POLITECNICO DI TORINO

MASTER OF SCIENCE
IN
AUTOMOTIVE ENGINEERING

Master's degree thesis

PERFORMANCE EVALUATION OF AN INNOVATIVE TRANSPORT SYSTEM FOR AUTOMATED WAREHOUSES

Supervisors
Prof. Franco Lombardi
Prof. Giulia Bruno

Laureandi
Mahilan Ravichandran
S250203

Academic Year 2019 – 2020

Abstract

In this Digital era, from using Autonomous robots in warehouses, to use of drones in the near future for order fulfillment, the Supply chain industry is undergoing a remarkable change, yet it struggles to meet the evolving customer expectations. Customers or Businesses nowadays both of them demand rapid delivery at no extra cost, which is a growing pressure put on the logistic industry. This is where the Intralogistics should make a greater modification on Automated storage and retrieval systems to retrieve maximum productivity and meet customer expectation up to the mark. Currently one of the highly successful type of warehousing systems are the shuttle-based storage and retrieval systems, even though they perform extremely well in all terms compared to traditional warehouses, there are limitations of this system which this paper seeks to address and propose a solution. With the aim of taking the Logistic industry to the next level, this work tests the performance of an Innovative design of shuttle-based storage and retrieval system proposed by an Italian automatic handling system producing company. This paper evaluates the new system performances using various tools and techniques and compares it with the traditional shuttle based automated storage and retrieval systems.

Keywords – Industry 4.0, Shuttle based automated storage and retrieval system, Automated picking, Rainbow pallet, Innovative order picking system.

Acknowledgement

I would like to express my gratitude and appreciation for my thesis supervisors Prof. Franco Lombardi and Prof. Giulia Bruno of the Department of Management and Production Engineering (DIGEP) of the Politecnico di Torino, whose guidance, support and encouragement has been invaluable throughout this study.

I would also like to sincerely thank Alberto Faveto who continuously provided encouragement and was always willing and enthusiastic to assist in any way he could throughout the research project.

To conclude, I cannot forget to thank my family and friends, for all the unconditional love and support in this very intense academic year. A special thanks to Miss. Karthika Srikanthithasan, you were always there to support me with a word of encouragement or listening ear.

Table of Contents

List of Code Snippets

List of Figures

List of Tables

1. Introduction

Generally, a warehouse is a planned environment for storage and handling of goods and materials. Warehousing dates back to the Ancient Roman times during the 2nd century BC, primitively constructed to store food grain, clothing, wine, marble, and oil which were used during the time of famine. Later in the 18th and 19th century the construction and spread of rail transport necessitated the erection of warehouses to store goods for transportation and distribution. In the 20th century the machine operated factories were established everywhere and the production rate of goods were steeply increased which led to the construction of plenty of warehouses all over the world and to use more mechanized methods for product storage and retrieval.

The estimated order-picking operations in a traditional warehouse can account for roughly 65% of the total operating cost, and 60% of all labor activities [1], as a result of this, the warehouses in the 21st century comprehended automation, which was a major leap in technology of warehouses. Despite first automated warehouses being tested during the late 20th century, the 21st century showed major advancements in the technologies of Automated storage and retrieval systems. The current automated storage and retrieval system (AS/RS) differ according to the type of technology used, which can include cranes, shuttles, vertical lift modules (VLMs), carousels, micro-loads, mini-loads, unit-loads, or other systems. In this Thesis we focus on the Shuttle based storage and retrieval system (SBS/RS).

SBS/RS (Shuttle based storage and retrieval system) are storage systems that perform the storage and retrieval operations of the products, usually pallets or boxes in rack structures, automatically through machines specifically developed for this purpose. [2] These machines are called storage/retrieval machines and they involve conveyors, stacker cranes, shuttle vehicles, satellite vehicles which store the goods on rack structures. The general layout of the SBS/RS is shown in Figure 1. Nevertheless, these systems require immense capital and initial investment and high maintenance cost so these systems must be planned and designed economically. Compared with traditional automated storage and retrieval system (AS/RS), SBS/RS could achieve higher warehouses volume and throughput capacity, as a result are capable of coping with constantly changing volumes [3].

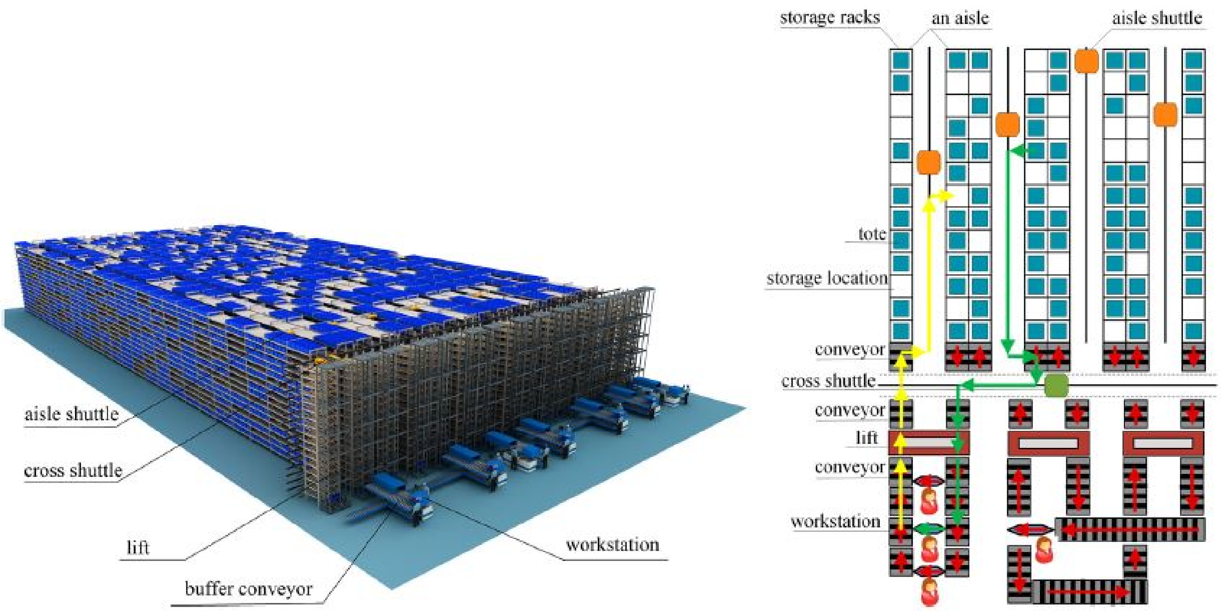


Figure 1 General layout of an SBS/RS (Shuttle based storage and retrieval system)

Source - [4]

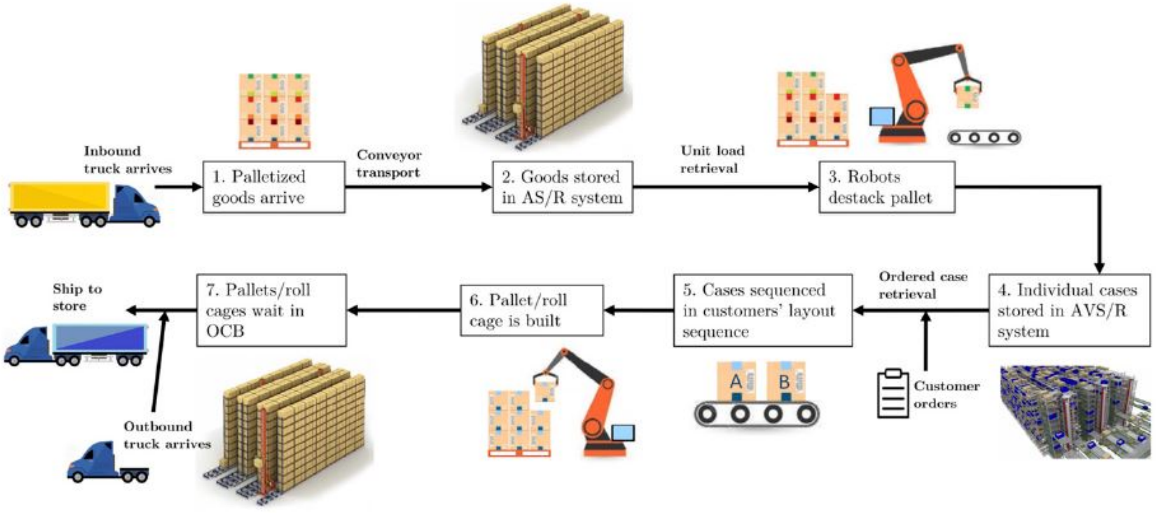

Inbound truck arrives
1. Palletized goods arrive
Conveyor transport
2. Goods stored in AS/R system
Unit load retrieval
3. Robots destack pallet
4. Individual cases stored in AVS/R system
Ordered case retrieval
Customer orders
5. Cases sequenced in customers' layout sequence
A
B
6. Pallet/roll cage is built
7. Pallets/roll cages wait in OCB
Ship to store
Outbound truck arrives

Figure 2 Material flow in a typical automated storage warehouse

Source - [5]

1.1. Motivation and Objective of the thesis

The scope of this thesis is to compare the performance of an innovative intralogistics system proposed by an Italian automatic handling system producing company Eurofork S.p.A with the traditional shuttle-based warehouse technology used. The company has been producing automatic handling equipment for over 20 years and today its one of the leading players on the global market. One of the main divisions of the company is the production of ESMARTSHUTTLE systems for multi depth automated warehouses. Esmartshuttle is very similar to a modern-day shuttle based automated storage and retrieval system. A new study proposed by this company is to develop a system that builds mixed or rainbow pallets according to the present trend of mixed orders by the customer. A mixed pallet is usually built by retrieval of the whole pallet to the workstation and picking only the selected number of products and returning it back to the storage location. Figure 2 presents the material flow of a typical automated storage warehouse. This new proposed system is expected to steeply decreases the time taken for the whole operation by building a mixed pallet within the storage rack using robotic picking technology integrated within the shuttle in an SBS/RS (Shuttle based storage and retrieval system). The Figure 3 and Figure 4 shows proposed system integrated with the picking technology.

This new technology aims to increase the system throughput, insertion and extraction times of goods and energy consumption, since it eliminates the need to retrieve the whole pallet to the picking station to build a mixed pallet. Since the system in Eurofork is still in the early development phase no physical testing can be done, so the performance of this system is studied considering a virtual environment taking the data of an UK-based and registered non-store online retail. The performance of the new proposed system and the traditional shuttle-based storage system are tested and compared to give a better understanding on its performances, before actual investment and development by the company. This work is more focused on cycle time of an order completion rather than other parameters such as energy consumption, throughput, etc.

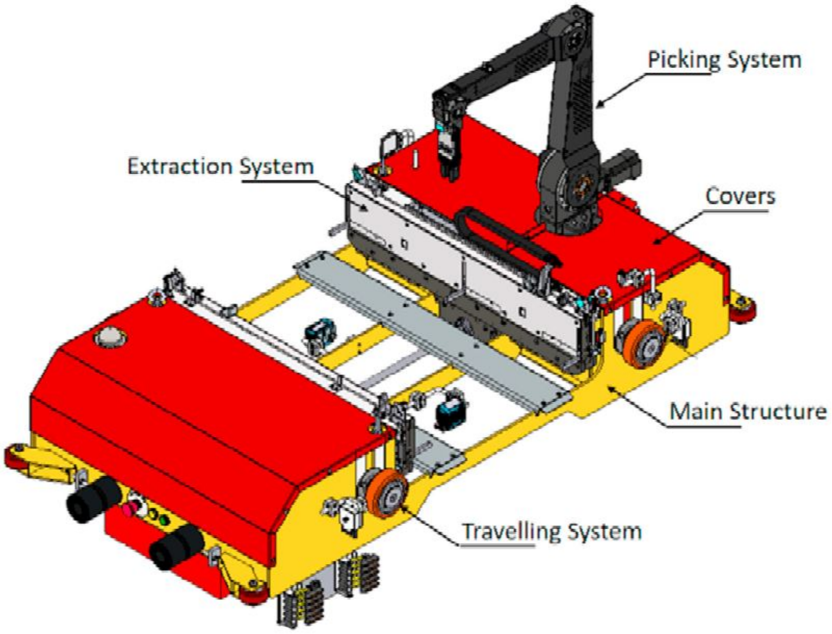


Figure 3 SBS/RS with integrated picking system

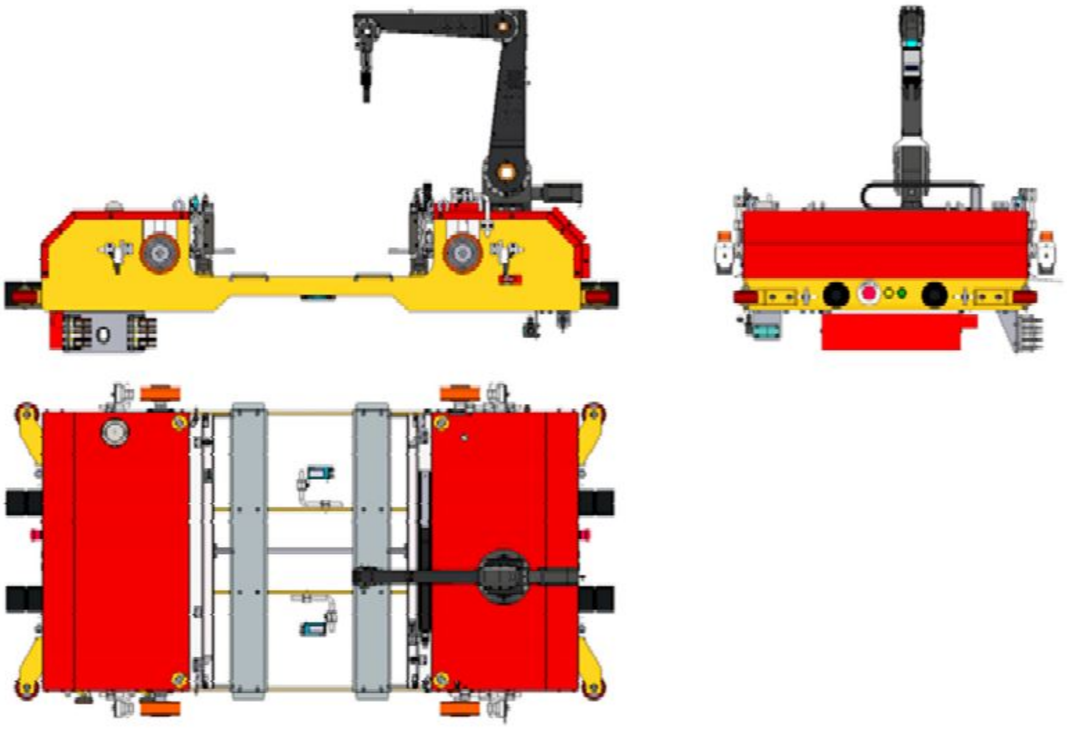


Figure 4 Various views of SBS/RS with integrated picking system

Source - [2]

1.2. Thesis outline

The thesis is organized as follows,

- Chapter 1 : It presents the introduction of the Automated storage and retrieval systems and the Objective of the thesis work.
- Chapter 2 : It presents the Past literature on various topics related to Lean management, Industry 4.0, Automated storage and retrieval systems and their types, also a detailed view of data mining techniques and their classification are discussed. This chapter concludes with the problem discussion.
- Chapter 3 & 4 : It presents the various tools utilized to carry out the data analysis and the pre-processing techniques used on the dataset to carry out the data cleansing. Moreover in this section the visualization of the dataset it done to better understand the data.
- Chapter 5 : It presents the data mining techniques used to extract the association rules from the dataset using the Apriori algorithm.
- Chapter 6 : It presents various simulation techniques and simulation software to build scenarios and evaluate the performance parameters to compare two different systems.
- Chapter 7 : It presents the conclusion and future aspects of this work.

2. Literature review

2.1 Lean management of warehouse

M. Bevilacqua, F. E. Ciarapica and S. Antomarioni [6] presented the paper on lean principles for organizing items in an automated storage and retrieval system: an association rule mining – based approach. The paper started off by presenting how warehouse management plays an important role from the organizational point of view and inefficiencies in the field may reflect on the production processes. It says, reducing the time dedicated to non-adding value tasks during the picking process represents a waste reduction. It also adds up the point that travel from the picking point is the most time-consuming task during the picking process, followed by search and the pick-up of the components.

The paper presents that the application of the 5S methodology to warehouse management represents an important step for all manufacturing companies, especially for managing products that consist of a large number of components. Also, it explains the five-step characterizing the 5S methodology proposed by [7]. The five steps are

- Seiri: sorting all the items in the workplace and eliminating the unnecessary ones.
- Seiton: place the items in the optimal location.
- Seiso: regularly clean the workplace.
- Seiketsu: standardize the process.
- Shitsuke: sustain the process, ensuring that the previous activities are regularly executed.

The paper also says that 5S methodology is not only suitable for improving production processes conditions, but also for space management issues like those faced in warehouse management. Then the authors introduce the topics related to Data Mining and states that the application of Data Mining (DM) techniques surely supports the interpretation of the data provided by production processes and supply chain operations in order to increase company's efficiency. Also, integrating the opportunities by DM techniques and the desire of efficiency growth, this research is directed to material management in AS/RS, which is a procedure for organizing the items in an AS/RS based on a well-known DM technique, namely the Association Rule Mining (ARM), is developed. Specifically, the historical data of the order picking processes are analyzed to determine the item categories frequently required together. Then, based on the confidence of the rules mined, on the area required by each category and on the dimensions of the AS/RS, items categories are assigned to the most convenient shelf of the storage system. In this way, the space and material management at the basis of the 5S philosophy is pursued.

The brief literature review presented by the author stated that the Automated Storage and Retrieval Systems were widely used in manufacturing environment for storing raw material, semi-finished or finished products, and retrieving them when an order is required. The three different objectives pursued by the storage assignment policies, were dedicate to,

(a) improve the operating efficiency,

(b) minimize storage costs and

(c) minimize picking distances.

The approaches as originally identified by other authors are namely

(a) The random storage assignment,

(b) The closest open location storage assignment,

(c) Full-turnover-based or class-based storage assignment.

The paper stated that in order to minimize picking distance and, thus, to reduce the time dedicated to the picking activity, [8] has proposed to store the most frequently picked items close to the picking point. In a similar perspective, [9] developed an algorithm for the picking of a number of order simultaneously, in order to achieve better productivity: the approach rely on the association rule mining and considers the support as the key performance indicator to define whether two items should be picked together.

The research approach of this paper aims at developing a procedure to organize items in AS/RS based on the Association Rule Mining and according to the space management aspect pursued by 5S methodology. Specifically, the reduction of the number and duration of picking processes is addressed. Starting with the concept of Association Rule Mining, the authors define that it is a methodology that can be applied to extract interesting and hidden relations from large datasets with the aim of supporting the decisional processes and the paper also depicts the procedure with a real-life example of a manufacturing company which produces shoes.

To conclude, the paper states that Managing inventory is one of the most vital and yet wasteful tasks in manufacturing. Labor is a large cost of warehouse operations. The more times one item is handled, the higher the costs associated with the item. In this work, a new approach has been proposed, based on Big Data Analytics methods, for reducing waste time during storage and retrieval activities of components and finished products. The authors specify the aim of the proposed algorithm to allow minimizing picking time when operators must prepare an assembly kit. In particular, the approach can be used during the implementation of 5S methodology since it helps assigning items to the shelf, removing items that are no longer needed (sort), organizing the items to optimize efficiency and flow (straighten), and developing behaviors that keep the workplace organized over the long term(sustain). In conclusion the author depicts that both manufacturing automation and lean manufacturing have the same goals that is to satisfy customers at the lowest possible cost.

