416 non-null  object
 5   AGIRLIK             290416 non-null  object
 6   TOPLAM_AGIRLIK      290416 non-null  float64
 7   Depo_Sevk_Hızı(Gün) 290416 non-null  float64
 8   skor1               290416 non-null  float64
 9   skor                290416 non-null  float64
dtypes: float64(4), object(6)
memory usage: 24.4+ MB
```

Figure 25 - Dataset including all features

```
[ ] grid_search.best_params_

{'poly__degree': 2, 'ridge__alpha': 10}

model_ridge = make_pipeline(PolynomialFeatures(1), Ridge(alpha=10))

model_ridge.fit(X_train1, y_train1)

y_pred1 = model_ridge.predict(X_test1)

model_ridge.score(X_test1, y_test1)

0.44764067304327837

[ ] mean_squared_error(y_test1, y_pred1)

0.07484634024730856
```

Figure 26 - Prediction Score

In this model, the score value can be improved. But the error value is fine. This may be related to the size of the test dataset. An underfitting situation may also have occurred. In this model, LightGBM may give better results than in the previous step. But because it was a big process, no result could be obtained.

4.2 Shelf Recommendation

Here, we received the warehouse layout as an excel table (Figure 27) and various information about the warehouse was obtained. Data such as shelf lengths and levels, corridor width, and the location of the shipping door were obtained.

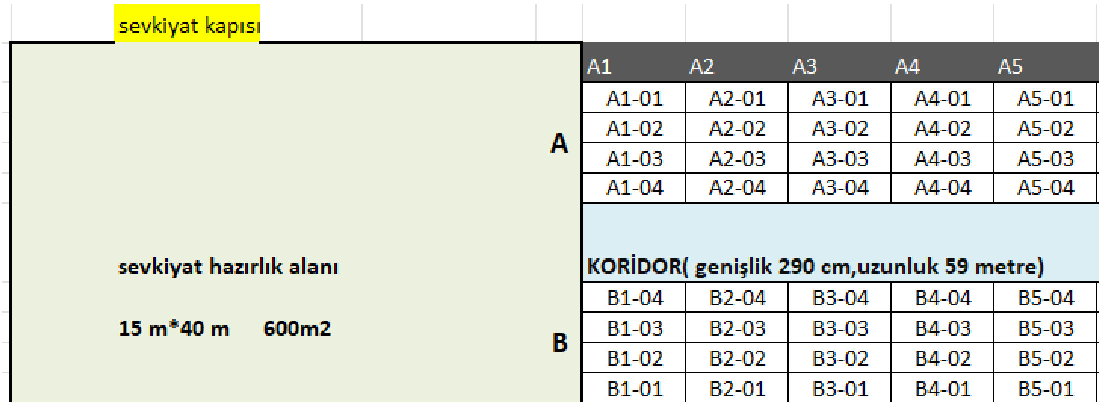


Figure 27 - Warehouse Layout (Appendix A)

In addition, the shelf data on which it was previously placed for each product were also examined. Various graphs and tables were obtained from the exploratory data analysis of this dataset. For example, the number of different types of motors put on each shelf for all the blocks (A, B, C …), the number of different motors placed on the blocks (A1, A2, … B1, B2), the number of motors with a body of 80 & 90 on each shelf.

Mostly, different product placements were made in the C, D, and E blocks. (Appendix D). The product variety is less in areas far from the exit door. I think for the company, the types of motors that will be put on distant shelves are generally clearer.

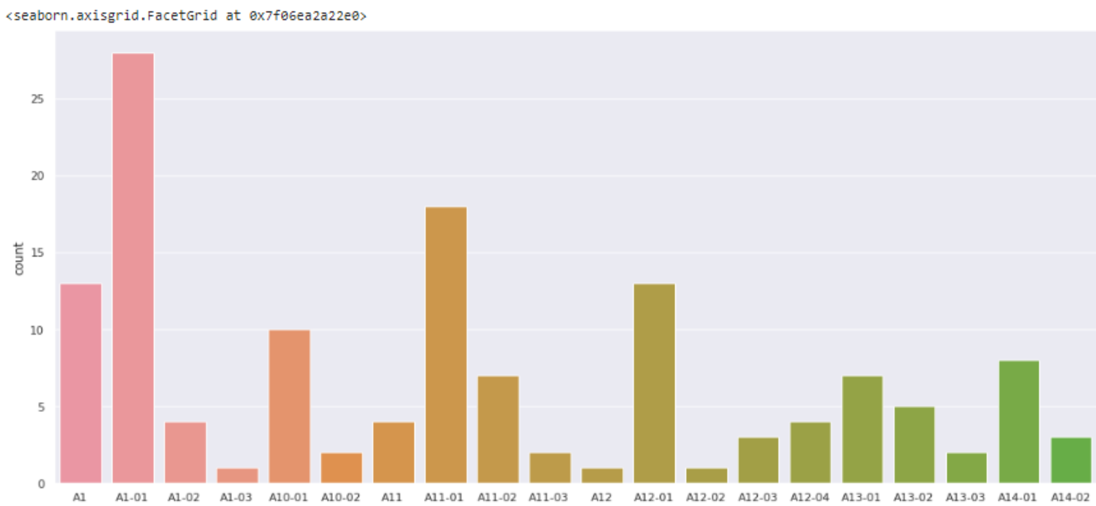


Figure 28 - Number of different types of motors put on each shelf(Appx. C)

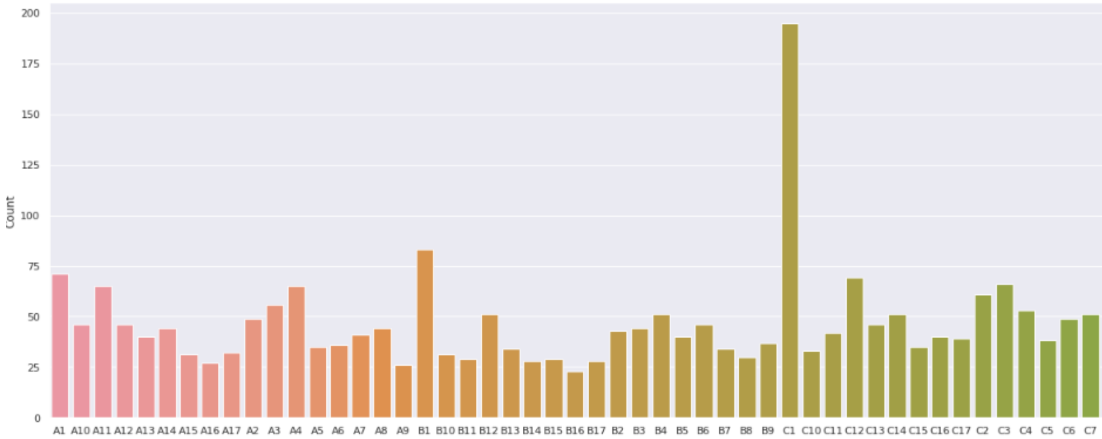


Figure 29 - Number of different motors placed on the blocks (Appx. D)

According to the information received from the company, it has been learned that motors with bodies 80 and 90 have higher priority. Therefore, the following plots have been obtained to see which shelves were most often placed.