2.2 Automated Storage and Retrieval Systems (AS/RS)

AS/RSs are used to store and retrieve loads in various settings and operate under computerized control, maintaining an inventory of stored items. The primary components of an Automated storage and Retrieval system are conveyors, cranes rack structure, aisles, input/output points, and picking area. Racks are described as metal structures with locations that can accommodate loads (e.g., pallets) that need to be stored. Cranes are the fully automated storage and retrieval machines that can autonomously pick up, move, and drop off loads. Aisles are the empty spaces between the racks, where the cranes can move. An input/output point (I/O point) is a location where retrieved loads are dropped off, and where incoming loads are picked up for storage. Picking positions are workspace where people are working to remove individual items from a retrieved load before the load is sent back into the system. [10]

Retrieval of items is fulfilled by indicating the type of item and quantity to be retrieved and the system determines where in the storage area the item can be retrieved from and schedules the retrieval. The system then commands the automated storage and retrieval machine to the location where the item is stored and directs the machine to retrieve it and deliver the item at a location where it is to be picked up.

Predominantly, only part of the unit-load may be needed to fulfill a customer's order. The author suggested that this can be resolved by having a separate picking area in the warehouse, in which the AS/RS serves to replenish the picking area, or the picking operation can be integrated with the AS/RS. An alternate solution proposed by the author was to design the crane such that a person can ride along (person-onboard). Instead of retrieving a full pallet automatically from the location, the person can pick one item from the location. A more common option to integrate item picking is when the AS/RS drops off the retrieved unit loads at a workstation. A picker at this workstation takes the required number of products from the unit-load after which the AS/RS moves the remainder of the load back into the storage rack. This system is often referred to as an end-of-aisle system. [10]. The classification of various types of AS/RS is presented in Figure 5.

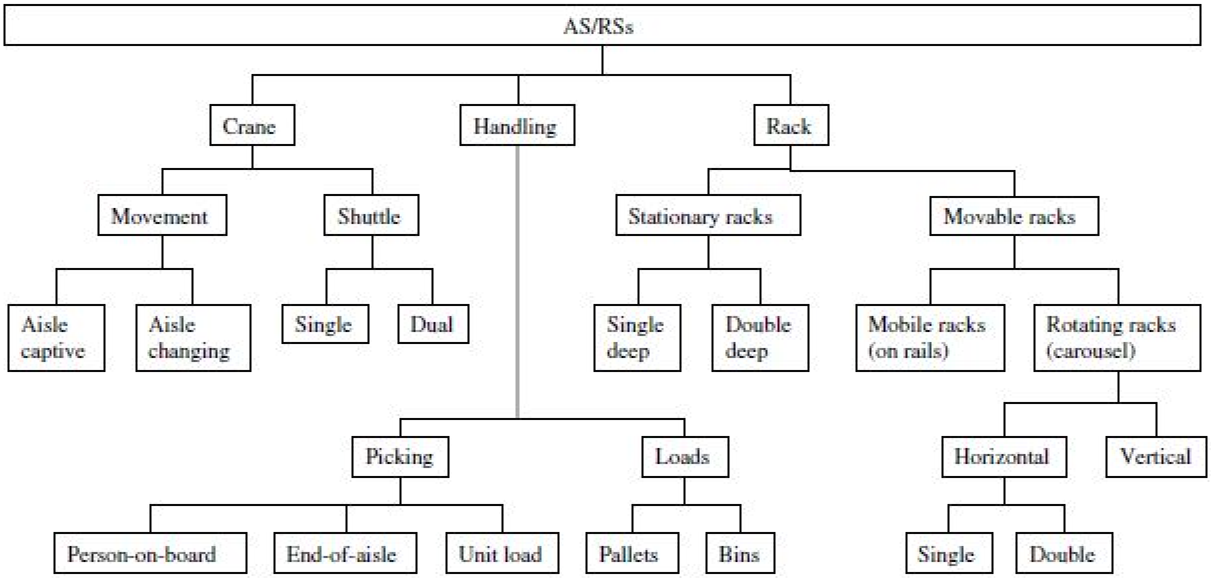


Figure 5 Classification of various AS/RS system options.

Source- [10]

2.2.1 Benefits of an AS/RS system

A recent study by [11] recognized that, automating the low-value and easily repeated task of inventory storage and retrieval, AS/RS brings many powerful benefits to the operations that utilize it [11], including,

- More efficient use of floor space
- Ability to reclaim unused vertical space
- Increased inventory storage density
- Improved ergonomics and safety, resulting in fewer accidents
- Increased throughput
- Reduced labor costs
- Fewer labor constraints due to labor shortages
- Modular design for maximum flexibility
- Increased order picking accuracy
- Improved product security for premium inventory

2.2.2 Types of Automated Storage & Retrieval Systems (AS/RS)

The two major types of Automated storage and retrieval systems (AS/RS) are Unit-Load AS/RS and Mini-Load AS/RS but between these two main types there are six primary ranges of AS/RS systems, [11]

I. Unit-Load AS/RS Cranes (Fixed-Aisle & Moveable Aisle)

II. Mini-Load AS/RS Cranes

III. Shuttle- and Bot-based AS/RS

IV. Carousel-based AS/RS (Vertical, Horizontal, and Robotic)

V. Vertical Lift Module (VLM) AS/RS

VI. Micro-Load (Stocker)

We explore each of these models in more detail below.

i. Unit-Load AS/RS

Unit-load AS/RS systems are typically used to handle especially large and bulky loads weighing more than five hundred kilos. Generally, a unit-load AS/RS consists of narrow aisle racks, which can extend to heights larger than one hundred feet and which hold pallets of product and inventory. A crane is assigned to the rack, which is used to place and retrieve pallets when commanded. Unit-load AS/RS is especially preferrable option when pallet-level storage is limited, and quick retrieval is critical.

The two types within the unit-load AS/RS are fixed-aisle and moveable aisle cranes.

- Fixed-Aisle Unit-Load AS/RS Crane (Tier captive)

In fixed-aisle unit-load AS/RS systems, pallet racks are arranged with narrow aisles between them. A crane travels between these aisles moving both horizontally and vertically to retrieve and store products. The crane is fixed to a single aisle. This setup is also known as Tier captive systems.

- Moveable-Aisle Unit Load AS/RS Crane (Tier to Tier)

The working of the Moveable-aisle unit load AS/RS is exactly similar to the fixed-aisle unit-load AS/RS. The key difference is that it is not fixed to a specific aisle. This allows the crane to serve multiple aisles and covers a greater working space. This setup is also known as Tier to Tier systems.

ii. Mini-Load AS/RS

Mini-load AS/RS generally handles smaller loads compared to unit-load systems. Instead of full pallets, mini load AS/RS deals with totes, trays, cartons, etc. The Mini-load storage systems are peculiarly apt for operations that involves thousands of SKUs and large storage, instead lack the necessary floor space required for traditional shelves to provide a pick face for each store keeping unit. Mini-load AS/RS systems can also be used to buffer and efficiently release/sequence product to picking or palletizing stations and can be used to automatically replenish pick locations like carton-flow.

iii. Shuttle-based AS/RS

Shuttle-based AS/RS delivers inventory via a shuttle or bot that runs on a track between a racking structure. They can operate on a single level or multiple levels, depending on the needs of the operation, and can be battery- or capacitor-powered. The shuttles deliver the tote or carton to a workstation integrated with the system. [11] When an item is requested, the shuttle drives to the location of the product and retrieves the tote or carton that contains the requested item. The shuttle then takes the tote or carton straight to a workstation or it transfer the tote/carton to a conveyor to convey it to a workstation. Different shuttle systems make use of different designs to provide distinct benefits. For example, some models are vertically oriented to optimize floor space. The shuttles move on the perimeter of the rack and then move into an aisle to extract a tote and delivers it to its integrated workstation. Another shuttle model handles vertical rack, but each bot moves on the floor and ascends vertically in order to extract a tote. After picking the tote it descends and independently delivers the product to an assigned workstation. If the workstation is busy it queues and waits until picked and then returns back to complete a new task and it repeats the process.

iv. AMR-Based High-Density AS/RS

An autonomous mobile robot-based high-density automated storage and retrieval system is designed in a way that uses three-axis AMR robots to travel vertically up storage rack to retrieve the required inventory tote or case. The AMR is built-in with a specific storage spot on itself, which is used to carry the products, then it navigates down the rack and reaches the specified order picking workstations. The AMR drives up the workstation’s ramp, and the integrated pick-to-light and software system commands the item type and quantity to be picked. The operator then places the appropriate item and quantity into one of the batched orders and the AMR leaves for its next assignment.

v. Carousel-based AS/RS

Carousel-based automated storage and retrieval systems comprises of products in bins or inventory which rotate along a track continuously. When the operator requests a particular item, the system will automatically rotate so that the appropriate bin is accessible so that the item can be picked. An integrated lightree notifies the picker which carousel, placed in which shelf, and which item to pick. Carousel-based automated storage and retrieval systems may consist of either a horizontal carousel (horizontal movement of bins) or a vertical carousel (vertically movement of bins). Horizontal carousels are usually used for smaller and lighter items and parts and Vertical carousels for little heavier parts. Robotic horizontal carousel AS/RS is another type of system which gives functionality of a fully automated AS/RS.

In these Robotic horizontal carousels, up to three tiers of carousels are stacked on top of each other, and totes or cases are loaded on each shelf level. All these three carousels work independently to provide the necessary inventory to an inserter/extractor device that runs horizontally in the front. The inserter/extractor takes as many as two totes or cartons per trip to take-away conveyor, which delivers the goods to a workstation, and picks up returning inventory, placing it back in a waiting shelf. It is possible to increase capacity and throughput by increasing the number of carousel rows with an inserter/extractor in front of it.

vi. Vertical Lift Module (VLM)

A vertical lift module (VLM) is a system consisting of an inserter/extractor in the center and a vertical set of trays on each side. It is an enclosed system and a form of goods-to-person technology. When an item is ordered, firstly the inserter/extractor locates the necessary tray, retrieves it, and then delivers it to an operator and the operator completes the order. Once the order is complete, the vertical lift module will return the tray to its original location before retrieving the next tray. The trays can be either fixed or dynamic. In trays in the fixed systems will always be returned to the same location, in the case of dynamic system, the trays stored will vary.

vii. Micro-Load Stocker

A Micro-Load Stocker is used for individual or discrete totes or carton storage and retrieval. It is optimal for sequencing, buffering, and point-of-use items in a high-density storage location. It is an enclosed system and has an inserter/extractor device that operates in the center of the system, picking a specific inventory and then delivering them onto the conveyor or workstation. Different models of this Micro load stocker work differently, by taking either one item at a time or a group of up to five items in one time. This system can also store SKUs until needed, discharging them onto awaiting conveyor. It can be paired with other AS/RS systems to improve the other system’s performance and drastically reduce conveyor and floor space requirements.

2.3 Data mining

Data mining is a process of extracting information and knowledge implicit in a large, incomplete, noisy, vague, random large-scale data. The association rule as the important branch of data mining is a high-level and intelligent data processing and analysis technology, We can use the association rules to mine the relationships among data, and we can get the potential value of the information hidden in the massive data. [9]

Since current manufacturing environment is moving towards a 4.0 perspective, there is a growing focus on big data analytics techniques. Indeed, data understanding represents a key aspect for extracting useful knowledge and new information with the aim of taking advantage from them. Thus, the application of Data Mining techniques surely supports the interpretation of the data provided by production processes and supply chain operations in order to increase company's efficiency. [6]

The various types of data mining techniques as per [12] are discussed below,

1. Classification
2. Clustering
3. Regression
4. Association rules
5. Outer detection
6. Prediction
7. Sequential patterns
8. Decision trees

2.3.1. Classification

This technique is used to obtain important and relevant information about data and metadata. This data mining technique helps to classify data in different classes. [12]

Data mining techniques can be classified by various criteria described by [12] as,

i. Classification of Data mining frameworks as per the type of data sources mined

This classification is based on the type of data handled. For example, multimedia, spatial data, text data, time-series data, World Wide Web, and so on.

ii. Classification of data mining frameworks as per the database involved

This classification is based on the data model involved. For example. Object-oriented database, transactional database, relational database, and so on.

iii. Classification of data mining frameworks as per the kind of knowledge discovered

This classification is based on the data mining functionalities or types of knowledge discovered. For example, classification, clustering, discrimination, characterization, etc. some frameworks can also to be extensive frameworks which offers a few data mining functionalities together.

iv. Classification of data mining frameworks according to data mining techniques used

This type of classification depends on the data analysis approach utilized, such as genetic algorithms, neural networks, visualization, machine learning, statistics, data warehouse-oriented or database-oriented, etc. It can also consider the level of user interaction involved in the data mining procedure, such as query-driven systems, autonomous systems, or interactive exploratory systems.

2.3.2. Clustering

Clustering is a division of information into groups of connected objects, which means that data in the same group are more similar to the others. Dividing the data into one or more clusters may loses some confine details but accomplishes improvement. Data modeling considers clustering from a historical point of view with the roots in mathematics, statistics, and numerical analysis. From a machine learning point of view, clusters relate to hidden patterns, the search for clusters is unsupervised learning, and the subsequent framework represents a data concept. From a technical point of view, clustering plays an important role in data mining applications. For example, exploration of scientific data, text mining, retrieval of information, spatial database applications, CRM, Web analysis, medical diagnostics, and much more. [12]

This clustering technique helps to recognize the differences and similarities between the data and Clustering is very similar to the classification in terms of concepts, but it involves grouping lump of data together based on their similarities.

There are various methods of clustering algorithms used in clustering datasets, a few top used methods are listed below

i. K-means clustering

K-means clustering is a type of unsupervised learning, which is used when unlabeled data are present in the dataset. Unlabeled data are data without defined categories or groups. The goal of the K-means algorithm is to discover groups in the data, with the number of groups defined by the variable K. The algorithm performs iterations to assign each data point to one of group of K's based on the input provided. The clustering of these data points is based on feature similarity. The results obtained from the K-means clustering algorithm are

1. The centroids of the K clusters, which can be used to label new data
2. Labels for the training data (each data point is assigned to a single cluster) [13]

Instead of defining groups by looking at the data, clustering algorithm allows us to find and analyze the groups that have formed naturally. The algorithm discovers the clusters and data set labels for a particular pre-chosen value of K. In order to find the appropriate number of clusters in the data, the user must run the algorithm for an increasing range of K values and compare the results. Generally, there is no method for determining the exact value of K, but an accurate estimation can be obtained using the following techniques.

One of the metrics that is frequently used to compare results across various values of K is the mean distance between data points and their cluster centroid. Increasing the number of clusters always reduce the distance to data points and increasing K will always reduce this metric, to the extreme of reaching zero when K is the same as the number of data points. Thus, we cannot be dependent only on this. Instead, the mean distance to the centroid as a function of K is plotted and the "elbow point," where the rate of decrease sharply shifts, can be used to roughly determine K. [13]

Therefore, this algorithm is composed of the following steps,

1. First step is placing K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Next is assigning each object to the group that has the closest centroid.
3. After all objects have been assigned, recalculating the positions of the K centroids.
4. Repeating Steps 2 and 3 until the centroids no longer move. This generates a separation of the objects into groups from which the metric to be minimized can be calculated. [14]

This algorithm is also undoubtedly sensitive to the initial randomly selected cluster centers, so the k-means algorithm must be run multiple times to reduce this effect.

There are other various techniques available for validating K, which includes cross-validation, the information theoretic jump method, information criteria, the silhouette method, and the G-means algorithm. Besides monitoring the data points distribution across groups provides insights on how the algorithm splits the data for each K. Each centroid of a cluster obtained is a collection of feature values defining the resulting groups. A qualitative interpretation of what kind of group each cluster belongs is examined using the centroid feature weights.

ii. Hierarchical Clustering Algorithm

Hierarchical Clustering Algorithm or Hierarchical cluster analysis is an unsupervised clustering algorithm which involves creating clusters that have predominant ordering from top to bottom. The algorithm groups similar objects into groups called clusters. The result of this algorithm is a set of clusters, where each cluster is discrete from each other cluster, and the values within each cluster are predominantly similar to each other.

This clustering technique is divided into two different types namely,

- Agglomerative Hierarchical Clustering
- Divisive Hierarchical Clustering

- **Agglomerative hierarchical clustering**

The Agglomerative Hierarchical Clustering is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It is also known as AGNES - Agglomerative Nesting. It is a "bottom-up" approach and each observation initiates in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. [15]

Working of this algorithm is based on these steps,

1. First step is to make each data point a single-point cluster → forms N clusters
2. Secondly taking the two closest data points and blending them to form one cluster → forms N-1 clusters
3. Again, taking the two closest clusters and making them one cluster → Forms N-2 clusters.
4. Repeating the third step until we are left with only one main cluster.

There are a lot ways to measure the distance between clusters for deciding the rules for clustering, and these are often called Linkage Methods. Some of the frequently used linkage methods mentioned in [15] are,

Complete linkage: The distance between two clusters is defined as the longest distance between two points in each cluster.

Single linkage: The distance between two clusters is defined as the shortest distance between two points in each cluster. This linkage may be used to detect high values in a dataset which may be outliers and they will be merged at the end.

Average linkage: The distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.

Centroid linkage: This linkage discovers the centroid of cluster 1 and centroid of cluster 2, and then evaluates the distance between the two clusters before merging.

The choice of linkage method entirely depends on our convenience and there is no universal choice that provides best results. However, different linkage methods lead to different clusters.

The ultimate aim of performing these linkages is to demonstrate the way hierarchical clustering works, this maintains a memory of how process was achieved. This memory is stored in Dendrogram. A Dendrogram is a kind of tree diagram which shows hierarchical relationships between different sets of data. This dendrogram contains the memory of hierarchical clustering algorithm, so just by looking at the Dendrogram we can conclude how the cluster was formed. A sample dendrogram with the parts marked is shown in Figure 6.

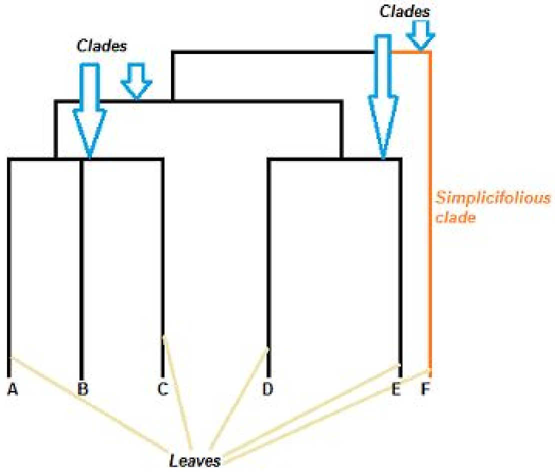


Figure 6 Parts of a dendrogram

Source - [16]

A dendrogram can be a column graph or a row graph. Some dendrograms are circular or have a fluid-shape, but the python library usually produces a row or column graph. Despite the shape, the main graph comprises the same parts.

- Clades are the branches which are arranged according to how similar or dissimilar they are. Clades that are almost the same height are mostly similar. clades with different heights are dissimilar. So, the greater the difference in height, the more dissimilarity is expected.
- Each clade comprises of one or more leaves.
- In the Figure 6 Leaves A, B, and C are more similar to each other than they are to leaves D, E, or F.
- The Leaves D and E are more similar to each other than they are to leaves A, B, C, or F.
- It is observed that the Leaf F is considerably different from all of the other leaves.
- A clade can theoretically have infinite amount of leaves but more the leaves, the harder it is to read the graph with the naked eye.