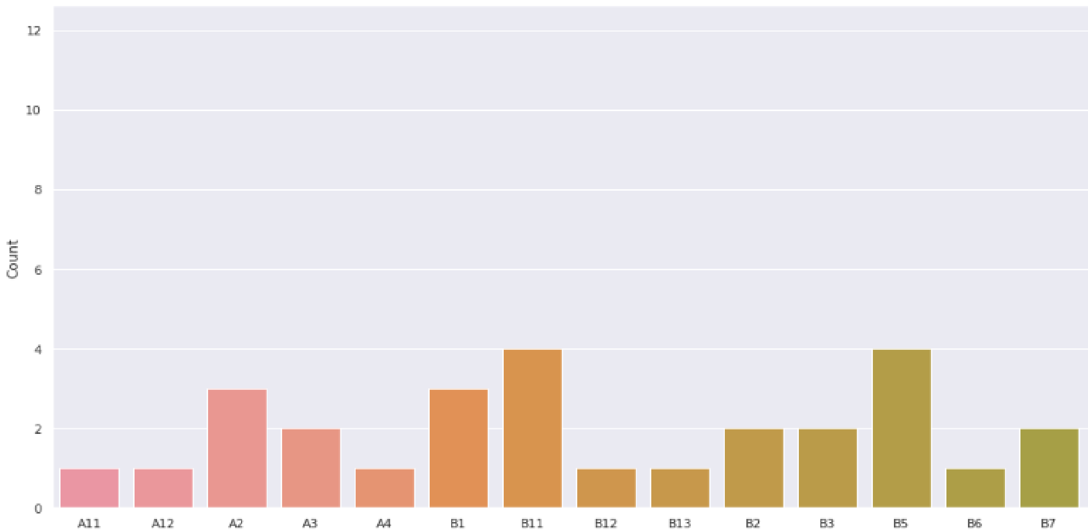


Figure 30 - Number of motors with a body of 80 on each shelf (Appx. E)

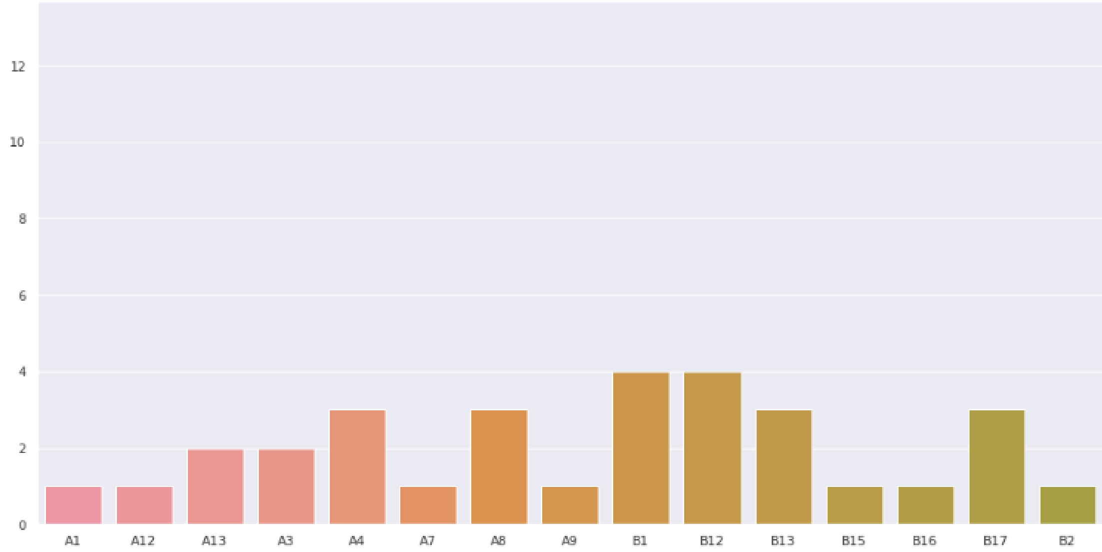


Figure 31 - Number of motors with a body of 90 on each shelf (Appx. F)

Motors with both bodies have a very different distribution of placements. However, those with body 80 are mostly placed in the R and S blocks, while those with body 90 are mostly placed in the X block. When we look at the warehouse layout, it is seen that these blocks are located in an area far from the exit door. Therefore, the previous choices for these two bodies do not seem very advantageous.

Next, the warehouse plan in the form of an excel table was turned into a pandas dataframe in which the numbers in the frame indicate the lengths. After several analyzes, the location of each shelf and their distance from the exit door were calculated. This dataframe merged with the motor dataframe which includes all the features of motors used in the first analysis.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
0	2.8	2.8	4.6	2.0	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	0.9	0.0
1	1.05	0.0	0.0	0.0	0.0	0.0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	0.9	1.05
2	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
3	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
4	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
5	1.05	0.0	0.0	0.0	0.0	0.0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18	B19	0	0	0.0	1.05
6	1.05	0.0	0.0	0.0	0.0	0.0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	0	0	0.0	1.05
7	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.0
8	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
9	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
10	1.05	0.0	0.0	0.0	0.0	0.0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	0	0	0.0	1.05
11	1.05	0.0	0.0	0.0	0.0	0.0	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18	E19	0	0	0.0	1.05
12	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
13	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
14	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
15	1.05	0.0	0.0	0.0	0.0	0.0	F1	F2	F3	F4	F5	F6	F7	F8	F9	1	K1	K2	K3	K4	K5	K6	K7	K8	K9	0	0	0.0	1.05
16	1.05	0.0	0.0	0.0	0.0	0.0	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	0	0	0.0	1.05
17	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
18	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
19	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
20	1.05	0.0	0.0	0.0	0.0	0.0	R1	R2	R3	R4	R5	R6	R7	R8	R9	0	0	1.05	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	0.9	0.9
21	1.05	0.0	0.0	0.0	0.0	0.0	S1	S2	S3	S4	S5	S6	S7	S8	S9	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
22	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
23	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
24	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
25	1.05	0.0	0.0	0.0	0.0	0.0	T1	T2	T3	T4	T5	T6	T7	T8	T9	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
26	1.05	0.0	0.0	0.0	0.0	0.0	U1	U2	U3	U4	U5	U6	U7	U8	U9	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
27	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
28	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
29	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
30	1.05	0.0	0.0	0.0	0.0	0.0	V1	V2	V3	V4	V5	V6	V7	V8	V9	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05

Figure 32 - Warehouse Plan in the form of pandas dataframe

	Shelf_id	Location	distance
0	A1	(1, 6)	12.2
1	A2	(1, 7)	15.0
2	A3	(1, 8)	17.8
3	A4	(1, 9)	20.6
4	A5	(1, 10)	23.4
...	...	...	...
183	X5	(31, 10)	55.0
184	X6	(31, 11)	57.8
185	X7	(31, 12)	60.6
186	X8	(31, 13)	63.4
187	X9	(31, 14)	66.2

188 rows × 3 columns

Figure 33 - Shelf Location and Distances to exit

```
merged.head()
```

	Serial_id	Serial_no	Material_code	material_id	Power [kW]	efficiency	body	shape	motor_pole_count	weight	shipping_speed	Sales_Amount	sales_score	shelf_parent	distance
0	4189197	MP02682307	VSPA90S4AB348MIS	70895	0.55	YOK	90S	AI	4P	11.0	6.400192	1151	12.162832	0	0
1	4189197	MP02682307	VSPA90S4AB348MIS	70895	0.55	YOK	90S	AI	4P	11.0	6.400192	1151	12.162832	X1	43.8
2	4189197	MP02682307	VSPA90S4AB348MIS	70895	0.55	YOK	90S	AI	4P	11.0	6.400192	1151	12.162832	C3	23.1
3	4189197	MP02682307	VSPA90S4AB348MIS	70895	0.55	YOK	90S	AI	4P	11.0	6.400192	1151	12.162832	R1	32.2
4	2028896	MP01487385	VSPA90S4AB348MIS	70895	0.55	YOK	90S	AI	4P	11.0	2.679494	1151	12.162832	0	0

Figure 34 - Merged Dataset

4.2.1 Recommendation System

It is a subfield of machine learning known as recommendation engines, and its primary focus is on assigning ratings to items and consumers. A recommender system is, in a broad sense, a system that estimates how a given user will rate a given product. The user will subsequently be given a ranking of these forecasts.

Google, Instagram, Spotify, Amazon, Reddit, Netflix, and many more often employ them to boost user and platform engagement. To keep you coming back to listen to music on their service, Spotify, for instance, will suggest more songs you might enjoy based on the ones you've already listened to or liked. Based on a user's browsing history and other data, Amazon will make product suggestions. There is a common misconception that recommender systems are a "black box," as the models developed by these major corporations are difficult to decipher. User suggestions for items they need or desire but didn't know they needed until they saw the recommendation top the list of results. [11]

According to the methodology used to determine which items and services will best suit each consumer, most recommendation engines may be grouped into one of three broad groups.