- **Divisive hierarchical clustering**

In Divisive or DIANA - Divisive Analysis Clustering is a top-down clustering method where we assign all of the observations to a single cluster and then partition the cluster to two least similar clusters. Then we progress recursively on individual clusters until there is one cluster for each observation. This clustering approach is exactly opposite to Agglomerative clustering. There is evidence that divisive algorithms produce more accurate hierarchies than agglomerative algorithms in some circumstances but is conceptually more complex. In both agglomerative and divisive hierarchical clustering, users need to specify the desired number of clusters as a termination condition (which defines the number at which merging should be stopped). [15]

2.3.3. Regression

Regression analysis is one of the data mining process used to identify and analyze the relationship between variables because of the presence of the other factor. It is used to define the probability of the specific variable. Regression, primarily a form of planning and modeling. For example, we might use it to project certain costs, depending on other factors such as availability, consumer demand, and competition. Primarily it gives the exact relationship between two or more variables in the given data set. [12]

2.3.4. Association rules

The association rule refers to the rule of the potential relation among the high frequency data items in the database. This data mining technique helps to discover a link between two or more items. It finds a hidden pattern in the data set. Association rules are if-then statements that support to show the probability of interactions between data items within large data sets in different types of databases. Association rule mining has several applications and is commonly used to help sales correlations in data or medical data sets. Because of the application of the association rules in the real field, the resources have been utilized effectively, and the service quality of the domain is relatively improved. The most standard algorithm of association rule is the Apriori algorithm, and it is an algorithm with great influence in data mining technology. Its core is a recursive algorithm based on two stage frequency sets. Primitively, all the frequency sets should be found, and then the association rules are generated by the frequency sets. These rules must satisfy the minimum support and the minimum confidence. Once the rules generated are less than the minimum support and the minimum confidence, they will be deleted. And then it will search and scan the remaining frequency sets. [9] [12]

In simple terms the way the algorithm works is that you have various data, for example, a list of grocery items that you have been buying for the last six months. It calculates a percentage of items being purchased together.

These are three major measurements technique used are Support, Confidence and Lift. The authors [17] well defines these terms as given below,

i. Support

Support is the proportion of transactions containing a particular combination of items, say A and B, relative to the total number of transactions N. Mathematically, support (A → B) is given by,

$$\textbf{Support(A} \rightarrow \textbf{B) = frequency (A, B)/N = P(A} \cap \textbf{B)}$$

For a single item termed A, the support is given by

$$\textbf{Support(A) = frequency (A)/N = P(A)}$$

ii. Confidence

Confidence measures how much the consequent item is dependent on the antecedent item, i.e. conditional probability of the consequent item given the antecedent item.

$$\textbf{Confidence (A} \rightarrow \textbf{B) = frequency (A, B)/ frequency (A) = P(A} \cap \textbf{B)/P(A)}$$

A minimum amount of support and confidence is necessary for the co-occurrence to be termed as an association rule. The minimum support value and minimum confidence value come from past experiences and change over time, season, region, culture, the standard of living, demographics, etc. A higher confidence depicts stronger association rules.

iii. Lift

Support and confidence by itself cannot determine accurately the strength of an association rule because of the high degree of randomness seen in the purchased items. Lift, which is also termed as improvement or impact, is a measure to overcome the problems with support and confidence.

$$\mathbf{Lift(A \rightarrow B) = P(A \cap B)/[P(A) * P(B)]}$$

The lift tells us how much better a rule is at predicting the result than just assuming the result in the first place. Greater lift values indicate stronger associations. [17]

2.3.5. Outer detection

Outer detection is a data mining technique that relates to the observation of data items in the data set, which do not match an expected pattern or expected behavior. This technique may be used in various domains like intrusion, detection, fraud detection, etc. This technique is also known as Outlier Analysis or Outlier mining. The outlier is a data point that diverges too much from the rest of the dataset. A greater part of the real-world datasets has an outlier, and this detection plays a significant role in the data mining field. Outlier detection is valuable in numerous fields like network interruption identification, credit, or debit card fraud detection, detecting outlying in wireless sensor network data, etc.

2.3.6. Sequential patterns

The sequential pattern is a data mining technique specially designed for evaluating sequential data to discover sequential patterns. It consists of steps to find interesting subsequences in a set of sequences, where the stake of a sequence is measured in terms of different criteria like length, weight, occurrence frequency, etc. Especially, this technique of data mining helps to discover or recognize similar patterns in a transaction data over some time.

2.3.7. Prediction

Prediction uses a combination of other data mining techniques such as trends, clustering, classification, etc. It analyzes past events or instances in the right sequence to predict a future event. [12]

2.4. Warehouse management based on Association rules

The Paper [9] starts off by stating how warehouse management is important for the development of enterprises and how a good warehouse management system can enable enterprises to operate solid foundation. This paper focuses on applying the Apriori algorithm to the warehouse management system to analyze the records of the amount of goods in the warehouse, and to obtain the association rules between the goods. From the point of views of the authors, this method proposed can help procurement staff save time, and reduce the influence of the shortage of goods for sales.

The paper states that the content of warehouse management is important to the decision makers and managers of the enterprises, and it is an integral part of the enterprises and the people have been using the method of traditionally manual management to manage warehouse goods information, but this management method cannot provide users with adequate information and quick means of inquiry, and there have been still some problems like low efficiency, poor confidentiality, frequent updating of data and difficult maintenance in the method of manual management. The Author's point is that a well-designed warehouse management system can reduce the cost that is spent on warehouse management and reduce the burden on warehouse managers and the rapid data analysis can provide timely information for enterprises.

So, the solution proposes was to implement Data Mining in the warehouse management, where Data mining is a process of extracting information and knowledge implicit in a large, incomplete, noisy, vague, random large-scale data. The research says that the association rule is the important branch of data mining and it is a high-level and intelligent data processing and analysis technology, which has been widely applied to various fields.

The paper states that so far, a lot of data mining algorithms have been proposed, in which Agrawal's Apriori algorithm [18] is the most famous one, and most data mining algorithms are based on the Apriori algorithm. The paper defines the term "warehouse" that includes various types of storage warehouses and distribution centers in the field of production and supply and Its main role is statistics and analysis of inventory data, so that decision-makers can early find the problems, take appropriate measures to adjust the inventory structure, shorten the reserve cycle, speed up cash flow, so as to ensure the smooth flow of production.

Particular attention is paid to Data Mining concept and in order to discover the relationships among the different commodities in the supermarket transaction database, the association rule was firstly proposed. It has been found that most typical example is the beer-diaper problem found in the Wal-Mart supermarket: the customers who buy diaper often buy beer and this result is gotten from analyzing many purchase records in supermarket. Concluding that this analysis method is the association rule of data mining.

The research states that the Apriori algorithm is the most used technique and Its core is a recursive algorithm based on two stage frequency sets. Firstly, all the frequency sets were to be found, and then the association rules were to be generated by the frequency sets then these rules must satisfy the minimum support and the minimum confidence and Once the rules generated are less than the minimum support and the minimum confidence, they will be deleted and then it will search and scan the remaining frequency sets.

Talking about the Algorithm and methodology the author proposes that their method overall is using Apriori algorithm to mine association rules, and then filtering out the rules that meet their requirements. If the amount of goods existing in the rules is less than the minimum inventory or less than half of the maximum inventory, the system will recommend the name of goods needed to be purchased at the same time for the procurement staff. Coming to the Association rules algorithm, it was proposed by [18] in 1993 firstly. And it was clear that the Association rules can discover the hidden relationships between things. The parameters support, confidence and lift are described in the Association Rules chapter.

The paper concludes by defining that the warehouse management system based on association rules has become a development trend and it will be welcomed by many procurement staff, and it can promote the further improvement and development of warehouse management.

Moreover, one of the methods of warehouse storage assignment policy that we use in this work is the Random storage assignment policy (RSAP). The RSAP is a fundamental method used in the order picking system in various situations in warehouse arrangement. The advantage of RSAP is that all stocks are distributed uniformly in every rack, and therefore each station’s workload is balanced. [19]

2.5. Problem Discussion

The literature on Automated storage and retrieval systems, their benefits and types were previously studied. Since Eurofork S.P.A. is developing a new system, there are no information about how the proposed system performs and the impact of the modifications when compared to the current system. So before actual investment and development, the company seeks a better understanding on its performances. The company expects higher performance in various parameters like arrival rates, number of SKUs, number of aisles and tiers, etc.

In order to understand how efficient the new system will be compared to the existing autonomous vehicle storage and retrieval systems under different configurations, a virtual simulation model is implemented, and the performances are recorded and compared. Since Eurofork is still in the early development phase of the product, we do not have real data to be analyzed, so the performance of this system is studied considering a virtual environment taking the data of an UK-based and registered non-store online retail from an online machine learning repository.

The main challenge for the Eurofork S.P.A. is creating a solution capable of collect different products with small or medium weight from different locations in the same warehouse line, completing an order emitted by an internal or external customer. The proposed system fulfils the Industry 4.0 concept, collecting automatically different products of a warehouse and building a rainbow pallet within the racks of the warehouse without having the need to retrieve the whole pallet.

One of the solution proposed by [10] was to design the crane such that a person can ride along (person-onboard). Instead of retrieving a full pallet automatically from the location, the person can pick one item from the location. A more common option to integrate item picking is when the AS/RS drops off the retrieved unit loads at a workstation. A picker at this workstation takes the required number of products from the unit-load after which the AS/RS moves the remainder of the load back into the storage rack. But these solution has many drawbacks in terms of safety and compactness of the warehouse, and overall cycle times, so the company is keenly interested in contributing its work for the Industry 4.0 revolution by developing a fully automated shuttle-based storage and retrieval system warehouse that can build rainbow pallets. Now for that reason, the work of this thesis is focused on simulating the idea of the company and getting results to continue with the new product development.

3. Tools utilized

3.1. Programming Language

For the execution of the thesis, Python has been used as the primary programming language. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed [20].

According to the latest TIOBE Programming Community Index, Python is one of the top 10 popular programming languages of 2017. Python is a general purpose and high-level programming language. We can use Python for developing desktop GUI applications, websites, and web applications. Also, Python, as a high-level programming language, allows us to focus on core functionality of the application by taking care of common programming tasks. The simple syntax rules of the programming language further make it easier for us to keep the code base readable and application maintainable. There are also a few reasons why Python is preferred to other programming languages [21].

1. Readable and Maintainable Code

   While writing a software application, we must focus on the quality of its source code to simplify maintenance and updates. The syntax rules of Python allow us to express concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes on code readability, and allows us to use English keywords instead of punctuations. Hence, we can use Python to build custom applications without writing additional code. The readable and clean code base helps us to maintain and update the software without putting extra time and effort.

2. Multiple Programming Paradigms

   Like other modern programming languages, Python also supports several programming paradigms. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features helps us to use Python for developing large and complex software applications.

3. Compatible with Major Platforms and Systems

   At present, Python is supported many operating systems. We can even use Python interpreters to run the code on specific platforms and tools. Also, Python is an interpreted programming language. It allows us to run the same code on multiple platforms without recompilation. Hence, we are not required to recompile the code after making any alteration. We can run the modified application code without recompiling and check the impact of changes made to the code immediately. The feature makes it easier for us to make changes to the code without increasing development time.

4. Robust Standard Library

   Its large and robust standard library makes Python score over other programming languages. The standard library allows us to choose from a wide range of modules according to our precise needs. Each module further enables us to add functionality to the Python application without writing additional code. For instance, while writing a web application in Python, we can use specific modules to implement web services, perform string operations, manage operating system interface or work with internet protocols. We can even gather information about various modules by browsing through the Python Standard Library documentation.

5. Many Open Source Frameworks and Tools

   As an open source programming language, Python helps us to curtail software development cost significantly. We can even use several open source Python frameworks, libraries, and development tools to curtail development time without increasing development cost. We even have option to choose from a wide range of open source Python frameworks and development tools according to our precise needs. For instance, we can simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle and Cherrypy. Likewise, we can accelerate desktop GUI application development using Python GUI frameworks and toolkits like PyQT, PyJs, PyGUI, Kivy, PyGTK and WxPython.

6. Simplify Complex Software Development

   Python is a general-purpose programming language. Hence, we can use the programming language for developing both desktop and web applications. Also, we can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. We can take advantage of the data analysis features of Python to create custom big data solutions without putting extra time and effort. At the same time, the data visualization libraries and APIs provided by Python helps us to visualize and present data in a more appealing and effective way. Many Python developers even use Python to accomplish artificial intelligence (AI) and natural language processing tasks. This feature has been utilized in the realization of this thesis work.

7. Adopt Test Driven Development

   We can use Python to create prototype of the software application rapidly. Also, we can build the software application directly from the prototype simply by refactoring the Python code. Python even makes it easier for us to perform coding and testing simultaneously by adopting test driven development (TDD) approach. We can easily write the required tests before writing code and use the tests to assess the application code continuously. The tests can also be used for checking if the application meets predefined requirements based on its source code.

3.2.Integrated Development Environment

An IDE, or Integrated Development Environment, enables programmers to consolidate the different aspects of writing a computer program. IDEs increase programmer productivity by combining common activities of writing software into a single application: editing source code, building executables, and debugging [22].

For the execution of python, Spyder has been used as the IDE for the realization of this thesis. Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers, and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package [23] .

Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used as a PyQt5 extension library, allowing developers to build upon its functionality and embed its components, such as the interactive console, in their own PyQt software.

Spyder has the following features [23],

1. An editor with syntax highlighting, introspection, code completion
2. Support for multiple IPython consoles
3. The ability to explore and edit variables from a GUI
4. A Help pane able to retrieve and render rich text documentation on functions, classes and methods automatically or on-demand
5. A debugger linked to IPdb, for step-by-step execution
6. Static code analysis, powered by Pylint
7. A run-time Profiler, to benchmark code
8. Project support, allowing work on multiple development efforts simultaneously
9. A built-in file explorer, for interacting with the filesystem and managing projects

10. A "Find in Files" feature, allowing full regular expression search over a specified scope
11. An online help browser, allowing users to search and view Python and package documentation inside the IDE
12. A history log, recording every user command entered in each console
13. An internal console, allowing for introspection and control over Spyder's own operation

3.3. Libraries used

Python library is a collection of functions and methods that allows you to perform lots of actions without writing your own code. To simplify, python library is a reusable chunk of code developed by programmers that can be included in our programs/ projects for better and easier analysis of our data. A brief description of the libraries used are provided below,

3.3.1. Pandas

Pandas is a python software package. It is a must to learn for data-science and dedicatedly written for Python language. It is a fast, demonstrative, and adjustable platform that offers intuitive data-structures. We can easily manipulate any type of data such as – structured or time-series data with this package [24].

Features of Pandas,

1. Pandas provide us with many Series and Data frames. It allows us to easily organize, explore, represent, and manipulate data.
2. Smart alignment and indexing featured in Pandas offers a perfect organization and data labeling.
3. Pandas has some special features that allows us to handle missing data or value with a proper measure.
4. This package offers such a clean code that even people with no or basic knowledge of programming can easily work with it.
5. It provides a collection of built-in tools that allows us to both read and write data in different web services, data-structure, and databases as well.
6. Pandas can support JSON, Excel, CSV, HDF5, and many other formats.

3.3.2. Matplotlib

Matplotlib is a Python library that uses Python Script to write 2-dimensional graphs and plots. Often mathematical or scientific applications require more than single axes in a representation. This library helps us to build multiple plots at a time. We can, however, use Matplotlib to manipulate different characteristics of figures as well [24].

Features of Matplotlib,

1. Matplotlib can create such quality figures that are really good for publication. Figures we create with Matplotlib are available in hardcopy formats across different interactive platforms.
2. We can use MatPlotlib with different toolkits such as Python Scripts, IPython Shells, Jupyter Notebook, and many other four graphical user interfaces.
3. A number of third-party libraries can be integrated with Matplotlib applications. Such as seaborn, ggplot, and other projection and mapping toolkits such as basemap.
4. An active community of developers are dedicated to help with any of the inquiries with Matplotlib.
5. Another good thing is that we can track any bugs, new patches, and feature requests on the issue tracker page from Github. It is an official page for featuring different issues related to Matplotlib.

3.3.3. Numpy

Numpy is a popular array – processing package of Python. It provides good support for different dimensional array objects as well as for matrices. Numpy is not only confined to providing arrays only, but it also provides a variety of tools to manage these arrays. It is fast, efficient, and good for managing matrices and arrays [24].

Features of Numpy,

1. Arrays of Numpy offer modern mathematical implementations on huge amount of data. Numpy makes the execution of these projects much easier and hassle-free.

2. Numpy provides masked arrays along with general array objects. It also comes with functionalities such as manipulation of logical shapes, discrete Fourier transform, general linear algebra, and many more.

3. While we change the shape of any N-dimensional arrays, Numpy will create new arrays for that and delete the old ones.

4. This python package provides useful tools for integration. We can easily integrate Numpy with programming languages such as C, C++, and Fortran code.

5. Numpy provides such functionalities that are comparable to MATLAB. They both allow users to get faster with operations.

3.3.4. Scipy

Scipy is an open-source python library that is used for both scientific and technical computation. It is a free python library. And very suitable for machine learning. However, computation is not the only task that makes scipy special. It is also very popular for image manipulation, as well [24].

Features of Scipy,

1. Scipy contains different modules. These modules are suitable for optimization, integration, linear algebra, and statistics, as well.
2. It makes the best use of Numpy arrays for general data structures. In fact, Numpy is an integrated part of Scipy.
3. Scipy can handle 1-d polynomials in two ways. Whether we can use poly1d class from numpy or we can use co-efficient arrays to do the job.
4. High-level scipy contains not only numpy but also numpy.lib.scimath as well. But it is better to use them from their direct source.
5. A supporting community of Scipy is always there to answer our regular questions and solve any issues if aroused.

3.3.5. Scikit Learn

Scikit learn is a simple and useful python machine learning library. It is written in python, cython, C, and C++. However, most of it is written in the Python programming language. It is a free machine learning library. It is a flexible python package that can work in complete harmony with other python libraries and packages such as Numpy and Scipy [24].

Features of Scikit Learn,

1. Scikit Learn comes with a clean and neat API. It also provides very useful documentation for beginners.
2. It comes with different algorithms – classification, clustering, and regression. It also supports random forests, k-means, gradient boosting, DBSCAN and others
3. This package offers easy adaptability.
4. Scikit Learn offers easy methods for data representation. Whether we want to present data as a table or matrix, it is all possible with Scikit Learn.
5. It allows us to explore through digits that are written in hands. We can not only load but also visualize digits-data as well.

3.3.6. Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. It is built on top of matplotlib and closely integrated with pandas data structures.

Here is some of the functionality that seaborn offers:

1. A dataset-oriented API for examining relationships between multiple variables
2. Specialized support for using categorical variables to show observations or aggregate statistics
3. Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
4. Automatic estimation and plotting of linear regression models for different kinds dependent variables
5. Convenient views onto the overall structure of complex datasets
6. High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations
7. Concise control over matplotlib figure styling with several built-in themes
8. Tools for choosing color palettes that faithfully reveal patterns in your data

Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. [25]

3.3.7. MLxtend

MLxtend is a library that implements a variety of core algorithms and utilities for machine learning and data mining. The fundamental goal of MLxtend is to make frequently used tools accessible to researchers and data scientists in industries focusing on user friendly and intuitive APIs and compatibility to existing machine learning libraries, such as scikit-learn, libraries related to Apriori Algorithms, etc. MLxtend implements a wide variety of functions, highlights include sequential feature selection algorithms, implementations of stacked generalization for classification and regression, and algorithms for frequent pattern mining. MLxtend provides a large variety of different utilities that build upon and extend the capabilities of Python’s scientific computing stack. [26]

3.3.8. Wordcloud

Wordcloud also termed as Tag cloud gives us the visual representation of text data. It was developed by Andreas Mueller. It displays a list of words and the importance of each being shown with different font size or color. The main use of wordcloud format is for quickly perceiving the most prominent terms in a dataset. Python is adapted to draw this kind of representation.