- Recommendation systems adopting collaborative filtering
- Recommendation systems leveraging content-based filtering
- Hybrid recommendation systems [12]

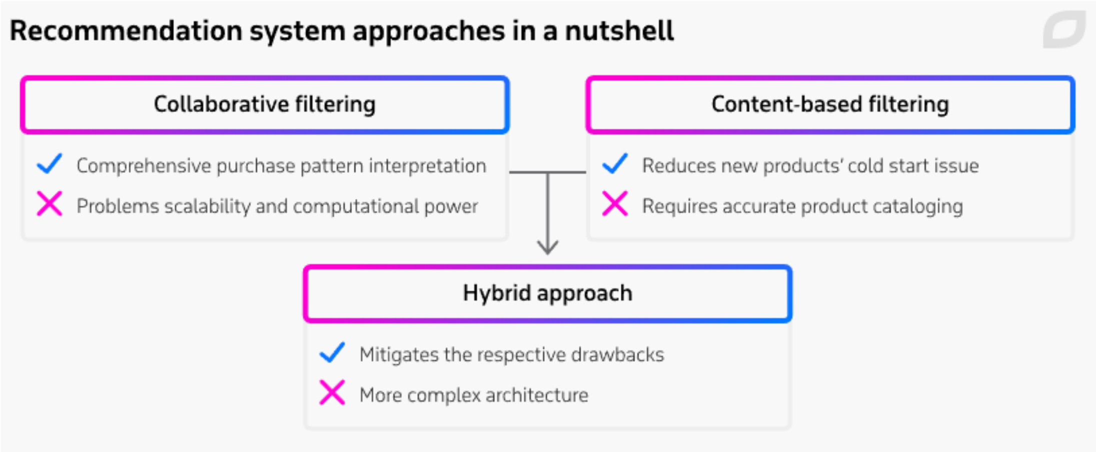


Figure 35 - Recommendation Systems

4.2.1.1 Collaborative Filtering Model based on Sales Score

Sales score values are taken from the first analysis. These values represent the sales priority of the engines and have been obtained by looking at all the features of the engines. Since the sales data will influence the shelf placement, this analysis was done. In this study, python's turicreate package is used for recommendation. Turi Create makes it easier to make custom models for machine learning. Recommendations, object identification, picture classification, image similarity, and activity categorization are all features that may be added to an app without the need for a machine-learning professional. Also, it would be good to explain cosine similarity and pearson correlation as they are used in this study.

Cosine Similarity:

The cosine similarity is a way to figure out how similar two samples are. The two samples can come from the same distribution or two different ones. The number of features on each sample should be the same.

Pearson Correlation:

With the Pearson correlation coefficient, you can figure out how closely two random variables are related. We took n samples from a joint distribution with two variables (X and Y). [13]

It can be seen below the cosine similarity ad pearson correlation of shelves based on the model of sales scores.

```
+-------------------------------+------------+
| Elapsed Time (Item Statistics) | % Complete |
+-------------------------------+------------+
| 1.003ms                       | 68.5       |
| 2.201ms                       | 100        |
+-------------------------------+------------+
Setting up lookup tables.
Processing data in one pass using dense lookup tables.
+-------------------------------------+-------------------+------------------+
| Elapsed Time (Constructing Lookups) | Total % Complete  | Items Processed  |
+-------------------------------------+-------------------+------------------+
| 2.592ms                             | 0                 | 0                |
| 6.489ms                             | 100               | 160              |
+-------------------------------------+-------------------+------------------+
Finalizing lookup tables.
Generating candidate set for working with new users.
Finished training in 0.162596s
recommendations finished on 1000/1483 queries. users per second: 101554
+-------------+--------------+--------------------+------+
| material_id | shelf_parent |       score        | rank |
+-------------+--------------+--------------------+------+
|    70895    |      S1      | 2.260395422577858  |  1   |
|    70895    |      C2      | 2.237390398979187  |  2   |
|    70895    |      X3      | 2.2126710265874863 |  3   |
|    70895    |      C1      | 2.1157464534044266 |  4   |
|    70895    |      X2      | 2.0738603323698044 |  5   |
|    64237    |      X3      | 2.3959529995918274 |  1   |
|    64237    |      X2      | 2.3752209842205048 |  2   |
|    64237    |      C1      | 2.313868761062622  |  3   |
|    64237    |      E5      | 2.1604091972112656 |  4   |
|    64237    |      D2      | 2.0869288742542267 |  5   |
|    37837    |     T11      | 3.3945065566471646 |  1   |
|    37837    |     T12      | 3.2531179445130483 |  2   |
|    37837    |      T6      | 3.100653350353241  |  3   |
|    37837    |      U7      | 2.6061417034694125 |  4   |
|    37837    |      T5      | 2.520461584840502  |  5   |
|    78173    |      S2      | 3.2059036990006766 |  1   |
|    78173    |      S3      | 2.853042592604955  |  2   |
|    78173    |      R4      | 2.6605294744173684 |  3   |
|    78173    |      S5      | 1.8836771547794342 |  4   |
|    78173    |      S8      | 1.623800088961919  |  5   |
|    17369    |      C1      | 1.3445714712142944 |  1   |
|    17369    |      R1      | 1.2286388278007507 |  2   |
|    17369    |      C2      | 1.190946638584137  |  3   |
|    17369    |     D17      | 1.1617159843444824 |  4   |
|    17369    |      C3      | 1.1521409749984741 |  5   |
|    81589    |      B1      | 2.1110139966011046 |  1   |
|    81589    |      R9      | 1.7575656175613403 |  2   |
|    81589    |     R12      | 1.7575656175613403 |  3   |
|    81589    |      D4      | 1.7078068852424622 |  4   |
|    81589    |      B7      | 1.5979076504707337 |  5   |
+-------------+--------------+--------------------+------+
[7415 rows x 4 columns]
```

Figure 36 - Cosine Similarity

```
+--------------------------------+------------+
| Elapsed Time (Item Statistics) | % Complete |
+--------------------------------+------------+
| 612us                          | 68.5       |
| 1.525ms                        | 100        |
+--------------------------------+------------+
Setting up lookup tables.
Processing data in one pass using dense lookup tables.
+-------------------------------------+-------------------+------------------+
| Elapsed Time (Constructing Lookups) | Total % Complete  | Items Processed  |
+-------------------------------------+-------------------+------------------+
| 2.552ms                             | 0                 | 0                |
| 6.931ms                             | 100               | 160              |
+-------------------------------------+-------------------+------------------+
Finalizing lookup tables.
Generating candidate set for working with new users.
Finished training in 0.184231s
recommendations finished on 1000/1483 queries. users per second: 56954.1
+-------------+--------------+--------------------+------+
| material_id | shelf_parent |       score        | rank |
+-------------+--------------+--------------------+------+
|    70895    |     R12      | 13.668741565675356 |  1   |
|    70895    |      R9      | 13.668741565675322 |  2   |
|    70895    |      X8      | 12.750841968851915 |  3   |
|    70895    |     R13      | 12.501327057723373 |  4   |
|    70895    |     X12      | 12.383327794691459 |  5   |
|    64237    |     R12      | 13.668741565675356 |  1   |
|    64237    |      R9      | 13.668741565675322 |  2   |
|    64237    |      X8      | 12.750841968851915 |  3   |
|    64237    |     R13      | 12.501327057723373 |  4   |
|    64237    |     X12      | 12.383327794691459 |  5   |
|    37837    |     R12      | 13.668741565675356 |  1   |
|    37837    |      R9      | 13.668741565675322 |  2   |
|    37837    |      X8      | 12.750841968851915 |  3   |
|    37837    |     R13      | 12.501327057723373 |  4   |
|    37837    |     X12      | 12.383327794691459 |  5   |
|    78173    |     R12      | 13.668741565675356 |  1   |
|    78173    |      R9      | 13.668741565675322 |  2   |
|    78173    |      X8      | 12.750841968851915 |  3   |
|    78173    |     R13      | 12.501327057723373 |  4   |
|    78173    |     X12      | 12.383327794691459 |  5   |
|    17369    |     R12      | 13.668741565675356 |  1   |
|    17369    |      R9      | 13.668741565675322 |  2   |
|    17369    |      X8      | 12.750841968851915 |  3   |
|    17369    |     R13      | 12.501327057723373 |  4   |
|    17369    |     X12      | 12.383327794691459 |  5   |
|    81589    |     R12      | 13.668741565675356 |  1   |
|    81589    |      R9      | 13.668741565675322 |  2   |
|    81589    |      X8      | 12.750841968851915 |  3   |
|    81589    |     R13      | 12.501327057723373 |  4   |
|    81589    |     X12      | 12.383327794691459 |  5   |
+-------------+--------------+--------------------+------+
[7415 rows x 4 columns]
```

Figure 37 - Pearson Correlation

Model evaluation (sales scores) and shelf recommendation tables are below.

PROGRESS: Evaluate model Cosine Similarity on Engine Body
recommendations finished on 1000/1295 queries. users per second:

Precision and recall summary statistics by cutoff

cutoff	mean_precision	mean_recall
1	0.021621621621621623	0.020334620334620333
2	0.010810810810810811	0.020334620334620333
3	0.0077220077220077205	0.021203346203346206
4	0.005984555984555984	0.021975546975546975
5	0.004942084942084942	0.02201064701064701
6	0.004247104247104246	0.02278284778284778
7	0.003640375068946497	0.02278284778284778
8	0.0031853281853281854	0.02278284778284778
9	0.0028314028314028314	0.02278284778284778
10	0.002625482625482626	0.022817947817947822

[10 rows x 3 columns]

Overall RMSE: 9.567751062753672

Figure 38 - Model Evaluation with cosine similarity

PROGRESS: Evaluate model Pearson Similarity on Engine Body
recommendations finished on 1000/1295 queries. users per second:

Precision and recall summary statistics by cutoff

cutoff	mean_precision	mean_recall
1	0.0	0.0
2	0.0	0.0
3	0.0	0.0
4	0.0	0.0
5	0.0	0.0
6	0.0	0.0
7	0.0	0.0
8	0.0	0.0
9	0.0	0.0
10	0.0	0.0

[10 rows x 3 columns]

Overall RMSE: 2.482470299261386

Figure 39 - Model Evaluation with pearson correlation

```
Finalizing lookup tables.
Generating candidate set for working with new users.
Finished training in 0.157891s
recommendations finished on 1000/1483 queries. users per second: 120
+-------------+-------------+--------------------+------+
| material_id | shelf_parent |       score        | rank |
+-------------+-------------+--------------------+------+
|    70895    |     R9      | 13.668741565675859 |  1   |
|    70895    |     R12     | 13.668741565675859 |  2   |
|    70895    |     X8      |  12.7508419688521  |  3   |
|    70895    |     R13     | 12.501327057723095 |  4   |
|    70895    |     X12     | 12.383327794691285 |  5   |
|    64237    |     R9      | 13.668741565675859 |  1   |
|    64237    |     R12     | 13.668741565675859 |  2   |
|    64237    |     X8      |  12.7508419688521  |  3   |
|    64237    |     R13     | 12.501327057723095 |  4   |
|    64237    |     X12     | 12.383327794691285 |  5   |
|    37837    |     R9      | 13.668741565675859 |  1   |
|    37837    |     R12     | 13.668741565675859 |  2   |
|    37837    |     X8      |  12.7508419688521  |  3   |
|    37837    |     R13     | 12.501327057723095 |  4   |
|    37837    |     X12     | 12.383327794691285 |  5   |
|    78173    |     R9      | 13.668741565675859 |  1   |
|    78173    |     R12     | 13.668741565675859 |  2   |
|    78173    |     X8      |  12.7508419688521  |  3   |
|    78173    |     R13     | 12.501327057723095 |  4   |
|    78173    |     X12     | 12.383327794691285 |  5   |
|    17369    |     R9      | 13.668741565675859 |  1   |
|    17369    |     R12     | 13.668741565675859 |  2   |
|    17369    |     X8      |  12.7508419688521  |  3   |
|    17369    |     R13     | 12.501327057723095 |  4   |
|    17369    |     X12     | 12.383327794691285 |  5   |
|    81589    |     R9      | 13.668741565675859 |  1   |
|    81589    |     R12     | 13.668741565675859 |  2   |
|    81589    |     X8      |  12.7508419688521  |  3   |
|    81589    |     R13     | 12.501327057723095 |  4   |
|    81589    |     X12     | 12.383327794691285 |  5   |
+-------------+-------------+--------------------+------+
[7415 rows x 4 columns]
```

Figure 40 - Shelf Recommendation to each engine (based on material id)

4.2.1.2 Collaborative Filtering Model based on Distance to exit

Since the most used products are intended to be placed in the parts close to the exit, distances of shelves were obtained for the priority status. Therefore, this analysis was done according to distance. The same process in the sales score was repeated.

material_id	shelf_parent	score	rank
70895	X3	5.73303297162056	1
70895	X2	5.599899515509605	2
70895	X9	3.8714098781347275	3
70895	X4	3.8377007842063904	4
70895	C2	3.5492898374795914	5
64237	X2	6.185910016298294	1
64237	X3	5.898034021258354	2
64237	E5	4.185516104102135	3
64237	X9	3.8714098781347275	4
64237	X4	3.8377007842063904	5
37837	T6	10.078310855797358	1
37837	U7	7.890514561108181	2
37837	T5	6.994289977209909	3
37837	A4	5.963346183300018	4
37837	D4	5.767942309379578	5
78173	S2	6.743210216363271	1
78173	S3	6.074271728595098	2
78173	R4	5.970969428618749	3
78173	S8	3.5419601500034332	4
78173	S5	3.3739861448605857	5
17369	A1	0.04564210414886474	1
17369	D2	0.0448633861541748	2
17369	D4	0.044785261154174805	3
17369	T8	0.041915500164031984	4
17369	D3	0.04160026788711548	5
81589	B1	3.52697434425354	1
81589	D4	3.1351998329162596	2
81589	B7	2.6849013805389403	3
81589	D2	[illegible]	4

Figure 41 - Cosine Similarity

material_id	shelf_parent	score	rank
70895	E17	67.6	1
70895	D17	66.5	2
70895	X9	66.20000000000014	3
70895	K5	66.20000000000002	4
70895	V9	65.09999999999916	5
64237	E17	67.6	1
64237	D17	66.5	2
64237	X9	66.20000000000014	3
64237	K5	66.20000000000002	4
64237	V9	65.09999999999916	5
37837	E17	67.6	1
37837	D17	66.5	2
37837	X9	66.20000000000014	3
37837	K5	66.20000000000002	4
37837	V9	65.09999999999916	5
78173	E17	67.6	1
78173	D17	66.5	2
78173	X9	66.20000000000014	3
78173	K5	66.20000000000002	4
78173	V9	65.09999999999916	5
17369	E17	67.6	1
17369	D17	66.5	2
17369	X9	66.20000000000014	3
17369	K5	66.20000000000002	4
17369	V9	65.09999999999916	5
81589	E17	67.6	1
81589	D17	66.5	2
81589	X9	66.20000000000014	3
81589	K5	[illegible]	4

Figure 42 - Pearson Correlation

Precision and recall summary statistics by cutoff

cutoff	mean_precision	mean_recall
1	0.0015479876160990704	0.0008091753447790601
2	0.0011609907120743027	0.0010671732807955722
3	0.0007739938080495358	0.0010671732807955722
4	0.0007739938080495358	0.0011961722488038277
5	0.0007739938080495358	0.0012313537855333525
6	0.0006449948400412796	0.0012313537855333525
7	0.0005528527200353823	0.0012313537855333525
8	0.0005804953560371516	0.0014248522375457362
9	0.0006019951840385278	0.0021988460455952706
10	0.0006191950464396284	0.0023923444976076554

[10 rows x 3 columns]

Figure 43 - Model Evaluation with cosine similarity

Precision and recall summary statistics by cutoff

cutoff	mean_precision	mean_recall
1	0.0	0.0
2	0.0	0.0
3	0.0	0.0
4	0.0	0.0
5	0.0	0.0
6	0.0	0.0
7	0.0	0.0
8	0.0	0.0
9	0.0	0.0
10	7.739938080495355e-05	3.5181536729524324e-05

[10 rows x 3 columns]

Figure 44 - Model Evaluation with pearson correlation

material_id	shelf_parent	score	rank
70895	R9	13.668741565675859	1
70895	R12	13.668741565675859	2
70895	X8	12.7508419688521	3
70895	R13	12.501327057723095	4
70895	X12	12.383327794691285	5
64237	R9	13.668741565675859	1
64237	R12	13.668741565675859	2
64237	X8	12.7508419688521	3
64237	R13	12.501327057723095	4
64237	X12	12.383327794691285	5
37837	R9	13.668741565675859	1
37837	R12	13.668741565675859	2
37837	X8	12.7508419688521	3
37837	R13	12.501327057723095	4
37837	X12	12.383327794691285	5
78173	R9	13.668741565675859	1
78173	R12	13.668741565675859	2
78173	X8	12.7508419688521	3
78173	R13	12.501327057723095	4
78173	X12	12.383327794691285	5
17369	R9	13.668741565675859	1
17369	R12	13.668741565675859	2
17369	X8	12.7508419688521	3
17369	R13	12.501327057723095	4
17369	X12	12.383327794691285	5
81589	R9	13.668741565675859	1
81589	R12	13.668741565675859	2
81589	X8	12.7508419688521	3

Figure 45 - Shelf Recommendation to each engine (based on material id)

5. CONCLUSIONS AND FUTURE WORK

This paper presents the combination of scoring (prioritization) of products based on all features of individual product categories (such as sales, weight, etc.) and a recommendation system for product category space allocation in retail stores.