3.4. Flexsim

FlexSim organization creates simulation software that models, simulates, predicts, and visualizes systems in material handling, warehousing, manufacturing, healthcare, logistics, mining, etc. It is a powerful and user-friendly software and helps to optimize current and planned processes, to identify and decrease waste, increase revenue, and reduce cost. [27]

Flexsim simulation software is the next generation in discrete-event simulator and it is a Windows based, object-oriented simulation environment for modeling discrete-event flow processes like manufacturing, material handling, and office workflow in stunning 3D virtual reality. The key advantages of the simulator are,

- Fully object-oriented with complete C++ integration
- Models created graphically, using drag and drop
- Amazing 3D virtual reality animation
- Exceptionally intuitive, easy-to-learn interface
- Unsurpassed flexibility and power [27]

The other main features of the simulator are mentioned below,

1. Robust standard objects

FlexSim simulator comprises a standard object library and each of this object contains the pre-built logic and task execution to imitate the resources found in real-world tasks and operations. These objects of FlexSim are programmed in four classes which include fixed resource class, task executer class, node class and visual object class. The FlexSim simulator operates on an object-oriented design.

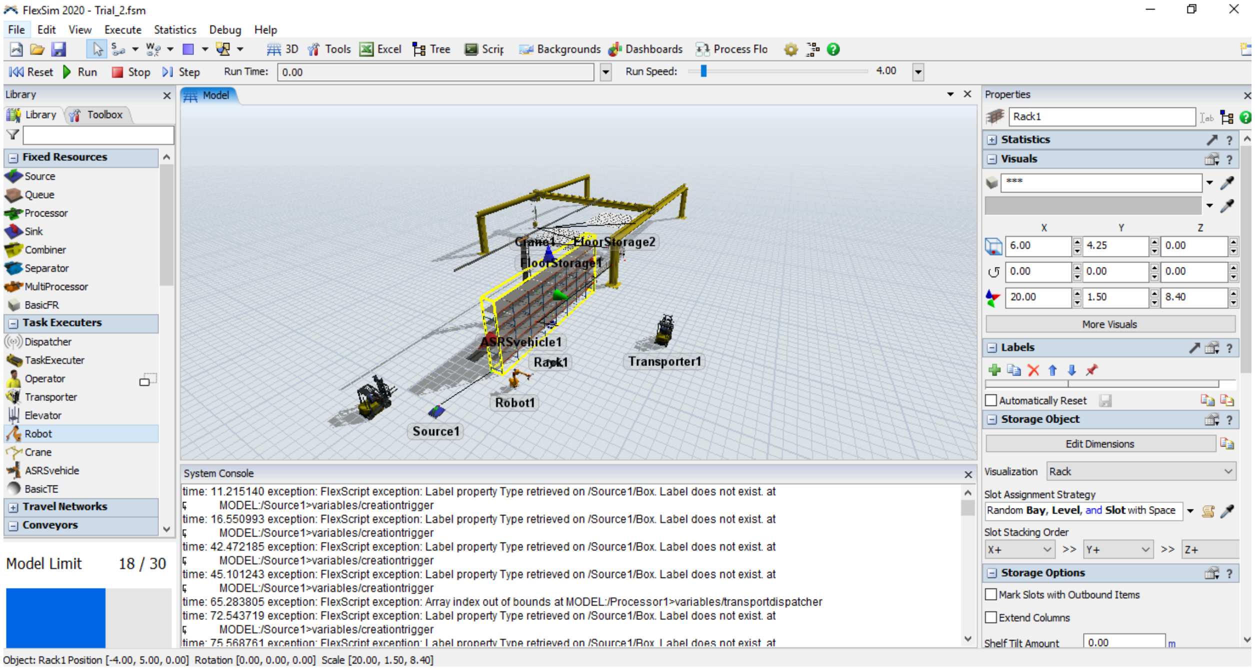
2. Logic building tools

The logic for a FlexSim model could be built using few or no computer coding. Most of the standard objects of flexsim contain an array of drop-down lists, properties windows, and triggers that grant the user to customize the logic necessary for an accurate model of the system. It also includes a flowcharting tool to create the logic for a model using pre-built activity blocks.

3. Drag-and-drop controls

Users of the simulator can construct the model by dragging and dropping predefined 3D objects into a "model view" to layout and link the model with each other according to needs. Pro users also have the choice to specify and modify predefined object parameters and behaviors using FlexScript and C++ programming languages.

A basic example of the Flexsim simulation software's layout is presented in the figure below,



Run Time: 0.00

Properties: Rack1

Visuals: ***

	X	Y	Z
	6.00	4.25	0.00
	0.00	0.00	0.00
	20.00	1.50	8.40

□ Automatically Reset

Visualization: Rack

Slot Assignment Strategy: Random **Bay**, **Level**, and **Slot** with Space

Slot Stacking Order: X+ >> Y+ >> Z+

□ Mark Slots with Outbound Items

□ Extend Columns

Shelf Tilt Amount: 0.00 m

```
time: 11.215140 exception: FlexScript exception: Label property Type retrieved on /Source1/Box. Label does not exist. at
        MODEL:/Source1>variables/creationtrigger
time: 16.550993 exception: FlexScript exception: Label property Type retrieved on /Source1/Box. Label does not exist. at
        MODEL:/Source1>variables/creationtrigger
time: 42.472185 exception: FlexScript exception: Label property Type retrieved on /Source1/Box. Label does not exist. at
        MODEL:/Source1>variables/creationtrigger
time: 45.101243 exception: FlexScript exception: Label property Type retrieved on /Source1/Box. Label does not exist. at
        MODEL:/Source1>variables/creationtrigger
time: 65.283805 exception: FlexScript exception: Array index out of bounds at MODEL:/Processor1>variables/transportdispatcher
time: 72.543719 exception: FlexScript exception: Label property Type retrieved on /Source1/Box. Label does not exist. at
        MODEL:/Source1>variables/creationtrigger
time: 75.568761 exception: FlexScript exception: Label property Type retrieved on /Source1/Box. Label does not exist. at
```

Object: Rack1 Position [-4.00, 5.00, 0.00] Rotation [0.00, 0.00, 0.00] Scale [20.00, 1.50, 8.40]

Figure 7 Flexsim Simulation software basic layout

4. Data Analysis

4.1. Description of dataset

The data selected contains events history of an UK-based and registered non-store online retail, and all the transactions occurring between 01st of December 2010 to 09th of December 2011. This company primarily sells unique all-occasion gifts, and many customers of the company are said to be wholesalers. This is a transnational dataset and each row in the file represents an event and all the events are related to products and users. Each event is like many-to-many relation between products and users.

Data Source - [28]
Data Provider – UCI Machine learning repository

UCI Machine learning repository is a large repository of data where plentiful datasets are available. This repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. According to the UCI Machine Learning Repository, this dataset was provided by Dr.Daqing Chen, Director: Public Analytics group. chend '@' lsbu.ac.uk, School of Engineering, London South Bank University, London SE1 0AA, UK.

Python language is picked to analyze the dataset since diverse collection of libraries are available for data analysis. The main reasons to choose python are given below and which are well explained in the chapter 3 Tools utilized.

1. Readable and Maintainable Code
2. Multiple Programming Paradigms
3. Compatible with Major Platforms and Systems
4. Robust Standard Library
5. Many Open Source Frameworks and Tools
6. Simplify Complex Software Development
7. Adopt Test Driven Development

Initially the dataset was analyzed and found out that it has the following number of rows and columns shown in the table below,

Number of rows	Number of columns
541909	8

The Columns contained the fields,

1	Invoice no.	6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
2	Stock code	Product code, a 5-digit integral number uniquely assigned to each distinct product.
3	Description	Product name.
4	Quantity	The quantities of each product per transaction.
5	Invoice date	Invoice Date and time. The day and time when each transaction was generated.
6	Unit price	Unit price. Numeric, Product price per unit in sterling pounds.
7	Customer ID	Customer number, a 5-digit integral number uniquely assigned to each customer.
8	Country	Country name. The name of the country where each customer resides.

Table 1 Description of the dataset

4.2.Techniques for preprocessing of dataset

In any Machine learning process, Data preprocessing is that step in which the data gets transformed, or encoded, to bring it to such a state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm. [29]

A dataset can be viewed as a collection of data objects, which are often also called as a records, points, vectors, patterns, events, cases, samples, observations, or entities. Data objects are described by a number of features, that capture the basic characteristics of an object, such as the mass of a physical object or the time at which an event occurred, etc. Features are often called as variables, characteristics, fields, attributes, or dimensions.

A feature is an individual measurable property or characteristic of a phenomenon being observed. For instance, color, mileage, and power can be considered as features of a car. There are different types of features that we can come across when we deal with data.

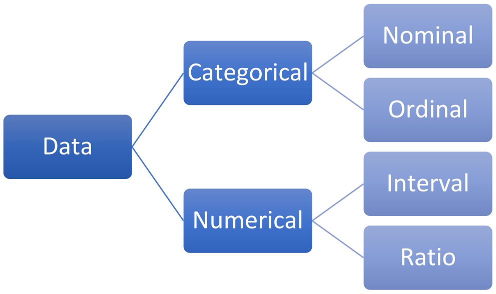


Figure 8 Statistical data types

Features can be,

Categorical: Features whose values are taken from a defined set of values. For instance, days in a week: {Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday} is a category because its value is always taken from this set. Another example could be the Boolean set: {True, False}

Numerical: Features whose values are continuous or integer valued. They are represented by numbers and possess most of the properties of numbers. For instance, number of steps you walk in a day, or the speed at which you are driving your car at.

The general techniques and steps that are involved in the preprocessing of the dataset are,

- Data Quality Assessment
- Feature Aggregation
- Feature Sampling
- Dimensionality Reduction
- Feature Encoding

4.2.1. Data Quality Assessment

Because data is often taken from multiple sources which are normally not too reliable and that too in different formats, more than half our time is consumed in dealing with data quality issues when working on a machine learning problem. It is simply unrealistic to expect that the data will be perfect. There may be problems due to human error, limitations of measuring devices, or flaws in the data collection process. Let us go over a few of them and methods to deal with them.

i. Missing values

It is very much usual to have missing values in your dataset. It may have happened during data collection, or maybe due to some data validation rule, but regardless missing values must be taken into consideration.

- Eliminate rows with missing data

Simple and sometimes effective strategy. Fails if many objects have missing values. If a feature has mostly missing values, then that feature itself can also be eliminated.

- Estimate missing values

If only a reasonable percentage of values are missing, then we can also run simple interpolation methods to fill in those values. However, most common method of dealing with missing values is by filling them in with the mean, median or mode value of the respective feature.

ii. Inconsistent values

We know that data can contain inconsistent values. Most probably we have already faced this issue at some point. For instance, the 'Address' field contains the 'Phone number'. It may be due to human error or maybe the information was misread while being scanned from a handwritten form.

It is therefore always advised to perform data assessment like knowing what the data type of the features should be and whether it is the same for all the data objects.

iii. Duplicate values

A dataset may include data objects which are duplicates of one another. It may happen when say the same person submits a form more than once. The term deduplication is often used to refer to the process of dealing with duplicates.

In most cases, the duplicates are removed so as to not give that particular data object an advantage or bias, when running machine learning algorithms.

4.2.2. Feature Aggregation

Feature Aggregations are performed so as to take the aggregated values in order to put the data in a better perspective. Think of transactional data, suppose we have day-to-day transactions of a product from recording the daily sales of that product in various store locations over the year. Aggregating the transactions to single store-wide monthly or yearly transactions will help us reducing hundreds or potentially thousands of transactions that occur daily at a specific store, thereby reducing the number of data objects.

This results in reduction of memory consumption and processing time. Aggregations provide us with a high-level view of the data as the behavior of groups or aggregates is more stable than individual data objects

4.2.3. Feature Sampling

Sampling is a very common method for selecting a subset of the dataset that we are analyzing. In most cases, working with the complete dataset can turn out to be too expensive considering the memory and time constraints. Using a sampling algorithm can help us reduce the size of the dataset to a point where we can use a better, but more expensive, machine learning algorithm.

The key principle here is that the sampling should be done in such a manner that the sample generated should have approximately the same properties as the original dataset, meaning that the sample is representative. This involves choosing the correct sample size and sampling strategy.

Simple Random Sampling dictates that there is an equal probability of selecting any particular entity. It has two main variations as well,

i. Sampling without Replacement: As each item is selected, it is removed from the set of all the objects that form the total dataset.

ii. Sampling with Replacement: Items are not removed from the total dataset after getting selected. This means they can get selected more than once.

Although Simple Random Sampling provides two great sampling techniques, it can fail to output a representative sample when the dataset includes object types which vary drastically in ratio. This can cause problems when the sample needs to have a proper representation of all object types, for example, when we have an imbalanced dataset.

An Imbalanced dataset is one where the number of instances of a class(es) are significantly higher than another class(es), thus leading to an imbalance and creating rarer class(es).

It is critical that the rarer classes be adequately represented in the sample. In these cases, there is another sampling technique which we can use, called Stratified Sampling, which begins with predefined groups of objects. There are different versions of Stratified Sampling too, with the simplest version suggesting equal number of objects be drawn from all the groups even though the groups are of different sizes.

4.2.4. Dimensionality Reduction

Most real-world datasets have a large number of features. For example, consider an image processing problem, we might have to deal with thousands of features, also called as dimensions. As the name suggests, dimensionality reduction aims to reduce the number of features, but not simply by selecting a sample of features from the feature-set, which is something else like Feature Subset Selection or simply Feature Selection.

Conceptually, dimension refers to the number of geometric planes the dataset lies in, which could be high so much so that it cannot be visualized with pen and paper. More the number of such planes, more is the complexity of the dataset.

A few major benefits of dimensionality reduction are:

- Data Analysis algorithms work better if the dimensionality of the dataset is lower. This is mainly because irrelevant features and noise have now been eliminated.
- The models which are built on top of lower dimensional data are more understandable and explainable.
- The data may now also get easier to visualize!

Features can always be taken in pairs or triplets for visualization purposes, which makes more sense if the feature set is not that big.

4.2.5. Feature Encoding

As mentioned before, the whole purpose of data preprocessing is to encode the data in order to bring it to such a state that the machine now understands it. Feature encoding is basically performing transformations on the data such that it can be easily accepted as input for machine learning algorithms while still retaining its original meaning.

4.3. Pre-processing of the dataset using python

4.3.1. Data Cleaning

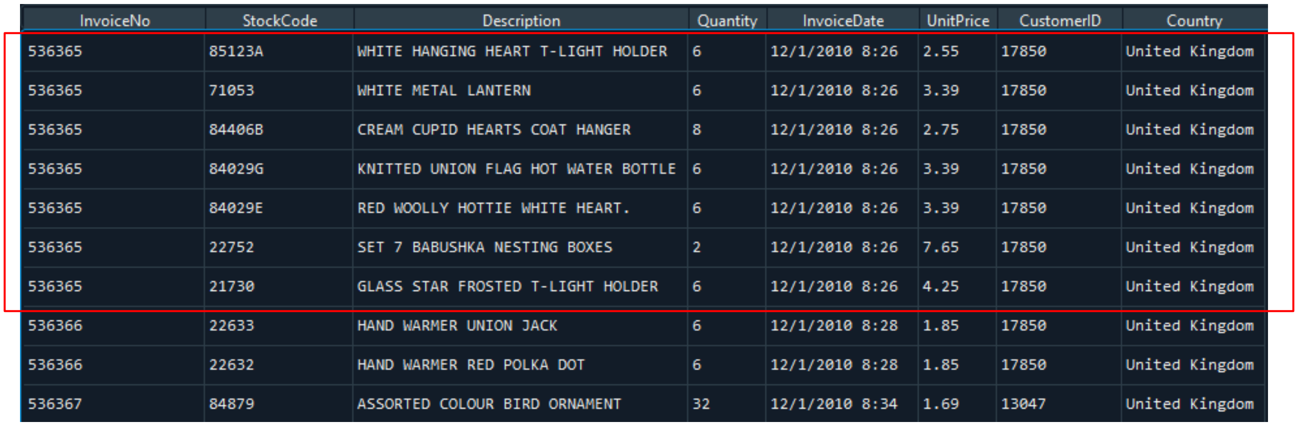
Data cleaning or cleansing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.

First of all, the dataset of the UK based retailer was loaded into the analysis software and the structure of the dataset was studied. In order to start the cleansing process first the visualization libraries seaborn and matplotlib are loaded in order to better understand the dataset visually.

```
"""Loading the dataset"""

df = pd.read_csv(r'C:\Users\S250203\Documents\Mahilan-S250203\
\Original_dataset.csv',sep=',',encoding='latin1')
```

Code Snippet 1 Loading the dataset into python

A portion of the dataset is shown below in Figure 9 and each of these parameters is explained in the Table 1. Initially the raw dataset has 541909 rows and 8 columns. Each entry in the dataset belongs to one unique Invoice number and a unique customer, and each stock code represents a unique product which is given in the description. In the Figure 9 the values inside the red box represents a single transaction consisting of different products. This is identified by a single Invoice number for all the 7 rows.



InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNION JACK	6	12/1/2010 8:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RED POLKA DOT	6	12/1/2010 8:28	1.85	17850	United Kingdom
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	12/1/2010 8:34	1.69	13047	United Kingdom

Figure 9 A view of a portion of dataset in python

```
import seaborn as sns

import matplotlib.pyplot as plt

"""Checking if there are any null values in the dataset"""

df.isnull().values.any()
```

```
OUTPUT – True
```

```
"""creating a heat map on missing values"""
```

```
cols = df.columns[:]

colours = ['#000099', '#ffff00'] # specify the colours - yellow is missing. blue is not missing.

sns.heatmap(df[cols].isnull(), cmap=sns.color_palette(colours))
```

OUTPUT

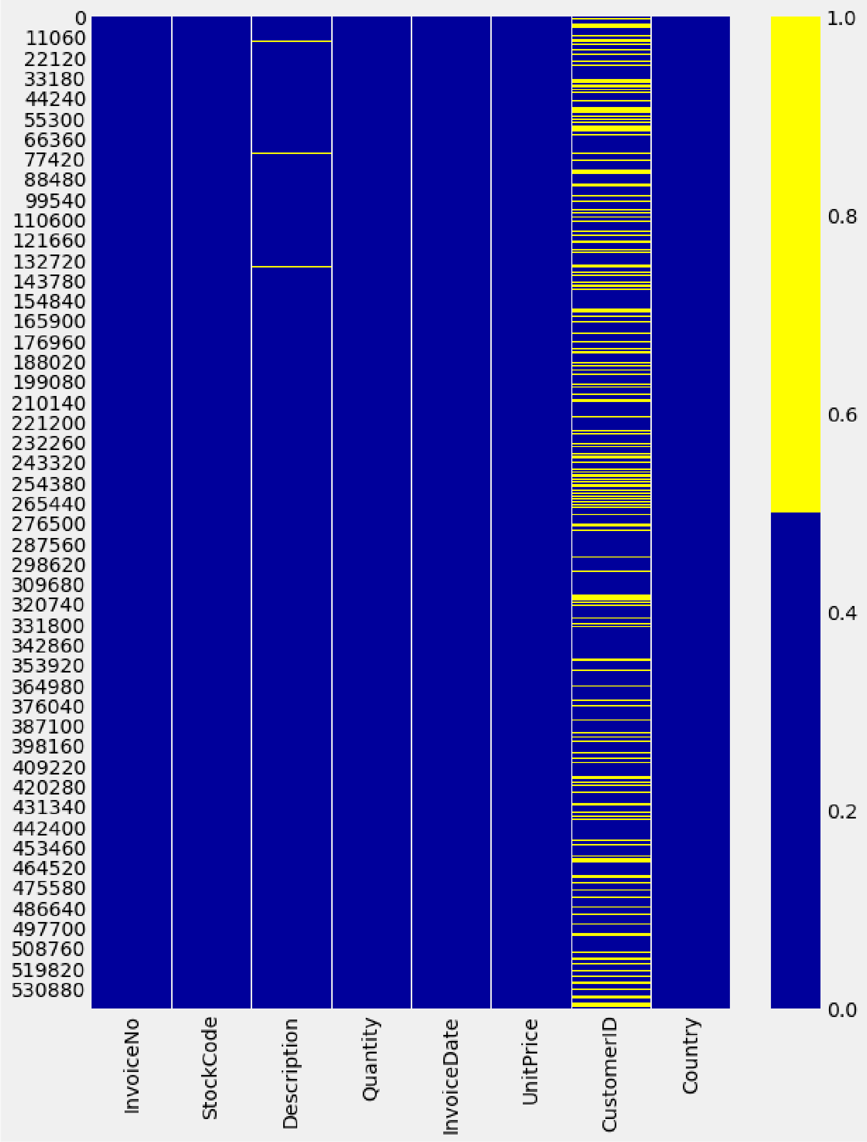
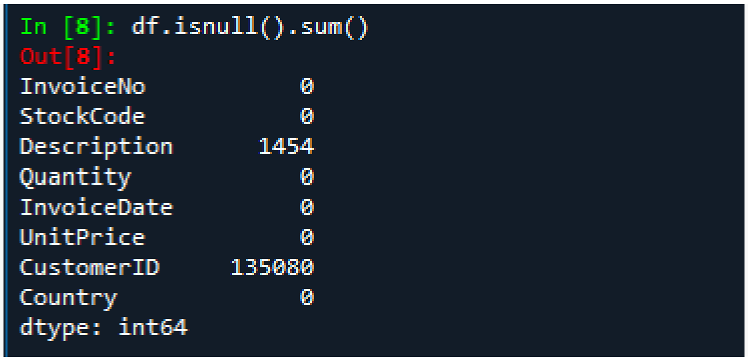