The data gathered from the day-to-day operations of retail businesses are not likely to be sufficient for modeling given the low frequency with which store layouts are changed. I created two indicators (scoring and distance) to evaluate data representativeness for each product type. The research found that these metrics were not the best indicators of the model's predictive performance. Because these metrics such as scoring, and distance are not sufficient to establish this system alone. This research is a simple approach to solving the shelf space allocation problem using machine learning, and it can be further improved by including different metrics. In this study, many metrics such as the height of the shelves, their width, the amount of load they can carry, and the occupancy rate of the shelves have been ignored. In addition to these, the enlargement of the present system with recommendations of which categories should be placed next to each other or depending on what kinds of characteristics is significant. Considering the importance of gathering sufficient information, obtaining this information through a methodical data collection plan is a very important goal for further studies. However, even if this information is obtained, another problem arises here. The problem is figuring out how to combine the different types of metrics in the recommendation system.

Only a very tiny portion of the issue of the shelf space allocation problem has been addressed during this research. Although not at the best stage yet, several of the created models in this study had good results and hence the recommendation system was built.

6. REFERENCES

[1] Tynong, A. (2021). Machine learning for planning in warehouse management (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-178108

[2] Puneet, Sharma, S., Deepika, D., & Singh, G. (2021). Intelligent Warehouse Stocking Using Machine Learning. 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC). https://doi.org/10.1109/icmnwc52512.2021.9688530

[3] Ghaouta, A., Riad, M., & Okar, C. (2021). Machine Learning for Warehouse Management: A conceptual framework. 2021 Third International Conference on Transportation and Smart Technologies (TST). https://doi.org/10.1109/tst52996.2021.00012

[4] Keung, K. L., Lee, C. K., & Yiu, Y. H. (2021). A machine learning predictive model for shipment delay and demand forecasting for warehouses and sales data. 2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). https://doi.org/10.1109/ieem50564.2021.9672946

[5] Zunic, E., Hasic, H., Hodzic, K., Delalic, S., & Besirevic, A. (2018). Predictive analysis-based approach for optimal warehouse product positioning. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). https://doi.org/10.23919/mipro.2018.8400174

[6] Czerniachowska, K., &; Hernes, M. (2021, February 13). A heuristic approach to shelf space allocation decision support including facings, capping, and nesting. MDPI. Retrieved December 6, 2022, from https://doi.org/10.3390/sym13020314

[7] A FICO™ Custom Decision Optimization application. Retail Space Planning and Optimization. http://optiware.co.kr/data/retail_space_planning_optimization_2335ms.pdf

[8] Czerniachowska, K., Sachpazidu-Wójcicka, K., Sulikowski, P., Hernes, M., & Rot, A. (2021). Genetic Algorithm for the Retailers' Shelf Space Allocation Profit Maximization Problem. Applied Sciences, 11(14), 6401. https://doi.org/10.3390/app11146401

[9] Ishichi, K., Ohmori, S., Ueda, M., & Yoshimoto, K. (2019). Shelf-space Allocation Model with Demand Learning. Operations and Supply Chain Management: An International Journal, 24–40. https://doi.org/10.31387/oscm0360219

[10] Zunic, E., Hasic, H., Hodzic, K., Delalic, S., & Besirevic, A. (2018). Predictive analysis-based approach for optimal warehouse product positioning. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). https://doi.org/10.23919/mipro.2018.8400174

[11] Vatsal (2021). Recommendation Systems Explained. https://towardsdatascience.com/recommendation-systems-explained-a42fc60591ed

[12] Andrea Di Stefano (2022). Recommendation systems and machine learning: driving personalization. https://www.itransition.com/machine-learning/recommendation-systems

[13] Lei Mao (2021). Cosine Similarity VS Pearson Correlation Coefficient https://leimao.github.io/blog/Cosine-Similarity-VS-Pearson-Correlation-Coefficient/

[14] Hübner, A. H., & Kuhn, H. (2012). Retail category management: State-of-the-art review of quantitative research and software applications in assortment and shelf space management. Omega, 40(2), 199–209. https://doi.org/10.1016/j.omega.2011.05.008

[15] Chen, M., & Lin, C. (2007). A data mining approach to product assortment and shelf space allocation. Expert Systems with Applications, 32(4), 976–986. https://doi.org/10.1016/j.eswa.2006.02.001

[16] Martello, S., & Toth, P. (1990). Knapsack Problems. John Wiley & Sons Ltd.

[17] Coskun, M. E., 2012. Shelf space allocation: A critical review and a model with price changes and adjustable shelf heights. Master Thesis.

[18] Hansen, P., Heinsbroek, H., 1979. Product selection and space allocation in supermarkets. European Journal of Operational Research 3 (6), 474 – 484.

[19] Corstjens, M., Doyle, P., 1981. A model for optimizing retail space allocations. Management Sciences 27 (7), 822–833.

[20] Zufryden, F. S., 1986. A dynamic programming approach for product selection and supermarket shelf-space allocation. Journal of the Operational Research Society 37 (4), 413–422.

[21] Peters, B. A., Klutke, G.-A., & Botsali, A. R. (2004). Research Issues in Retail Facility Layout Design Progress in Material Handling Research: 2004 (pp. 399-414).

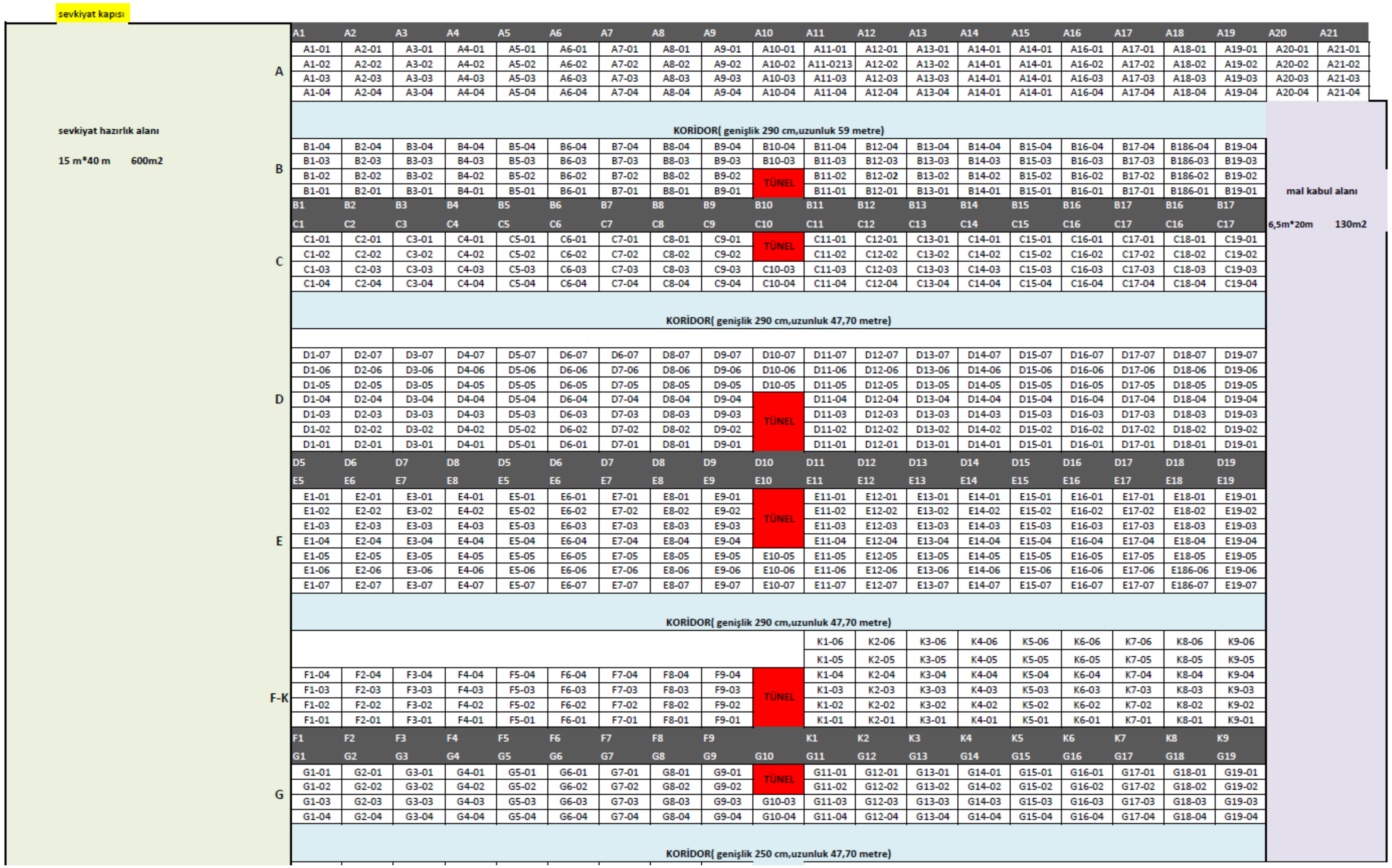
[22] Yapicioglu, H., & Smith, A. E. (2012a). Retail Space Design Considering Revenue and Adjacencies Using a Racetrack Aisle Network. IIE Transactions, 44(6), 446-458.

[23] Yapicioglu, H., & Smith, A. E. (2012b). A Bi-Objective Model for the Retail Spatial Design Problem. Engineering Optimization, 44(3), 243-266.

[24] Pinto, Fábio, and Carlos Soares. "Space Allocation in the Retail Industry: A Decision Support System Integrating Evolutionary Algorithms and Regression Models." Advanced Information Systems Engineering, Springer Berlin Heidelberg, 2013, pp. 531–46. Crossref, https://doi.org/10.1007/978-3-642-40994-3_34

7. APPENDIXES

A. Warehouse Plan (top):



Warehouse Plan (bottom):

KORİDOR(genişlik 290 cm,uzunluk 47,70 metre)

											K1-06	K2-06	K3-06	K4-06	K5-06	K6-06	K7-06	K8-06	K9-06
											K1-05	K2-05	K3-05	K4-05	K5-05	K6-05	K7-05	K8-05	K9-05
	F1-04	F2-04	F3-04	F4-04	F5-04	F6-04	F7-04	F8-04	F9-04		K1-04	K2-04	K3-04	K4-04	K5-04	K6-04	K7-04	K8-04	K9-04
F-K	F1-03	F2-03	F3-03	F4-03	F5-03	F6-03	F7-03	F8-03	F9-03	TÜNEL	K1-03	K2-03	K3-03	K4-03	K5-03	K6-03	K7-03	K8-03	K9-03
	F1-02	F2-02	F3-02	F4-02	F5-02	F6-02	F7-02	F8-02	F9-02		K1-02	K2-02	K3-02	K4-02	K5-02	K6-02	K7-02	K8-02	K9-02
	F1-01	F2-01	F3-01	F4-01	F5-01	F6-01	F7-01	F8-01	F9-01		K1-01	K2-01	K3-01	K4-01	K5-01	K6-01	K7-01	K8-01	K9-01
	F1	F2	F3	F4	F5	F6	F7	F8	F9		K1	K2	K3	K4	K5	K6	K7	K8	K9
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19
	G1-01	G2-01	G3-01	G4-01	G5-01	G6-01	G7-01	G8-01	G9-01	TÜNEL	G11-01	G12-01	G13-01	G14-01	G15-01	G16-01	G17-01	G18-01	G19-01
	G1-02	G2-02	G3-02	G4-02	G5-02	G6-02	G7-02	G8-02	G9-02		G11-02	G12-02	G13-02	G14-02	G15-02	G16-02	G17-02	G18-02	G19-02
G	G1-03	G2-03	G3-03	G4-03	G5-03	G6-03	G7-03	G8-03	G9-03	G10-03	G11-03	G12-03	G13-03	G14-03	G15-03	G16-03	G17-03	G18-03	G19-03
	G1-04	G2-04	G3-04	G4-04	G5-04	G6-04	G7-04	G8-04	G9-04	G10-04	G11-04	G12-04	G13-04	G14-04	G15-04	G16-04	G17-04	G18-04	G19-04

KORİDOR(genişlik 250 cm,uzunluk 47,70 metre)

	R1-04	R2-04	R3-04	R4-04	R5-04	R6-04	R7-04	R8-04	R9-04
R	R1-03	R2-03	R3-03	R4-03	R5-03	R6-03	R7-03	R8-03	R9-03
	R1-02	R2-02	R3-02	R4-02	R5-02	R6-02	R7-02	R8-02	R9-02
	R1-01	R2-01	R3-01	R4-01	R5-01	R6-01	R7-01	R8-01	R9-01
	R1	R2	R3	R4	R5	R6	R7	R8	R9
	S1	S2	S3	S4	S5	S6	S7	S8	S9
	S1-01	S2-01	S3-01	S4-01	S5-01	S6-01	S7-01	S8-01	S9-01