Figure 10 Heatmap of missing values in dataset

```
"""how many null values each column has"""

df.isnull().sum()
```

```
OUTPUT
```



```
In [8]: df.isnull().sum()
Out[8]:
InvoiceNo           0
StockCode           0
Description      1454
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     135080
Country             0
dtype: int64
```

Figure 11 Missing values in each column

```
"""Drop rows having missing values"""

Drop_data = df.dropna()

"""To drop duplicate values"""

Dataset = Drop_data.drop_duplicates()

"""dropping transactions that were cancelled"""

dataset = dataset[~dataset['InvoiceNo'].astype(str).str.startswith('C')]

""" Stripping extra spaces in the description """

dataset['Description'] = dataset['Description'].str.strip()
```

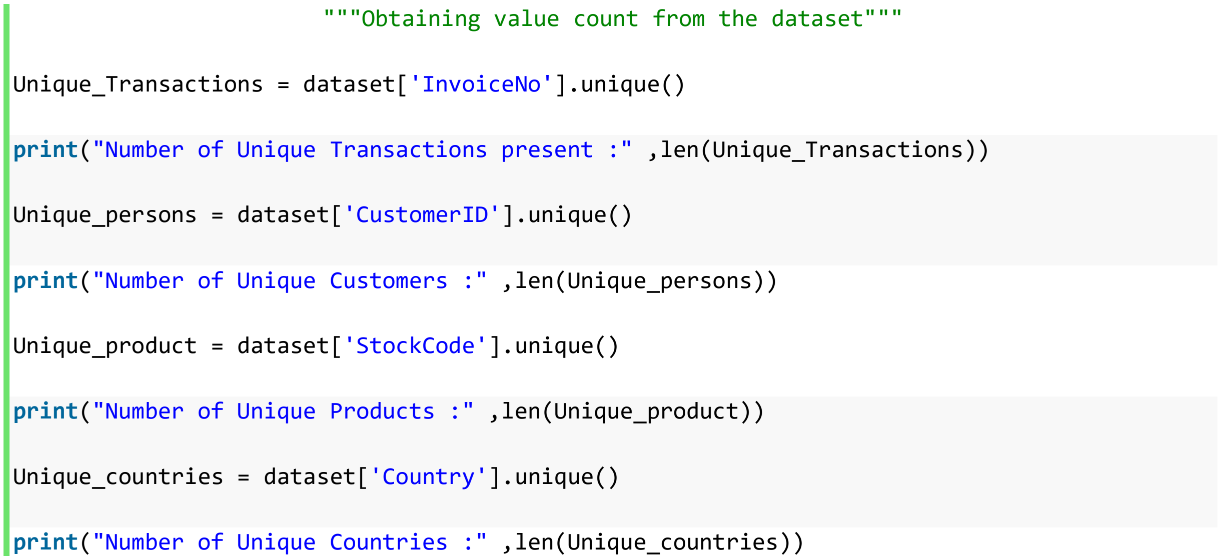
Code Snippet 2 Data set Cleansing

After the data was loaded into the program, the visualization libraries were also loaded along. Then the data cleansing process is started, initially a code that looks for any missing values in the dataset and gives Boolean output was run and the output was obtained. The output was **True**, so the next step was to find how many values were missing and from which columns. For this purpose, the seaborn library has been used, which provided us greater insight of the missing values using a heatmap. The heatmap is shown in Figure 8. Here the missing values were denoted by yellow lines and blue represented complete data. This heatmap clearly shows us that the fields Description and Customer Id has missing values. Using another code, it has been found that the Description column lacked **1454** values and the Customer Id lacked **135080** values. Now there were two choices to be made, whether to impute data or to delete it. Since the Description column and Customer Id has unique information, it has been decided not to impute data and deletion of the fields were made. The deletion of rows was done using a code that drops rows which had missing values. Now after these fields were dropped, fields that had duplicate rows (which means rows having same values) were also dropped using the drop duplicates command. For our analysis we are only concerned on the orders on which purchase were made, but this dataset also has orders which were cancelled, which were represented by the character 'C' in the Invoice number. We delete these transactions which were cancelled using a python code. Now the last step in this data cleansing was clearing the extra spaces in the description field, so that we do not have any error in the future during analysis. Now after all the cleansing operations were done the data is ready for the next step, that is the Data Visualization.

	Before Cleansing	After Cleansing
Rows	541909	392732
Columns	8	8

Table 2 Dataset size before and after cleansing

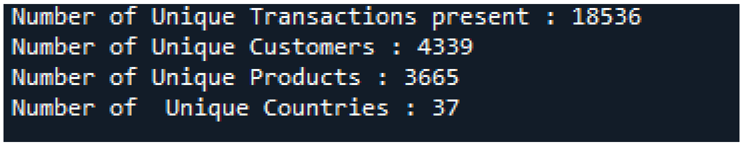
From the above table we can see that an 27.53 % reduction of size of dataset is obtained after cleansing. If we were to use the dataset without cleansing, various errors could have occurred in the data analysis.



```
                        """Obtaining value count from the dataset"""

Unique_Transactions = dataset['InvoiceNo'].unique()

print("Number of Unique Transactions present :" ,len(Unique_Transactions))

Unique_persons = dataset['CustomerID'].unique()

print("Number of Unique Customers :" ,len(Unique_persons))

Unique_product = dataset['StockCode'].unique()

print("Number of Unique Products :" ,len(Unique_product))

Unique_countries = dataset['Country'].unique()

print("Number of Unique Countries :" ,len(Unique_countries))
```

Code Snippet 3 Obtaining value count from the dataset

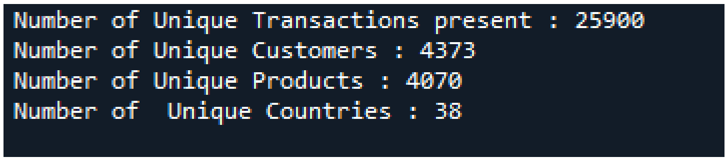
OUTPUT



```
Number of Unique Transactions present : 18536
Number of Unique Customers : 4339
Number of Unique Products : 3665
Number of  Unique Countries : 37
```

Figure 12 Data count after Cleansing

The data count of each required variable is obtained and before and after cleansing values are compared to get a clear idea of cleansed data. The data count of after and before cleansing are showed in Figure 12 and Figure 13 respectively.



```
Number of Unique Transactions present : 25900
Number of Unique Customers : 4373
Number of Unique Products : 4070
Number of  Unique Countries : 38
```

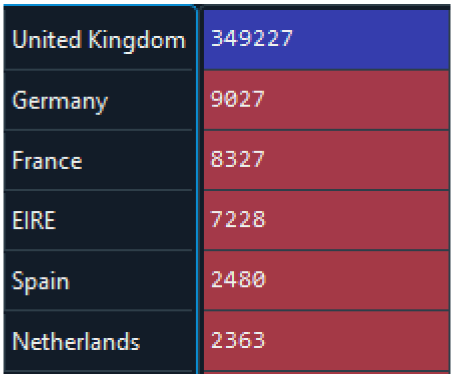
Figure 13 Data count before cleansing

The next step is Visualizing the data, before that we obtain the top countries in terms of orders received using the below code. The output of the top countries is given in Figure 14.

```
"""Top 5 countries in terms of sales"""

Top_countries=dataset['Country'].value_counts()[:5]
```

Code Snippet 4 Top 5 countries in terms of sales

OUTPUT



```
United Kingdom   349227
Germany          9027
France           8327
EIRE             7228
Spain            2480
Netherlands      2363
```

Figure 14 Top 5 countries in terms of sales

We can see that 88.92 % of transactions (349227 transactions out of 392732 transactions) were from the United Kingdom. The chart in the Figure 15 clearly shows the dominance of sales in United Kingdom compared to the other countries.

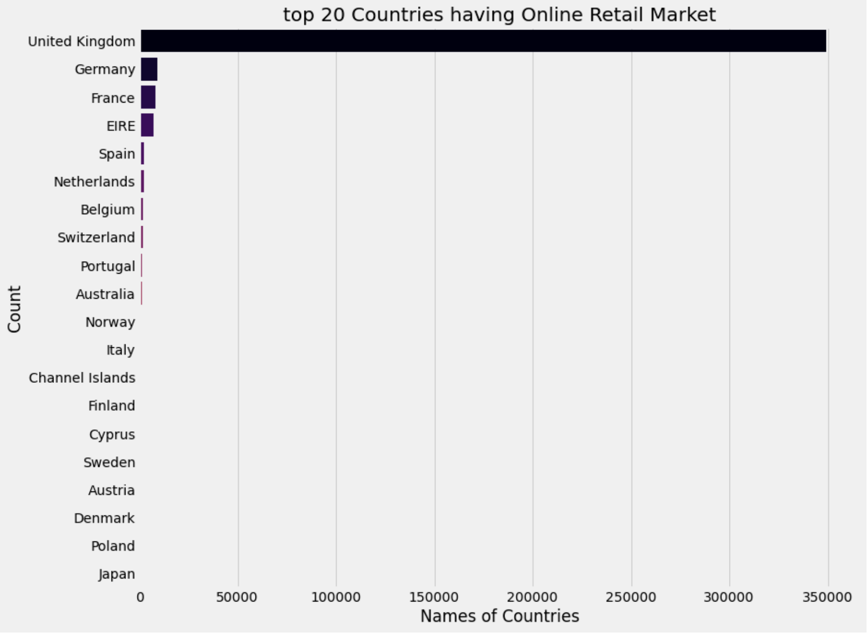
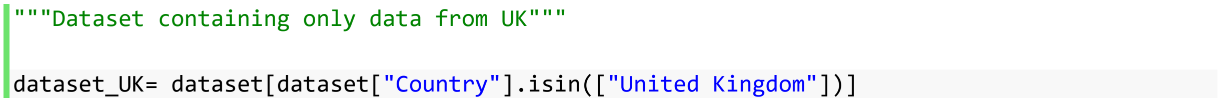


Figure 15 Top 20 countries in terms of sales

So, the choice was made to use only the data from the country United Kingdom for future analysis for simplification. The dataset is limited to have only the transactions from UK using the code below. The new dataset is now on is named as dataset_UK.



```
"""Dataset containing only data from UK"""

dataset_UK= dataset[dataset["Country"].isin(["United Kingdom"])]
```

Code Snippet 5 To create dataset containing orders only from UK

4.3.2. Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data [30]. The libraries like seaborn and matplotlib are used to create various charts and graphs for better understanding of the dataset. First we use a uselibrary called Wordcloud to find which words are used often in the description column of the dataset. This gives us an insight of what type products are sold by the retailer and what products are often bought by the customers.

```
                  """Generating most occurring word in description list"""

from wordcloud import WordCloud

from wordcloud import STOPWORDS

stopwords = set(STOPWORDS)

wordcloud = WordCloud(background_color = 'white', width = 900, height = 900).generate(str(
dataset_UK['Description']))

print(wordcloud)

plt.rcParams['figure.figsize'] = (12, 12)

plt.axis('off')

plt.imshow(wordcloud)

plt.title('Most Occuring words in the Description list', fontsize = 20)

plt.show()
```

OUTPUT

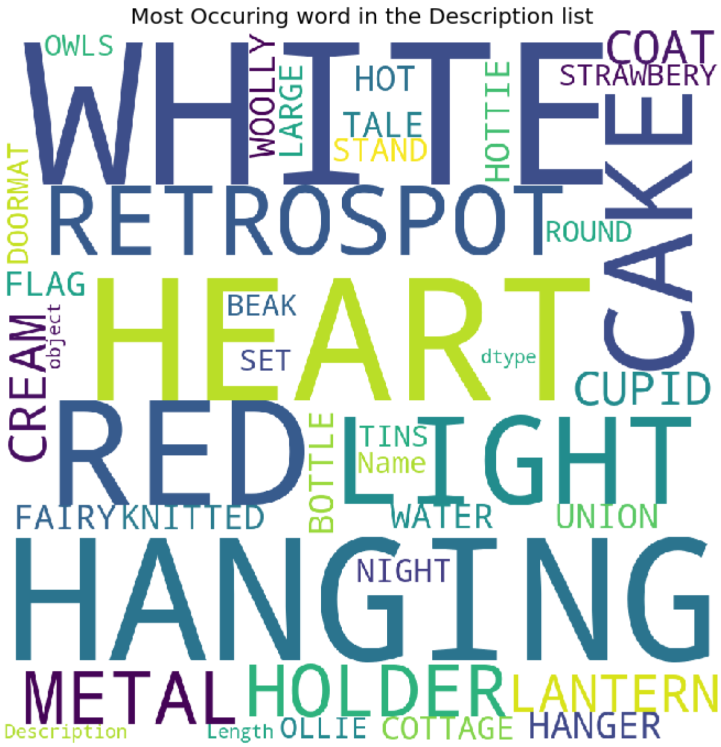


Figure 16 Most Occurring words in the Description list

The next visualization done was to find the top 30 products that were sold in the United Kingdom. From this plot we can understand which products were bought by most of the people.

```
"""Top 30 products sold in UK"""

Top_sold_Products_2=dataset_UK['Description'].value_counts()[:30]

plt.bar(Top_sold_Products_2.index.to_list(), Top_sold_Products_2.to_list())

plt.xticks(rotation=90)

plt.title('Dataset only UK')

plt.ylabel('Number of occurance')

plt.xlabel('Brands')

plt.show()
```

Code Snippet 6 Generating chart for top 30 products sold in UK

OUTPUT

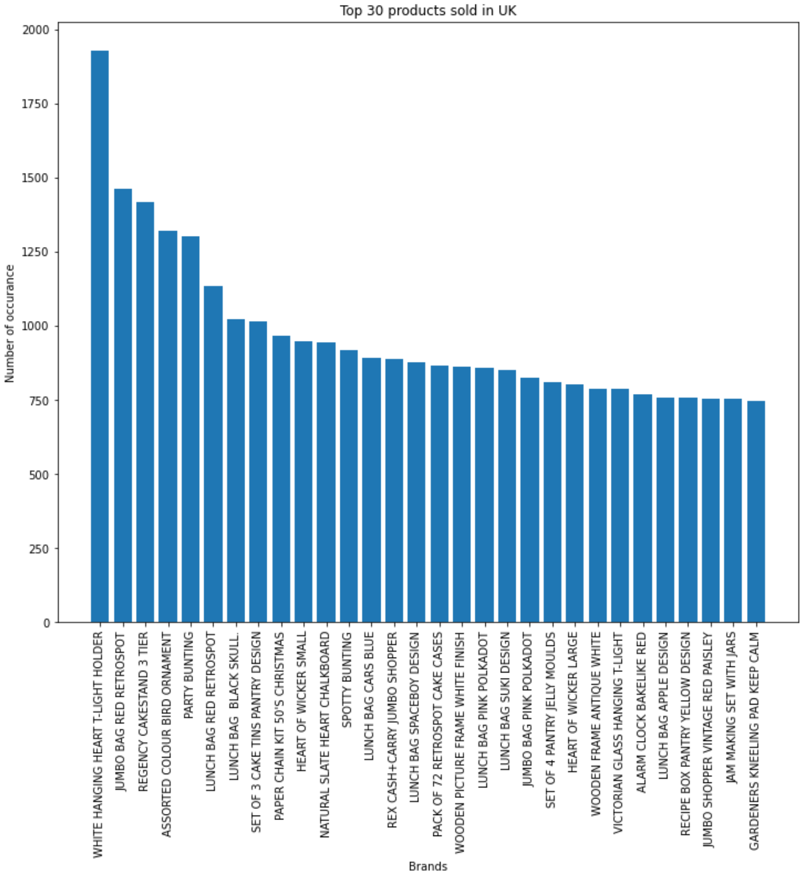


Figure 17 Top 30 products sold in UK

A time series analysis is visualized on the purchase date and sales amount per day. This graph shows various peaks and troughs in terms of sales in specific period of time. This time series is generated by sales versus Invoice date, where sales represent Unit price of a product multiplied by quantity of product sold in each transaction.

```python
                """ Each row sales obtained from unit price x quantity sold """

dataset['Sales'] = dataset['UnitPrice'] * dataset['Quantity']

                           """Time series Analysis of orders in UK"""

timeseries = dataset[dataset['Country'] == 'United Kingdom']

timeseries.plot(x = 'InvoiceDate', y = 'Sales')

plt.title('Time-Series for United Kingdom', fontsize = 20)

plt.xlabel('Date of Purchase')

plt.ylabel('Sales Amount')

plt.ylim(0, 10000)

plt.show()
```

Code Snippet 7 Time series Analysis of orders in UK

OUTPUT

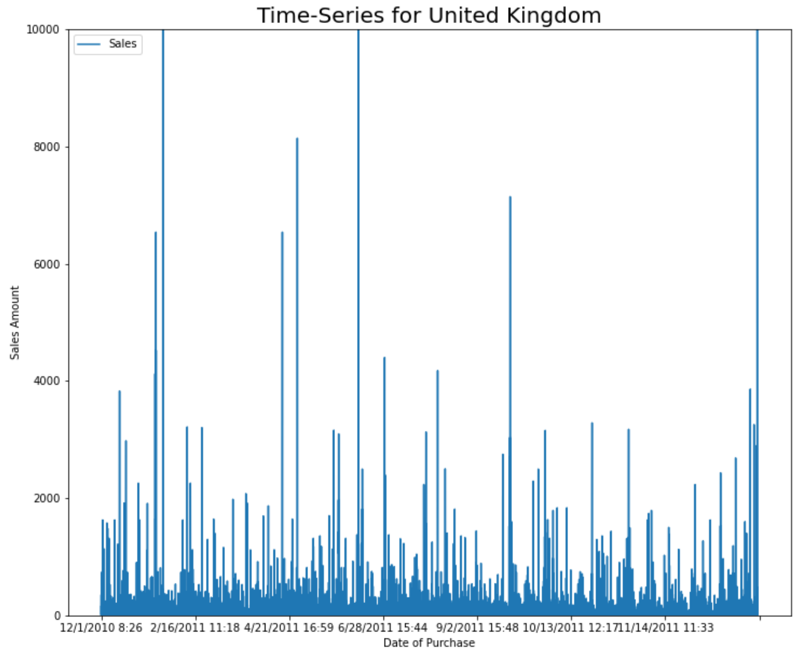


Figure 18 Time series Analysis of orders in UK

5. Dataset Mining to generate Association Rules

Therefore, having completed the data pre-processing and data analysis we proceed into the very crucial part, that is the data mining. As already discussed, Data mining is a process of extracting information and knowledge implicit in a large, incomplete, noisy, vague, random large-scale data. The association rule as the important branch of data mining is a high-level and intelligent data processing and analysis technology, We can use the association rules to mine the relationships among data, and we can get the potential value of the information hidden in the massive data. [9]

In order to generate the Association rules, we have to further prepare the data. Before generating the Association rules let us take a brief look at the popular method called Market Basket Analysis also known as MBA. Market Basket Analysis (MBA) is an accidental transaction pattern that tells us the fact that purchasing some products will affect the purchasing of other products in a store or any retail. The Market Basket Analysis uses the Apriori algorithm to extract rules to find out which product is associated with which in terms of sales to increase the profit. Since we are focused more on warehousing, we can use this data obtained to arrange the warehouse in such a manner that the frequently bought products are placed nearby in the warehouse racks in order to reduce the distance travelled by the shuttle and also to reduce the picking time.