S	S1-02	S2-02	S3-02	S4-02	S5-02	S6-02	S7-02	S8-02	S9-02
	S1-03	S2-03	S3-03	S4-03	S5-03	S6-03	S7-03	S8-03	S9-03
	S1-04	S2-04	S3-04	S4-04	S5-04	S6-04	S7-04	S8-04	S9-04

KORİDOR(genişlik 295 cm,uzunluk 25,30metre)

	T1-04	T2-04	T3-04	T4-04	T5-04	T6-04	T7-04	T8-04	T9-04
T	T1-03	T2-03	T3-03	T4-03	T5-03	T6-03	T7-03	T8-03	T9-03
	T1-02	T2-02	T3-02	T4-02	T5-02	T6-02	T7-02	T8-02	T9-02
	T1-01	T2-01	T3-01	T4-01	T5-01	T6-01	T7-01	T8-01	T9-01
	T1	T2	T3	T4	T5	T6	T7	T8	T9
	U1	U2	U3	U4	U5	U6	U7	U8	U9
	U1-01	U2-01	U3-01	U4-01	U5-01	U6-01	U7-01	U8-01	U9-01
U	U1-02	U2-02	U3-02	U4-02	U5-02	U6-02	U7-02	U8-02	U9-02
	U1-03	U2-03	U3-03	U4-03	U5-03	U6-03	U7-03	U8-03	U9-03
	U1-04	U2-04	U3-04	U4-04	U5-04	U6-04	U7-04	U8-04	U9-04

KORİDOR(genişlik 295 cm,uzunluk 25,30metre)

	V1-04	V2-04	V3-04	V4-04	V5-04	V6-04	V7-04	V8-04	V9-04
V	V1-03	V2-03	V3-03	V4-03	V5-03	V6-03	V7-03	V8-03	V9-03
	V1-02	V2-02	V3-02	V4-02	V5-02	V6-02	V7-02	V8-02	V9-02
	V1-01	V2-01	V3-01	V4-01	V5-01	V6-01	V7-01	V8-01	V9-01
	V1	V2	V3	V4	V5	V6	V7	V8	V9
	X1	X2	X3	X4	X5	X6	X7	X8	X9
	X1-01	X2-01	X3-01	X4-01	X5-01	X6-01	X7-01	X8-01	X9-01
X	X1-02	X2-02	X3-02	X4-02	X5-02	X6-02	X7-02	X8-02	X9-02
	X1-03	X2-03	X3-03	X4-03	X5-03	X6-03	X7-03	X8-03	X9-03
	X1-04	X2-04	X3-04	X4-04	X5-04	X6-04	X7-04	X8-04	X9-04

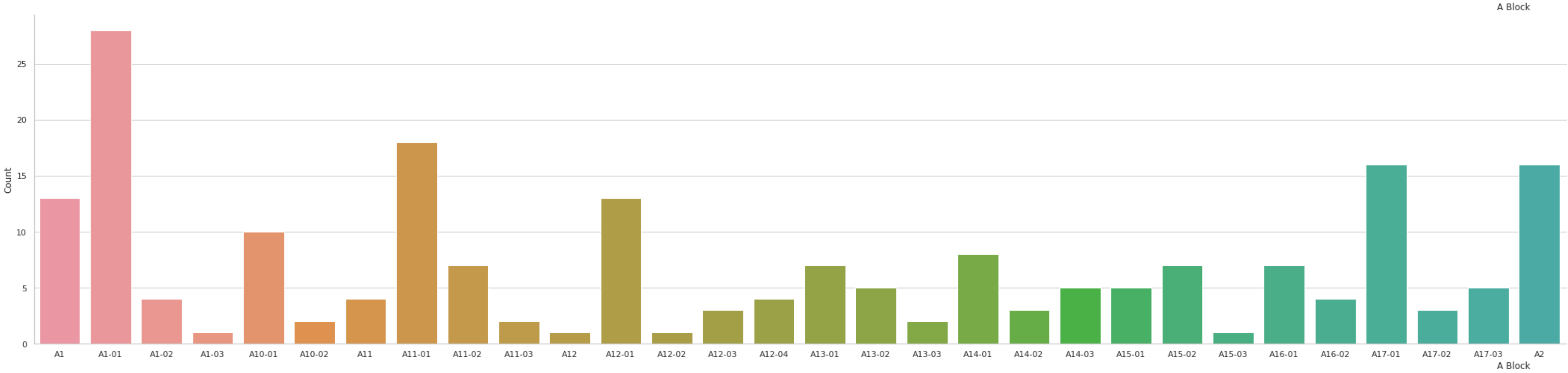
KORİDOR(genişlik 310 cm,uzunluk 25,30metre)

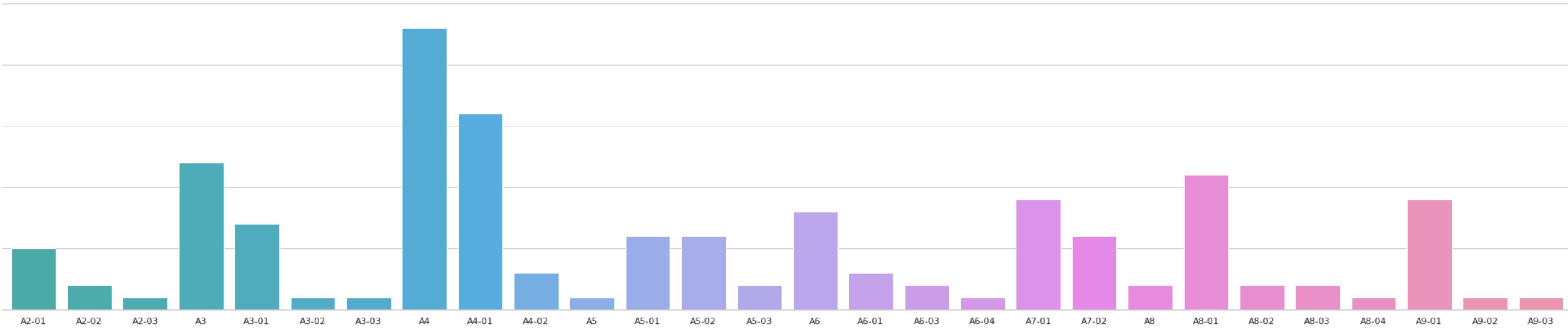
KORİDOR(genişlik 450 cm,uzunluk 20 metre)

B. Warehouse plan in the form of pandas dataframe

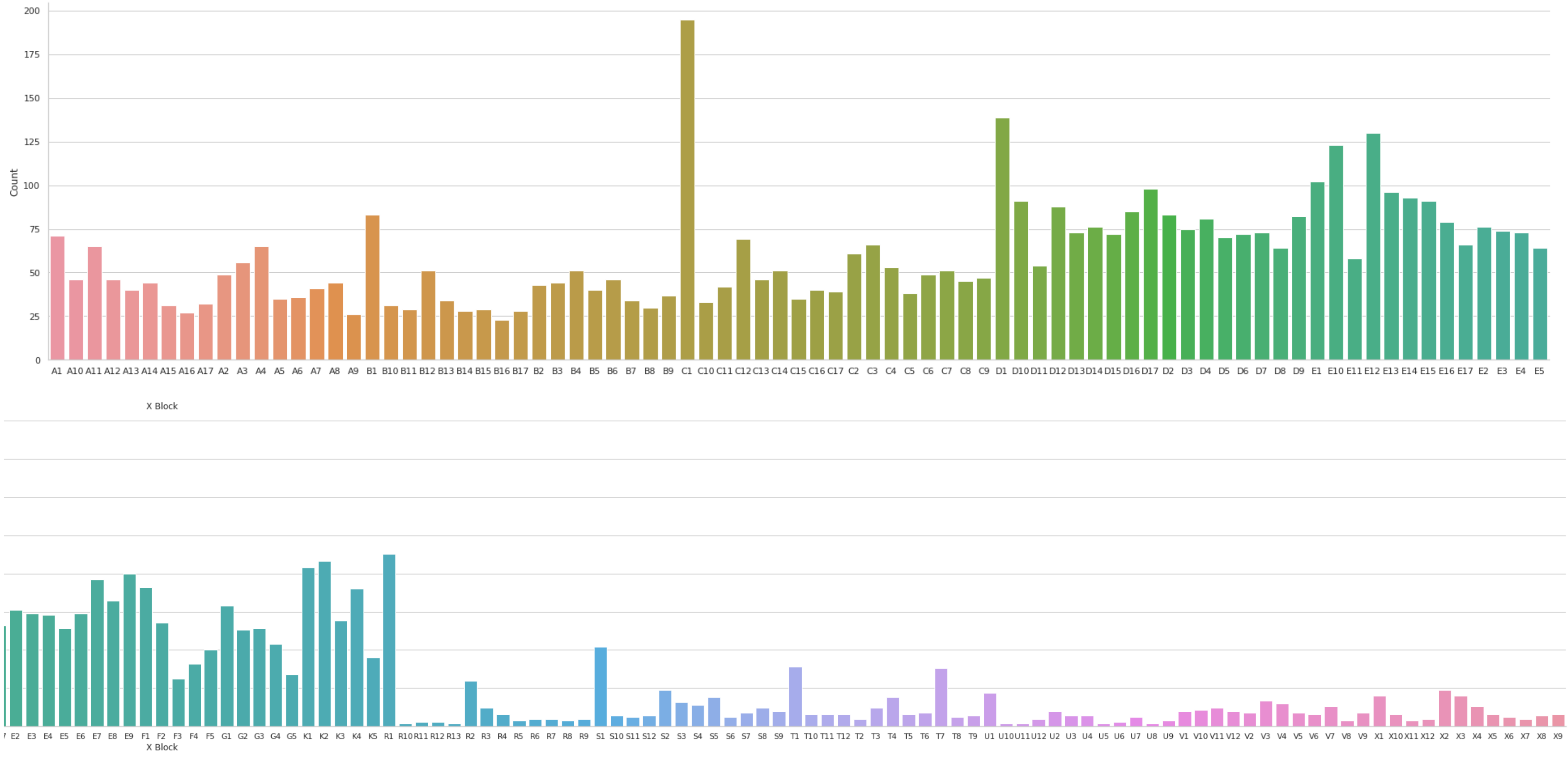
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
0	2.8	2.8	4.6	2.0	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	0.9	0.0
1	1.05	0.0	0.0	0.0	0.0	0.0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	0.9	1.05
2	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
3	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
4	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
5	1.05	0.0	0.0	0.0	0.0	0.0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18	B19	0	0	0.0	1.05
6	1.05	0.0	0.0	0.0	0.0	0.0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	0	0	0.0	1.05
7	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.0
8	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
9	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
10	1.05	0.0	0.0	0.0	0.0	0.0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	0	0	0.0	1.05
11	1.05	0.0	0.0	0.0	0.0	0.0	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18	E19	0	0	0.0	1.05
12	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
13	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
14	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
15	1.05	0.0	0.0	0.0	0.0	0.0	F1	F2	F3	F4	F5	F6	F7	F8	F9	1	K1	K2	K3	K4	K5	K6	K7	K8	K9	0	0	0.0	1.05
16	1.05	0.0	0.0	0.0	0.0	0.0	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	0	0	0.0	1.05
17	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
18	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
19	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	1.05
20	1.05	0.0	0.0	0.0	0.0	0.0	R1	R2	R3	R4	R5	R6	R7	R8	R9	0	0	1.05	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	0.9	0.9
21	1.05	0.0	0.0	0.0	0.0	0.0	S1	S2	S3	S4	S5	S6	S7	S8	S9	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
22	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
23	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
24	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
25	1.05	0.0	0.0	0.0	0.0	0.0	T1	T2	T3	T4	T5	T6	T7	T8	T9	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
26	1.05	0.0	0.0	0.0	0.0	0.0	U1	U2	U3	U4	U5	U6	U7	U8	U9	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
27	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
28	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
29	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
30	1.05	0.0	0.0	0.0	0.0	0.0	V1	V2	V3	V4	V5	V6	V7	V8	V9	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
31	1.05	0.0	0.0	0.0	0.0	0.0	X1	X2	X3	X4	X5	X6	X7	X8	X9	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
32	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
33	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
34	1.05	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0	0	0	0	0	0	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
35	1.5	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	0	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05

C. Number of different types of motors put in each shelf

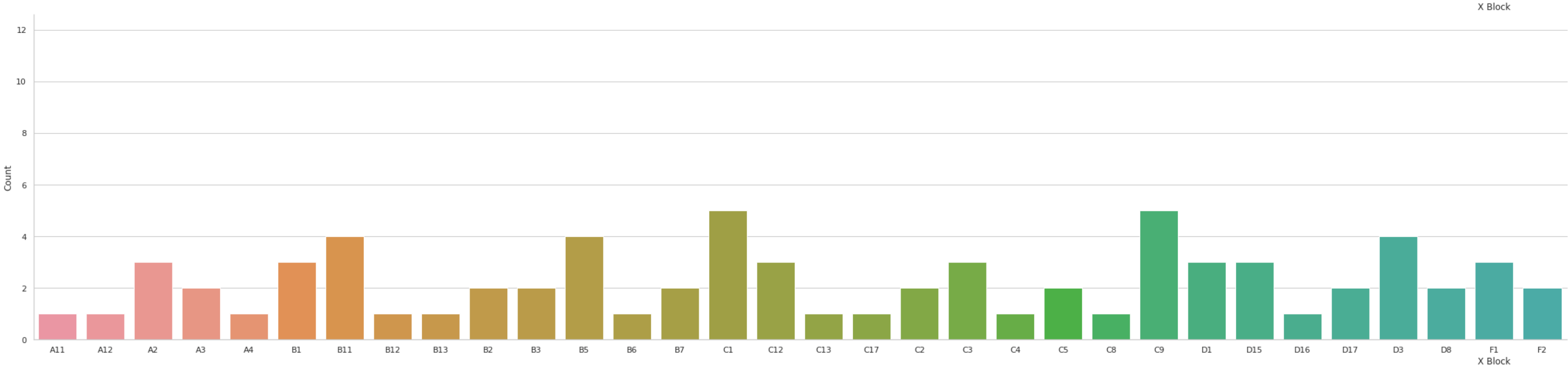


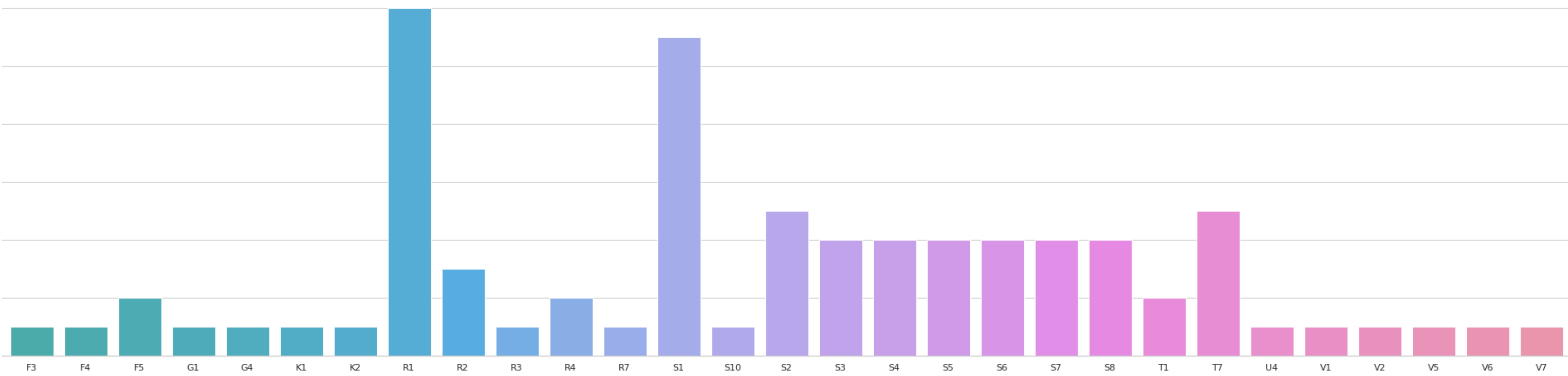


D. Number of different motors placed on the blocks



E. Number of motors with a body of 80 on each shelf





F. Number of motors with a body of 90 on each shelf