Starting off with the Market Basket Analysis, we use the latest version of the dataset obtained after cleansing containing transactions from United Kingdom. Firstly, a basket data should be created. A Basket data here refers to a table that will contain the Quantity of each items bought per transaction. Then this Basket data must be encoded to be used in the python library MLxtend. In market basket analysis, the quantity of each item bought is not important, instead whether an item is bought or not is of first priority. Because, we only would like to know, what is the association of buying some items with buying some others. So, we encode the basket data into a binary data that shows whether an item is bought or not.

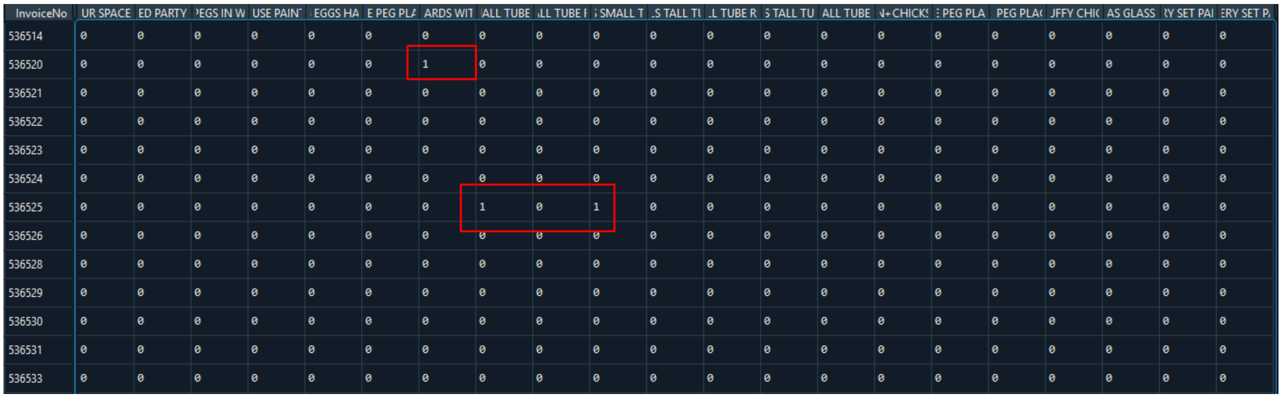
First we create the Data Basket and encode the data using one hot encoding technique using the following code,

```
"""Creating the Basket data"""

basket_UK = (dataset[dataset['Country'] =="United Kingdom"]

          .groupby(['InvoiceNo', 'Description'])['Quantity']

          .sum().unstack().reset_index().fillna(0)

          .set_index('InvoiceNo'))
```

```
"""Creating one hot encoding"""

def hot_encode(x):

    if(x<= 0):

        return 0

    if(x>= 1):

        return 1

basket_encoded = basket_UK.applymap(hot_encode)

basket_UK = basket_encoded
```

Code Snippet 8 Creating one hot encoded data on dataset

OUTPUT



InvoiceNo	UR SPACE	ED PARTY	EGS IN W	USE PAINT	EGGS HA	E PEG PLA	ARDS WIT	ALL TUBE	LL TUBE I	SMALL T	S TALL TU	L TUBE R	TALL TU	ALL TUBE	N+CHICKS	E PEG PLA	PEG PLA(	JFFY CHI(	AS GLASS	RY SET PAI	ERY SET P
536514	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
536520	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
536521	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
536522	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
536523	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
536524	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
536525	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
536526	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
536528	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
536529	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
536530	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
536531	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
536533	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 19 Result of one hot encoded data

From the Figure 19 we see that a cross tab table is created between Invoice Number and Description of products columns (Table 1). This basket has **16649 rows and 3833 columns**. When a product is bought by a customer it is represented as (1) or if it is not it is represented as (0). This is obtained through the unique Invoice number assigned for each order for a unique customer.

Now having completed the creation of the data basket we move on to the next step that is applying the Apriori algorithm and generating the rules. Apriori algorithm is simply used to find the frequently bought items in the whole dataset.

```
   """Importing the libraries mlxtend and sub libraries Apriori and association rules"""

from mlxtend.frequent_patterns import apriori

from mlxtend.frequent_patterns import association_rules
```

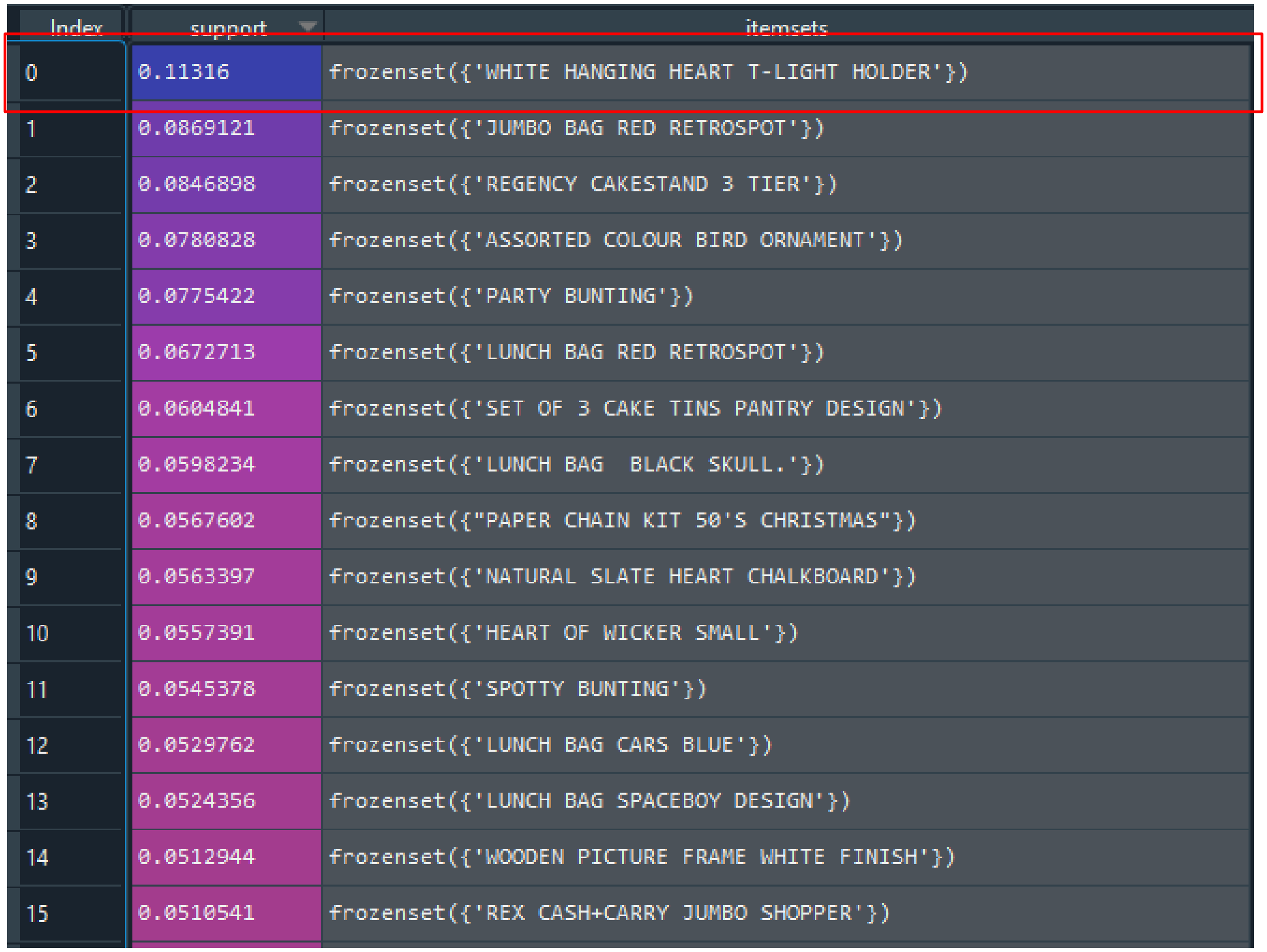
Code Snippet 9 Importing libraries required for generating rules

```
frq_items = apriori(basket_UK, min_support = 0.02, use_colnames = True)
```

Code Snippet 10 Applying the Apriori algorithm

A threshold value of 2% is used in this analysis for the minimum support. A low threshold value ensures optimal association between the items. The minimum support value is a parameter that must be given to the program to find all the value that satisfy the certain threshold, All the values above the minimum support is generated as the output.

OUTPUT



Index	support	itemsets
0	0.11316	frozenset({'WHITE HANGING HEART T-LIGHT HOLDER'})
1	0.0869121	frozenset({'JUMBO BAG RED RETROSPOT'})
2	0.0846898	frozenset({'REGENCY CAKESTAND 3 TIER'})
3	0.0780828	frozenset({'ASSORTED COLOUR BIRD ORNAMENT'})
4	0.0775422	frozenset({'PARTY BUNTING'})
5	0.0672713	frozenset({'LUNCH BAG RED RETROSPOT'})
6	0.0604841	frozenset({'SET OF 3 CAKE TINS PANTRY DESIGN'})
7	0.0598234	frozenset({'LUNCH BAG BLACK SKULL.'})
8	0.0567602	frozenset({"PAPER CHAIN KIT 50'S CHRISTMAS"})
9	0.0563397	frozenset({'NATURAL SLATE HEART CHALKBOARD'})
10	0.0557391	frozenset({'HEART OF WICKER SMALL'})
11	0.0545378	frozenset({'SPOTTY BUNTING'})
12	0.0529762	frozenset({'LUNCH BAG CARS BLUE'})
13	0.0524356	frozenset({'LUNCH BAG SPACEBOY DESIGN'})
14	0.0512944	frozenset({'WOODEN PICTURE FRAME WHITE FINISH'})
15	0.0510541	frozenset({'REX CASH+CARRY JUMBO SHOPPER'})

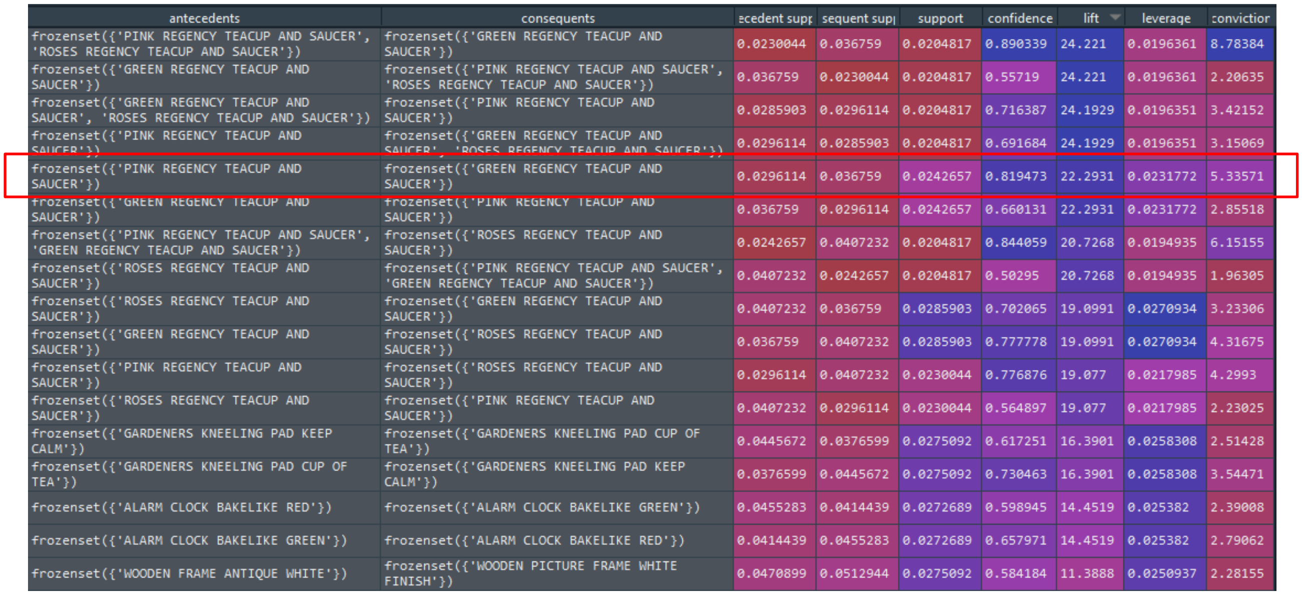
Figure 20 A part of the frequent itemsets

After the implementation of Apriori algorithm to the encoded dataset **236 frequent itemsets** were found for a threshold of 2% of the data. Now we generate the association rules with these 256 frequent items. In the Figure 19 we can see that the product ‘WHITE HANGING HEART T-LIGHT HOLDER’ has a support value of 0.11316, It means that the item is bought 1884 times out of the whole transaction of 16649 transactions containing more than two products.

```
rules1 = association_rules(frq_items, metric ="lift", min_threshold = 1)

rules1 = rules1.sort_values(['lift'], ascending =[False])

rules1.head()
```

Code Snippet 11 Generating Association rules from frequent itemsets

OUTPUT



antecedents	consequents	ecedent supp	sequent sup	support	confidence	lift	leverage	convictior
frozenset({'PINK REGENCY TEACUP AND SAUCER', 'ROSES REGENCY TEACUP AND SAUCER'})	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	0.0230044	0.036759	0.0204817	0.890339	24.221	0.0196361	8.78384
frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	frozenset({'PINK REGENCY TEACUP AND SAUCER', 'ROSES REGENCY TEACUP AND SAUCER'})	0.036759	0.0230044	0.0204817	0.55719	24.221	0.0196361	2.20635
frozenset({'GREEN REGENCY TEACUP AND SAUCER', 'ROSES REGENCY TEACUP AND SAUCER'})	frozenset({'PINK REGENCY TEACUP AND SAUCER'})	0.0285903	0.0296114	0.0204817	0.716387	24.1929	0.0196351	3.42152
frozenset({'PINK REGENCY TEACUP AND SAUCER'})	frozenset({'GREEN REGENCY TEACUP AND SAUCER', 'ROSES REGENCY TEACUP AND SAUCER'})	0.0296114	0.0285903	0.0204817	0.691684	24.1929	0.0196351	3.15069
frozenset({'PINK REGENCY TEACUP AND SAUCER'})	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	0.0296114	0.036759	0.0242657	0.819473	22.2931	0.0231772	5.33571
frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	frozenset({'PINK REGENCY TEACUP AND SAUCER'})	0.036759	0.0296114	0.0242657	0.660131	22.2931	0.0231772	2.85518
frozenset({'PINK REGENCY TEACUP AND SAUCER', 'GREEN REGENCY TEACUP AND SAUCER'})	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	0.0242657	0.0407232	0.0204817	0.844059	20.7268	0.0194935	6.15155
frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	frozenset({'PINK REGENCY TEACUP AND SAUCER', 'GREEN REGENCY TEACUP AND SAUCER'})	0.0407232	0.0242657	0.0204817	0.50295	20.7268	0.0194935	1.96305
frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	0.0407232	0.036759	0.0285903	0.702065	19.0991	0.0270934	3.23306
frozenset({'GREEN REGENCY TEACUP AND SAUCER'})	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	0.036759	0.0407232	0.0285903	0.777778	19.0991	0.0270934	4.31675
frozenset({'PINK REGENCY TEACUP AND SAUCER'})	frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	0.0296114	0.0407232	0.0230044	0.776876	19.077	0.0217985	4.2993
frozenset({'ROSES REGENCY TEACUP AND SAUCER'})	frozenset({'PINK REGENCY TEACUP AND SAUCER'})	0.0407232	0.0296114	0.0230044	0.564897	19.077	0.0217985	2.23025
frozenset({'GARDENERS KNEELING PAD KEEP CALM'})	frozenset({'GARDENERS KNEELING PAD CUP OF TEA'})	0.0445672	0.0376599	0.0275092	0.617251	16.3901	0.0258308	2.51428
frozenset({'GARDENERS KNEELING PAD CUP OF TEA'})	frozenset({'GARDENERS KNEELING PAD KEEP CALM'})	0.0376599	0.0445672	0.0275092	0.730463	16.3901	0.0258308	3.54471
frozenset({'ALARM CLOCK BAKELIKE RED'})	frozenset({'ALARM CLOCK BAKELIKE GREEN'})	0.0455283	0.0414439	0.0272689	0.598945	14.4519	0.025382	2.39008
frozenset({'ALARM CLOCK BAKELIKE GREEN'})	frozenset({'ALARM CLOCK BAKELIKE RED'})	0.0414439	0.0455283	0.0272689	0.657971	14.4519	0.025382	2.79062
frozenset({'WOODEN FRAME ANTIQUE WHITE'})	frozenset({'WOODEN PICTURE FRAME WHITE FINISH'})	0.0470899	0.0512944	0.0275092	0.584184	11.3888	0.0250937	2.28155

Figure 21 A part of generated Association rules from frequent itemsets

76 Rules were generated from the frequent itemsets which were sorted in descending order of the lift values. A portion of the generated rules are presented in Figure 21.

The basic ideology of Association rule is an ‘If’ ‘then’ relationship. Which means ‘**If’** product A is bought ‘**Then’** there are chances of product B being bought together. The **IF** component of an association rule is known as the antecedent, the **THEN** component is known as the consequent which is shown in the Figure 21.

From the results obtained, we could see from the highlighted part that “PINK REGENCY TEACUP AND SAUCER” and “GREEN REGENCY TEACUP AND SAUCER” has a higher association between each other since these two items have a higher “lift” value. The higher the lift value, the higher the association between the items. Normally If the lift value is more than 1, those two items are associated well with each other and has the chances of sold together.

Now that we have generated the association rules we have to proceed by using this data to arrange the warehouse structure and compare the results.

6. Simulation and Results discussion

A Simulation is the imitation of the operation of an actual real-world process or system over time. One can use mathematical or computational models in simulations to modify any system to study how it works, or what happens when parts of it are changed. The simulations are generally utilized in various contexts related to engineering, supply chain, testing, education, Logistics and much more. In the case of simulation of a warehouse an advanced method of simulation called the Discrete event simulation are usually used.

Discrete event simulation (DES) is a type of simulation that considers a system as a discrete collection of events, with each event having a defined effect on the rest of the system. The individual processes that make up a system can each be defined in terms of their impact on the system, their resource requirements, and their trigger (the processes may be scheduled or they might occur at random, or they might occur in response to another system event). After these constituent parts have been determined, it can be combined within the simulation to recreate the system from the ground up. The main advantage of the DES is its speed and configurability. [31]

We conduct two types of simulation in this work to conclude the performance of the two different systems (the traditional Shuttle based storage and retrieval system (SBS/RS) and the new proposed system by Eurofork S.p.A.). The first simulation is based on a simulator developed by the research team in which we find out how well the system performs by two conditions, i.e. by application of the generated association rules and without application of association rules to arrange the warehouse in the rack. The cycle times and energy consumption are calculated and compared.

The second simulation is implemented using a simulation software called Flexsim, in which we create a virtual environment of two scenarios to be compared in 3D (without using the association rules, instead creating two scenarios that can replicate the traditional system comparison with the new system with random SKU placement in the racks) and run it to get the cycle times of both systems. Then we utilize the two simulation results and compare them to get the final results.

6.1. Simulation I

The simulator that we utilize for the purpose of the simulation 1 was developed by the research team and we do not go in depth on the details about the construction and working methodology of the simulator which is not the scope of this thesis. Instead simply we give the Input to the simulator and we generate the necessary outputs and then compare its performance. The simulator is developed using the python programming language using the SymPy framework, and contains the following extension packages,

- IdeaSim
- SimMain
- Resources
- Task
- Trace

SimPy is a process-based discrete-event simulation framework based on standard Python programming language. The processes in SimPy are characterized by Python generator functions and can be used to model active components like vehicles, customers, or agents. SimPy also gives various types of shared resources to model limited capacity congestion points (like servers, checkout counters and tunnels). The simulations in SimPy can be performed “as fast as possible”, in real time (wall clock time) or by manually stepping through the events. [32]

For running of the simulator a few parameters have to be calculated from the UK dataset. The parameters are

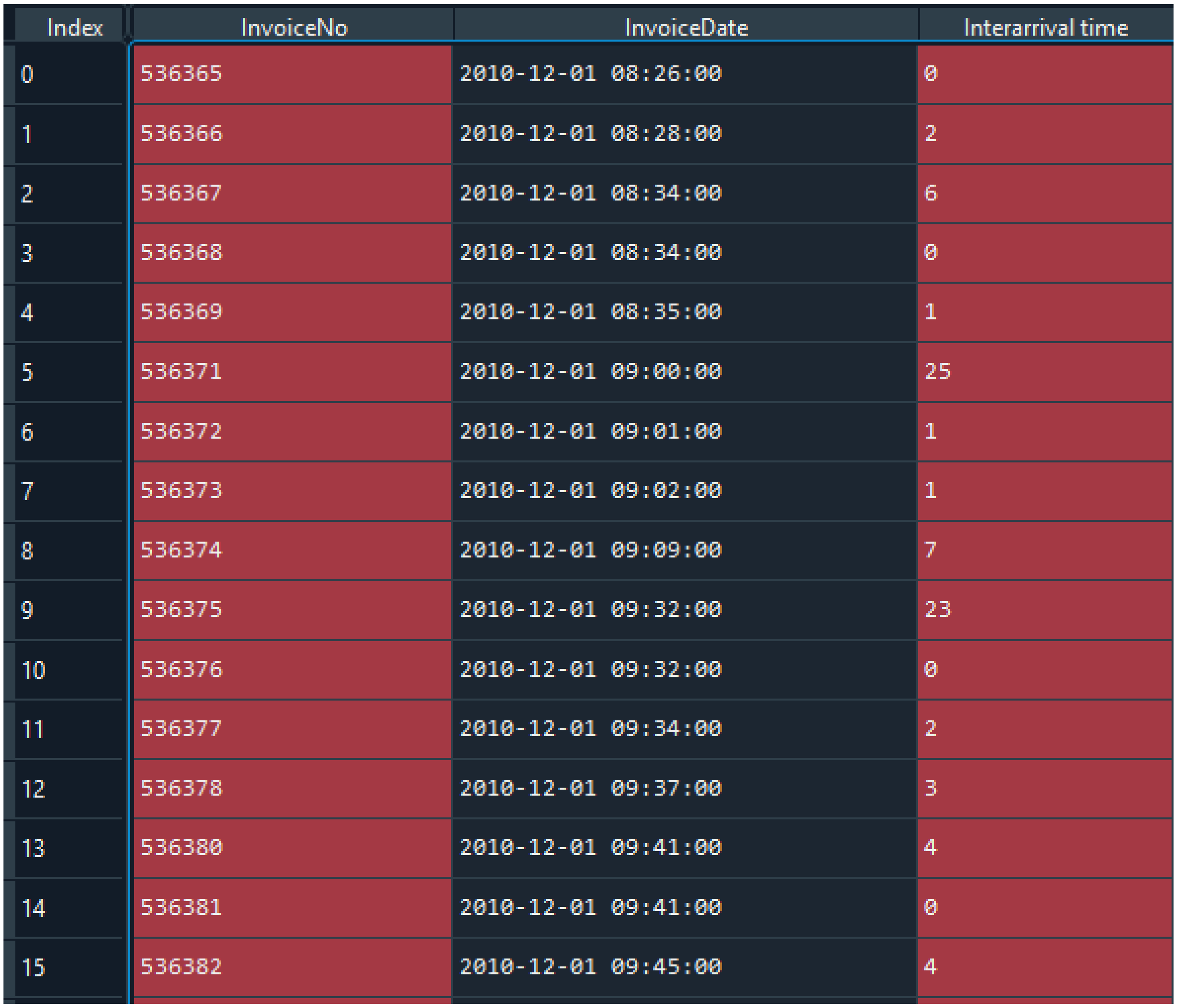
1. Interarrival time between orders,
2. Weight of rules generated in association rule mining.

To simplify, the time difference between arrival of one order and then the next order is referred to as Interarrival time. For the calculation of interarrival times between orders python program is used once again. A loop is run to return the output as the difference between each order arrival time using the Invoice date and time for each unique transaction.

```
                                 """Parsing Date format"""

dataset_UK['InvoiceDate'] = pd.to_datetime(dataset_UK['InvoiceDate'], format = "%m/%d/%Y %
H:%M")



                           """Inter arrival time calculation"""

dataset_interarrival = pd.DataFrame([], columns=['InvoiceNo','InvoiceDate','Interarrival t
ime'])

for invoice in dataset_UK.InvoiceNo.unique():

    if dataset_UK[dataset_UK['InvoiceNo']==invoice].index[0] != 0:

        data = []

        timedelta = dataset_UK.loc[dataset_UK[dataset_UK['InvoiceNo'] == invoice].index[0]
]['InvoiceDate']

        time_diff = (timedelta - dataset_interarrival['InvoiceDate'][dataset_interarrival.
shape[0] - 1]).total_seconds() / 60

        data.append(invoice)

        data.append(timedelta)

        data.append(time_diff)

        dataset_interarrival.loc[dataset_interarrival.shape[0]] = data

    else:

        data = []

        data.append(invoice)

        data.append(dataset_UK[dataset_UK['InvoiceNo'] == invoice]['InvoiceDate'][

                        dataset_UK[dataset_UK['InvoiceNo'] == invoice].index[0]])

        data.append(0)

        dataset_interarrival.loc[dataset_interarrival.shape[0]] = data
```

Code Snippet 12 Calculation of order interarrival time

OUTPUT



Index	InvoiceNo	InvoiceDate	Interarrival time
0	536365	2010-12-01 08:26:00	0
1	536366	2010-12-01 08:28:00	2
2	536367	2010-12-01 08:34:00	6
3	536368	2010-12-01 08:34:00	0
4	536369	2010-12-01 08:35:00	1
5	536371	2010-12-01 09:00:00	25
6	536372	2010-12-01 09:01:00	1
7	536373	2010-12-01 09:02:00	1
8	536374	2010-12-01 09:09:00	7
9	536375	2010-12-01 09:32:00	23
10	536376	2010-12-01 09:32:00	0
11	536377	2010-12-01 09:34:00	2
12	536378	2010-12-01 09:37:00	3
13	536380	2010-12-01 09:41:00	4
14	536381	2010-12-01 09:41:00	0
15	536382	2010-12-01 09:45:00	4

Figure 22 A part of order interarrival time output

The Order interarrival time is obtained as minutes. In the program code if the condition of the loop is satisfied then it calculates the Order interarrival time and appends it to the relevant dataset row. The lowest value obtained for interarrival between orders are 0 minutes i.e. the order is received immediately one after another and the maximum value is 11.74 days (16914 minutes).

The weight of the rule generated is simply taken from the Lift column of the results of Association rules. A part of the values can be viewed from the Figure 21.

As a part of the simulation the following values are assumed as the input to the simulator, the values were assumed based on replicating a traditional AS/RS.

Efficiency	0.9
Friction parameter	Fr = 1.15
Number of satellites per shuttle, put in the same section as the shuttle. Total number of satellites $N_{l}i * N_{s}h * N_{s}at$	Nsa = 2
Number of shuttles per lift, put in the same section of the lift. Total number of shuttles is $N_{l}i * N_{s}h$	Nsh = 3
Number of lifts	Nli = 3
Max velocity of the satellite (ms-1)	Vz = 1.2
Acceleration of the satellite (ms-2)	Az = 0.7
Max velocity of the shuttle (ms-1)	Vy = 0.9
Acceleration of the shuttle (ms-2)	Ay = 0.8
Max velocity of the lift (ms^{-1})	Vx = 4
Acceleration of the lift (ms^{-2})	Ax = 0.8
Weight of the lift in kg	Wli = 1850
Weight of the shuttle in kg	Wsh = 850
Weight of the lift in kg	Wsa = 350
Dimension of unit in z (satellite or fork) movement direction.	Lz = 1.2
Dimension of channel in the x (shuttle movement) direction (meter).	Ly = 1.5
Floor height in meter	Lx = 1
Width of the warehouse in number of units.	Nz = 50
Number of channels layer per floor.	Ny = 10
Number of floors	Nx = 50
The level at which the bay is placed. Can be between floor if a decimal is passed.	bay level = 0
The strategy used to select the channel where to put/take a pallet. A value of 0 means random, a value of 1 means nearest available channel with weighted manhattan distance and 2 emptier channels.	strategy = 1
Weighted manhattan distance for x distance	strategy par x = 1
Weighted manhattan distance for y distance	strategy par y = 1

Table 3 Values applied in the simulation I

The simulator code has been kept confidential by the research team for the purpose of privacy and future development. The simulator was run replicating the processes of a shuttle based automated storage and retrieval system. The obtained 76 association rules were utilized to arrange the warehouse structure and only products that occurs at least 500 times out of the whole transaction is considered to simplify the simulation (the products that occurred at least 500 times were 200 products out of 4070 products in the dataset). Another purpose of choosing this strategy to simplify the simulation also gives us another advantage of using only the most sold products in the dataset which is more profitable to the company. As the purpose of this simulator 1 is solely based on comparing performance of the AS/RS with and without using the association rules and not on rainbow pallet building. So two iterations were run for this purpose, the first one had the products arranged in the warehouse randomly using the random SKU placement strategy and the second iteration included the association rules to arrange the warehouse. Second scenario uses the association rules with highest lift values to place the products together in the racks. The simulator was run for both the scenarios i.e. by using the association rule and without using the association rules with the data from Table 3. According to the simulator one product is picked at a time and multiple picking is done in order to complete a single order. The data were run and analyzed in the research lab and the results were obtained. After the simulation was completed, the following graph was obtained that shows the energy consumption and cycle times of the shuttle based automated storage and retrieval system.

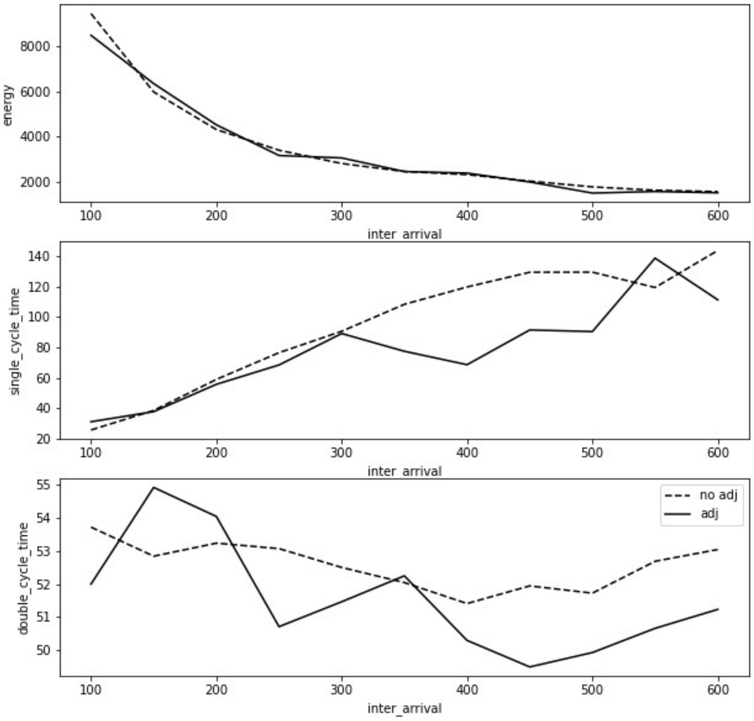


Figure 23 Results from simulation I

From the Figure 23 we can see that the correlation between with using and without using association rules, the dotted line represents the simulation done without using the association rules and the continuous line represent the simulation using association rules , the energy consumption curve almost remains the same, although having slight variations, they reach the same line, since energy consumption problem is not the main issue of our work, we focus on cycle time. Therefore on carefully observing the results in Figure 23 in terms of Cycle time we see a considerable variation.

So what we conclude with the results of simulation I are, using the association rules the cycle time is reduced to some extent, so when designing an actual warehouse, the products should be arranged according to the association rules, i.e. items with higher association must be placed nearby in the racks and also most bought products can be placed in an area nearer to the rack entrance so that the cycle time can be further reduced.

6.2. Simulation II

This simulation is accomplished on the Flexsim simulation software that models the virtual environment of a warehouse in 3D to carry out the performance tests. A simple layout of the warehouse that utilizes a Shuttle based storage and retrieval system (SBS/RS) is designed using various tools available within the software. This warehouse houses an SBS/RS, the racks where the items are picked, the queue area for the arrival of goods and the picked products storage area. Here in the simulation a traditional AS/RS is considered to be the SBS/RS because we do not still have the software that can build the new proposed system by Eurofork.

In FlexSim, we can create 3D models of the system that we want to simulate, or we can create purely theoretical models using the Process Flow tool alone. The process flow tool is a powerful logical flowcharting tool that can build systems solely based on flowcharts. we might find it beneficial to build theoretical models using the Process Flow tool in the earlier stages of model building, especially while we are still determining the scope and purpose of the model. However, we should consider building a 3D model at some point during the process for a few important reasons such as Model validation, Ease of use and better understanding [33]. In this work we however use both a combination of 3D module and the process flow tools. We first see how we design the warehouse,

In this simulation II, we consider two scenarios to replicate the old system (i.e. the traditional method of order picking) and new system (i.e. rainbow pallet building in the racks), these methods are named scenario 1 and scenario 2, respectively. Both these scenarios do not use the association rules to arrange the products in the warehouse, instead the scenarios replicate the traditional system comparison with the new system with random SKU placement in the racks. In scenario 1, in order to create a rainbow pallet the whole pallets are retrieved to the workstation and then required number of products are picked by a robotic arm to create a new combined order pallet or rainbow pallet. Then each pallet is returned back to the original location in the racks. In scenario 2, the picking of the order is entirely fulfilled by the system within the racks i.e. creating a rainbow pallet in the warehouse rack structure. We compute the time of both the scenarios and discuss the results to find which system is better.

6.2.1. Scenario 1

To initiate the scenario 1 we design a simple 3d module of the picking station and we connect this picking station with the traditional AS/RS system with racks, which bring the whole pallet to the picking area and takes it back. The general layout of picking area is shown in Figure 25 and the AS/RS structure and rack structure is very similar to the one shown in Figure 26.

In the next step we create a process flow for the Random order generator. The random order generator is a generator that creates random orders for the warehouse that can be fulfilled using the AS/RS. This flow contains the random order seed first in which we give the time as 100 seconds so that the order arrival starts after 100 seconds. Then the flow passes to order number assignment and the order arrival delay and other parameters on specifying a new order.

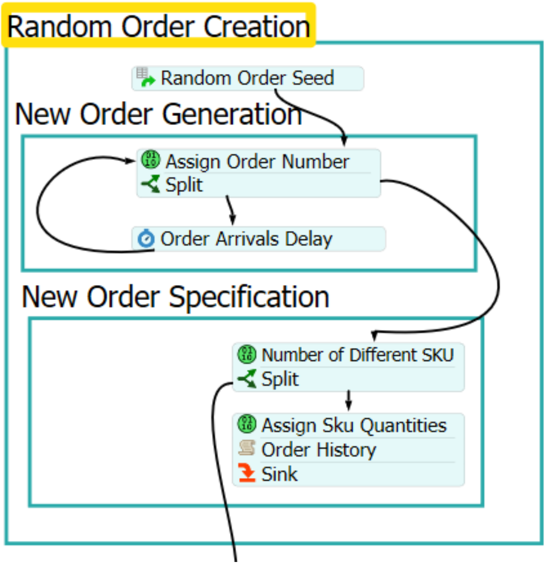


Figure 24 Process flow for Random order generator

There are 20 different types of product in the warehouse which implies top 20 sold products. The order dimension is set to pick 3 different products of 3 quantity each to complete one mixed pallet and the order interarrival time is set to 50 seconds. The speed of the AS/RS and the picking robots are set to default as 1m/s. The simulation is run and immediately the random order generator starts creating orders and sends it to the AS/RS and picking robots, and the cycle time readings are taken.

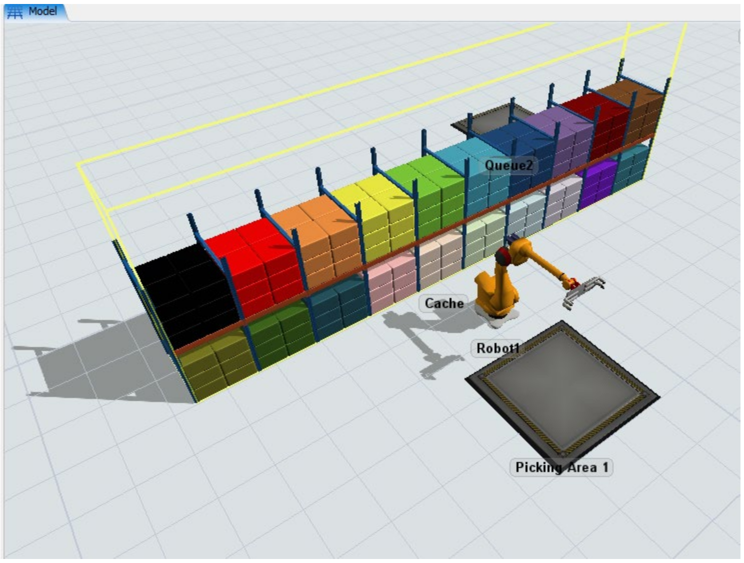


Figure 25 General layout of picking station for scenario 1

The picking station is assumed to be located close to the racks for the simplicity of simulation, so the average travel time from the racks to the picking station one way is 30 seconds. The picking starts only after all the three whole pallets are brought to the picking station.

From the simulation results,

The average time to build a pallet in the picking area = 162.08 seconds,

but without considering the time for the AS/RS to bring the pallets to the picking area.

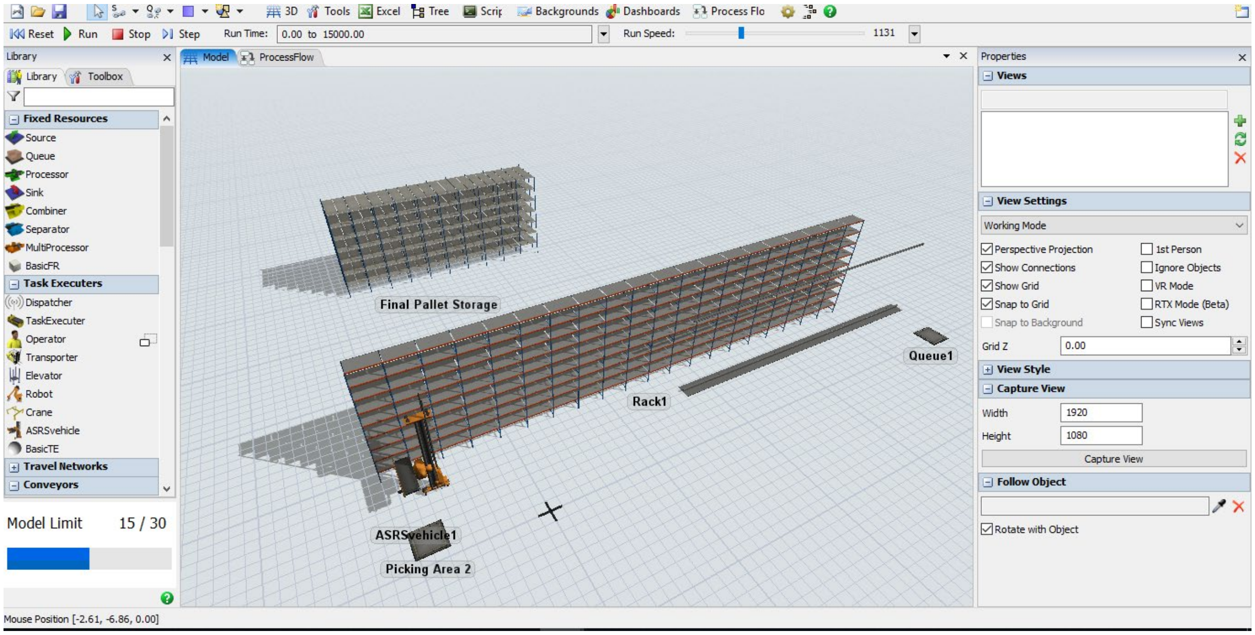
Average time for picking and delivering the pallet to the picking station by AS/RS = 400 seconds

Total cycle time is given by summing up the average time spent in picking area (162.08 seconds) and the average time for picking and delivering the pallet to the picking station by AS/RS (400 seconds) which gives the result as 562.08 seconds (rounding off to 562 seconds).

Average cycle time for picking an order in scenario 1 in the simulation II = 562 seconds or 9.36 minutes

6.2.2. Scenario 2

To model the scenario 2, initially we add the 3D modules required from the library of the flexsim, then we connect the objects with each other to make it a complete system. Then we use the process flow tool to inject the data into each individual system to carry out the whole process of picking. The 3D module created for scenario 2 is presented in Figure 26.



Run Time: 0.00 to 15000.00

Run Speed: 1131

Model Limit 15 / 30

Working Mode

☑ Perspective Projection
☑ Show Connections
☑ Show Grid
☑ Snap to Grid
□ Snap to Background
□ 1st Person
□ Ignore Objects
□ VR Mode
□ RTX Mode (Beta)
□ Sync Views

Grid Z 0.00

Width 1920

Height 1080

☑ Rotate with Object

Mouse Position [-2.61, -6.86, 0.00]

Figure 26 General layout of the warehouse structure for scenario 2

After the layout has been created, we now model the process flow. The first process we have to initiate is placement of products in the racks, we consider only 20 different products i.e. we arrange only the top 20 sold products from our data. But here we do not represent any names for our products as it is not necessary for the scope of this simulation. Now we create the process flow for the arrival of goods which is presented in Figure 27.

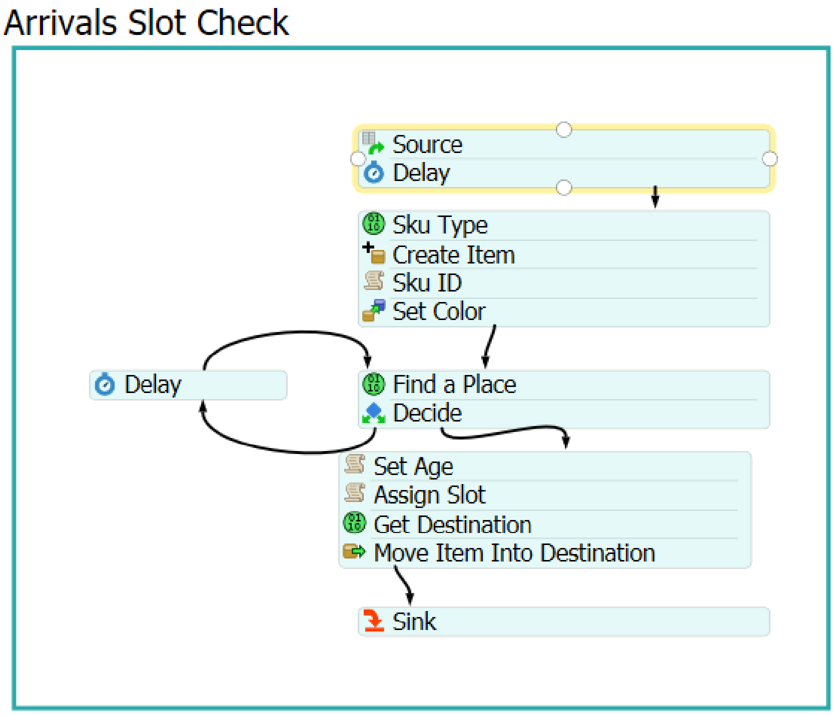


Figure 27 Process flow for products arrival

In the source we represent how many number of products that needs to be placed in the rack at the beginning of the process and then other steps include creating a SKU(store keeping unit), assigning the Id and color, then finding a place in the rack to place the SKU and other related parameters.

The next process flow we generate is for the Random order generator. For this step, the flow mentioned in Figure 24 is implemented. The random order generator is a generator that creates random orders for the warehouse that can be fulfilled using the AS/RS. This flow contains the random order seed first in which we give the time as 100 seconds so that the order arrival starts after 100 seconds. Then the flow passes to order number assignment and the order arrival delay and other parameters on specifying a new order. The next flow we must create is for the Order picking of the generated orders. For this we follow the steps mentioned in Figure 28.

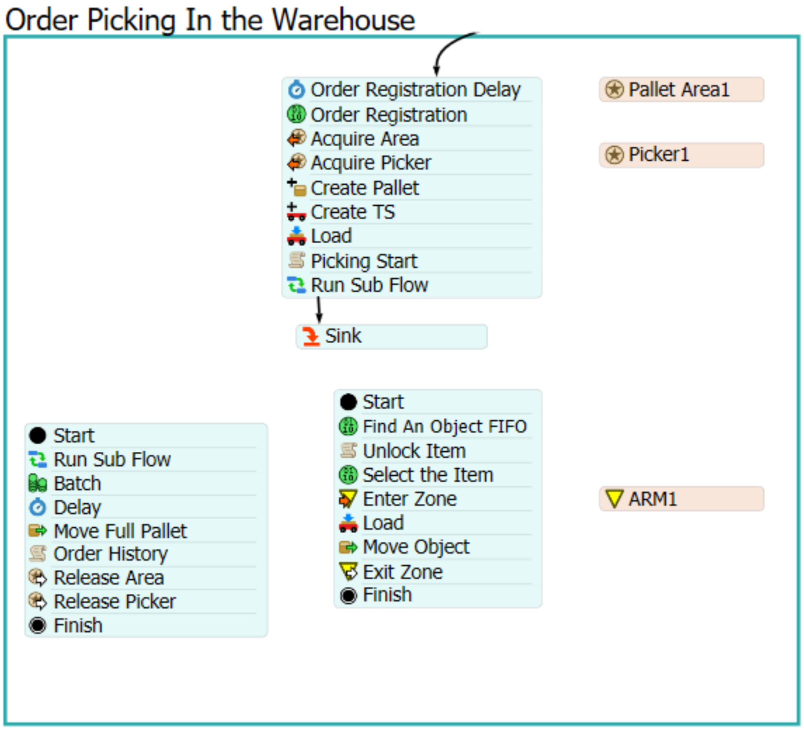


Figure 28 Process flow for order picking

This process flow for the order picking includes the registration of generated order, creation of pallet, loading the pallet and then picking the SKUs present in the order. After the order picking is complete, the pallet is delivered to a pre-assigned location and the next order picking is started.

The parameters that we use for the order generation and picking are given in the table below,

No_Dif_Sku_Max	Max range of number of different objects in a single order.	3
No_Sku_Max	Max range of order dimension	3
Ord_Interarr_Sec	Order Interarrival time in seconds.	50
No_Dif_Sku_Min	Min range of number of different objects in a single order.	3
No_Sku_Min	Min range of order dimension	3

Table 4 Parameters for order generation and picking

As per the table the reason that we have set the max and min values to be equal because we want to create equal dimension of orders for the reason of comparison of cycle times for this work. As the ultimate aim of this work is to compare the performance of this novel system with the traditional system. Since we have placed the SKUs at random positions in the warehouse we perform three iterations and calculate the average cycle time from the results.

The simulation was run to pick three different products of the three quantity each. The products were mostly placed nearby so that the AS/RS does not move farther away in the racks.

The simulation was run for completing 5 orders and the interarrival time between orders were set to 50 seconds as mentioned in Table 4. The simulation time limit was set to 900 seconds for each iteration, however the simulation start time was set at 100 seconds, which means the system is idle for the first 100 seconds and just collects the orders, then it starts picking the SKUs. The speed of movement of the AS/RS was set to default as 1 m/s.

The results of the three iterations of the simulation are presented in Table 5, Table 6 and Table 7.

Iteration 1

Order number	Start time of order in seconds	Cycle time per order in seconds
1	116.12	123.76
2	239.88	174.63
3	414.51	174.62
4	589.13	95.5
5	684.63	157.67
6	842.30	-

Table 5 Iteration 1 for cycle time calculation

Iteration 2

Order number	Start time of order in seconds	Cycle time per order in seconds
1	116.12	123.76
2	239.88	168.98
3	408.86	174.62
4	583.48	123.76
5	707.24	146.37
6	853.61	-

Table 6 Iteration 2 for cycle time calculation

Iteration 3

Order number	Start time of order in seconds	Cycle time per order in seconds
1	116.12	157.67
2	273.79	128.34
3	402.13	163.32
4	565.45	197.24
5	762.69	83.58
6	846.27	-

Table 7 Iteration 3 for cycle time calculation

Average cycle time for Iteration 1 = 145.236 seconds

Average cycle time for Iteration 2 = 147.498 seconds

Average cycle time for Iteration 2 = 146.03 seconds

Therefore the Average for the averages of all the three iterations are taken and we obtain the average cycle time for scenario 1. we round off the average cycle time obtained from 146.25 seconds to 146 seconds.

Average cycle time for picking an order in scenario 2 in the simulation II = *146 seconds* or 2.43 minutes.

Results of Scenario 1 and scenario 2 are given in the table below,

	Average cycle time to complete an order
Scenario 1	562 seconds or 9.36 minutes
Scenario 2	146 seconds or 2.43 minutes.

Table 8 Results of Scenario 1 and 2 from Simulation II

From the above table we can compare the cycle times of two different scenarios, As expected we obtained a positive result. The scenario 2 of the new system shows a much-improved speed compared to the traditional picking scenario 1. Through implementing this scenario 2 in the Automated warehouses we improve the cycle time and overall material handling time to a great extent.

6.3. Results discussion

Therefore having the result of Simulation I and Simulation II as evidence we can strongly point out that the new system proposed by Eurofork S.p.A outperforms the traditional SBS/RS in terms of cycle time for completing an order. The combination of these two simulations may help the company to reach further stage in development of the product.

6.3.1. Discussion on results of Simulation I

The graphs from the Figure 23 presents that the correlation between with using and without using association rules, the energy consumption curve almost remains the same, although having slight variations, they reach the same line. But in terms of cycle time we see a considerable difference.

So what we conclude with the results of simulation I are, using the association rules the cycle time is reduced to some extent, so when designing an actual warehouse, the products should be arranged according to the association rules, i.e. items with higher association must be placed nearby in the racks and also most bought products can be placed in an area nearer to the rack entrance so that the cycle time can be further reduced.

6.3.2. Discussion on results of Simulation II

The results from the Table 8 compares the cycle times of two different scenarios, As anticipated we see an improved cycle time reduction in the new system. The scenario 2 of the new system shows a much-improved speed compared to the traditional picking scenario 1. The speed has quadrupled from the scenario 2 order completion time being 2.43 minutes and the scenario 1 order completion time 9.36 minutes. Therefore implementing new system in the shuttle based automated warehouses improves the cycle time and overall material handling time to a great extent bringing a new revolution in the logistic industry.

7. Conclusion and Future aspects

Our work has led us to an obvious conclusion that the new proposed system by Eurofork shows increased performance in terms of cycle time compared to the traditional SBS/RS picking technique. The results of the simulation I and simulation II can be referred from Figure 23 and Table 8 respectively to prove this fact.

Overall this study being focused on the performance evaluation of an SBS/RS system also clearly covers a wide range of topics related to the problem.

1. This work researched various past literatures based on the topic, understood its inventions and limitations, and utilized important techniques opt for the work.
2. This work explored into the diverse ways of adopting the cleansing and visualization of dataset, to optimize the simulation of the model.
3. This work investigated on various mining techniques and chose the one that could give appropriate and accurate results.
4. This work also covered basic understanding of the python programming language in the field of data mining and market basket analysis.
5. Finally this work briefly described the usage of simulation software Flexsim which created excellent 3D warehouse simulations to give us the results expected to complete this study.

Considerable progress has been made in the comparison of the old technique with the Eurofork proposed technique, however it is plausible that a number of limitations may include in our work and our work can be taken as a primary idea to perform the future research.

The limitations and future aspects are mentioned below, and we propose that further research should be undertaken in the following areas,

1. The energy consumption calculation is not well focused in this work since we are more aimed at cycle time. Further experimental investigations are needed to estimate the energy consumption of the traditional and new systems.
2. The simulations are done in two different interfaces i.e. python and Flexsim to calculate the performance parameters individually, we could not still implement the association rules obtained into a single 3D simulator that gives cycle time, energy consumption and other performance parameters. Research on advanced techniques can be done on this point.
3. Another limitation in the field of 3D simulation is that the Flexsim simulation software does not have the ability to replicate an actual SBS/RS operation, instead an AS/RS is assumed to be an SBS/RS in this study and the shuttle travel is limited only focusing on building a rainbow pallet with an single rack in the warehouse. The system is simplified to obtain the results, which can be a vital issue for future research.

References

[1] C. C. Murray and H.-. Y. Lee, "Robotics in order picking: evaluating warehouse layouts for pick, place, and transport vehicle routing systems," *International Journal of Production Research,* 2018.

[2] F. A. R. G. and F. , "Intralogistics and industry 4.0: designing a novel shuttle with picking system," in *29th International Conference on Flexible Automation and Intelligent Manufacturing*, Limerick, Ireland, 2019.

[3] Y. Ma and J. Wang, "Travel time analysis for shuttle-based storage and retrieval system with middle input/output location," 2019.

[4] Y. Wu, C. Zhou, W. Ma and X. . T. R. Kong, "Modelling and design for a shuttle-based storage and retrieval system," *International Journal of Production,* 2020.

[5] K. Azadeh, R. D. Koster and D. Roy, "Robotized and Automated Warehouse Systems: Review and Recent Developments," *Transportation Science 53(4):917-945.,* 2019.

[6] M. Bevilacqua, F. E. Ciarapica and S. Antomarioni, "Lean principles for organizing items in an automated storage and retrieval system: An Association rule mining – based approach," *Management and Production Engineering Review,* 2019.

[7] T. Osada, "The 5S's: five keys to a total quality environment," *Asian Productivity Organization (1993)..*

[8] Y. L. H. a. L. Y. Chuang, "Item-associated cluster assignment model on storage allocation problems.," *Computers & industrial engineering,* 2012.

[9] Z. Chen, W. Song, C. Du and L. Liu, "Research on Warehouse Management System Based on Association Rules," in *6th International Conference on Computer Science and Network Technology (ICCSNT)*, 2017.

[10] K. J. Roodbergen and I. F. Vis, "A survey of literature on automated storage and retrieval systems," *European Journal of Operational Research,* 2009.

[11] E. Romaine, "Automated Storage & Retrieval System (AS/RS) Types & Uses," 18 August 2020. [Online]. Available: https://www.conveyco.com/automated-storage-and-retrieval-types/.

[12] JavaTpoint, "Data Mining Techniques," [Online]. Available: https://www.javatpoint.com/data-mining-techniques.

[13] A. Trevino, "Introduction to K-means Clustering," 2016. [Online]. Available: https://blogs.oracle.com/datascience/introduction-to-k-means-clustering.

[14] "A Tutorial on Clustering Algorithms," [Online]. Available: https://matteucci.faculty.polimi.it/Clustering/tutorial_html/kmeans.html.

[15] N. S. Chauhan, "Hierarchical Clustering," 2019. [Online]. Available: https://www.kdnuggets.com/2019/09/hierarchical-clustering.html.

[16] S. Glen, "Hierarchical Clustering / Dendrogram: Simple Definition, Examples," [Online]. Available: https://www.statisticshowto.com/hierarchical-clustering/.

[17] A. R. Pillai and D. A. Jolhe, "Market Basket Analysis: Case Study," in *International Conference on Advances in Mechanical Engineering, ICAME 2020*, Nagpur, India, 2020.

[18] R. I. T. a. S. A. Agrawal, "Mining association rules between sets of items in large databases.," *Proceedings of the 1993 ACM SIGMOD international conference on Management of data.*

[19] Y. S. Z. T. L. Y. J. Q. X. D. Dongwen Zhang, "A genetic-algorithm based method for storage location assignments in mobile rack warehouses," in *IEEE Global Communications Conference*, 2019.

[20] Python.org. [Online]. Available: https://www.python.org/doc/essays/blurb/.

[21] S. Mindfire, "Mindfire Solutions," 2017. [Online]. Available: https://medium.com/@mindfiresolutions.usa/python-7-important-reasons-why-you-should-use-python-5801a98a0d0b.

[22] "Codecademy," [Online]. Available: https://www.codecademy.com/articles/what-is-an-ide.

[23] "Spyder," Spdyer, [Online]. Available: https://www.spyder-ide.org/.

[24] M. Hasan, "Ubuntupit," [Online]. Available: https://www.ubuntupit.com/best-python-libraries-and-packages-for-beginners/.

[25] "pypi Seaborn," [Online]. Available: https://pypi.org/project/seaborn/.

[26] S. Raschka, "MLxtend: Providing machine learning and data science utilities and extensions to Python’s scientific computing stack," *The Journal of Open Source Software (JOSS),* 2018.

[27] "Flexsim," [Online]. Available: http://www.flexcon.it/products/flexsim-en/.

[28] D. D. Chen, "UCI Machine learning repository - Online Retail Data Set," [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Online+Retail#.

[29] P. Pandey, "Data Preprocessing : Concepts," 25 November 2019. [Online]. Available: https://towardsdatascience.com/data-preprocessing-concepts-fa946d11c825.

[30] "Data visualization beginner's guide: a definition, examples, and learning resources," [Online]. Available: https://www.tableau.com/learn/articles/data-visualization#:~:text=Data%20visualization%20is%20the%20graphical,outliers%2C%20and%20patterns%20in%20data..

[31] "Discrete event simulation," [Online]. Available: https://www.bmt.org/industries/defence-and-security/discrete-event-simulation/.

[32] "simpy," [Online]. Available: https://pypi.org/project/simpy/.

[33] "FlexSim Manual," [Online]. Available: https://docs.flexsim.com/en/21.0/Introduction/Welcome/.

[34] M. H. C. C. K. a. W. H. Chen, "Aggregation of orders in distribution centers using data mining.," *Expert systems with applications,* 2005.

[35] A. Beklemysheva, "Steel kiwi," [Online]. Available: https://steelkiwi.com/blog/python-for-ai-and-machine-learning/..