Implementation of a Recommendation System for a retail store based on basket analysis

Amirhossein Layegh Kheirabadi

Final Masters Project 2018/2019
School of Electronic Engineering and Computer Science
Queen Mary, University of London

Supervisor:
Dr Arman Khouzani

Masters in Big Data Science
London, August 2019

Contents

Implementation of a recommendation system for a retail store based on basket analysis

Amirhossein Layegh Kheirabadi

Abstract:

Online shopping grows rapidly. Great competition among online store owners causes the rise of business tools that are reshaping the world of E-commerce. Decision making and understanding the behavior of customers and similarity among products have become a challenging issue for organizations. Data mining is no longer a tool in the shed rather a main asset for analyzing huge amount of data in making correct decisions. One of these powerful tools is Recommendation System. A recommender system learns customer's behaviour and recommends products that they will find most attractive among the available products. This project used Apriori algorithm and K-Nearest Neighbor to implement a recommender system for the purpose of mining association rules on a real-world dataset obtained from Koolbitz Company. Both models brought about almost same results. Unlike Apriori, K-NN is robust with any newly added data but a little slower. 186 association rules were acquired, most of the items in each rule belong to the same category and 22% of the rules included cosmetic items.

Experimental analysis has been done prior to implementing Recommendation System in order to have a better understanding about our data. As some consequences we figured the reason of fluctuation in their sale, fetch the list of products with highest sale, the relation between the price and amount of sale and so on.

Chapter1: Introduction

1.1 Introduction

Data analytics has become an extremely important and challenging issue in various fields like computer science, biology, medicine, finance, and homeland security. Since massive amount of data is available for analysis, scalable integration techniques has become important. First, large volumes of data must be imported and stored relying on cleansing and filtering techniques. Next, sophisticated algorithms are used to analyze the data and extract "useful" information and Finally, various user interfaces can be used to visualize and understand the data.

Due to the exponential increase in the amount of data available, analyzing in this scale is definitely very demanding.

For instance, social networks like Facebook generate terabytes of data every day in the form of photos, videos, wall posts, etc., and will generate significantly larger amount of data in the near future. The size of data that we have available to date, is unprecedented and cannot simply be analyzed with conventional data management techniques.

Coobitz is an online retail store, Located in London,that has cooperation with eBay and Amazon to sell their products. For the purpose of this analytic project, Koobitz has provided access to its data .

1.2 Motivation

The growth in online sales of products (the share of online stores in the global retail from 2015 to 2021 can be seen in Figure 1) (Fatoni, Utami and Wibowo, 2018) has amounted to development of the technologies that makes the business-customer interaction easier. The main concern for businesses is how to create this interface simpler for the buyers to find/seek what they are looking for.

As a result of the growth of digital economy and also heavy competition among the businesses, ever increasing online stores not only use this interface to display products to their online visitors but also they track their fashion of product selection to better attract their engagement. The high number of product variations, which is offered to customers, may cause confusion for them. Thereby, the employment of a product recommendation system is suggested.

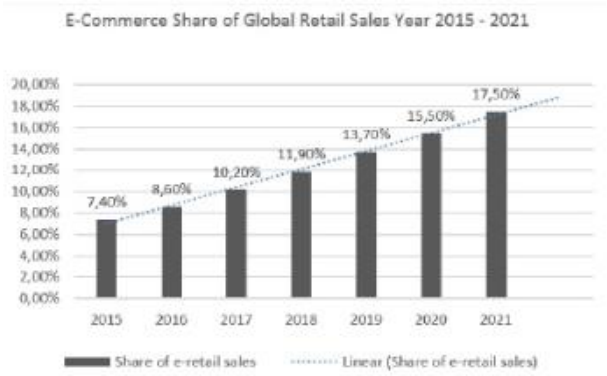


Fig1 Total E-commerce Global Retail Sales

1.3 Project Aims

The aim of this project is to use a real market dataset that is provided by Koolbitz on their sale, transactions, customer information, channel listing, to gain better insight about their system. The data that is used in the project is collected from eight separate datasets. Five of them are for their eBay accounts and the rest belong to their 3 Amazon accounts. This helps the business owner to make better decisions by drawing many insights on their retail business. Prior to any modelling, a great amount of attention was devoted to data exploration. This exploration is crucial for identifying the underlying trends and observe remarkable correlations. In addition, Apriori

algorithm is employed to implement the recommendation systems. The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules.

1.4 Literature Review

Picture a word without data storage; a place where every detail about a person or organization, every transaction performed, or every aspect which can be documented is lost directly after use. Organizations would thus lose the ability to extract valuable information and knowledge, perform detailed analyses, as well as provide new opportunities and advantages. Anything ranging from customer names and addresses, to products available, to purchases made, to employees hired, etc. has become essential for day-to-day continuity. Data is the building block upon which any organization thrives.

Being able to efficiently "make sense" out of big data is becoming even more important than ever in various areas. E-commerce space is extremely competitive and current day consumers want personalized experiences, easy to access product information and exceptional services. In order to deal with high expectations from consumers, online retailers need to leverage data being collected and actively integrate analytics to improve their decision making process. This is true for any retailer. They should use analytics to better understand consumer preferences, and further provide them with the right product offerings. Also, online retailers should extract value from their data beyond just information gathering and monitoring.(Schafer, Konstan and Riedi, 1999)

Here we have mining the required integrated, cleaned, trustworthy data and also running queries, modeling and building algorithms to find new insights.

Chapter 2: Related Works

2.1 Big Data Analytics Infrastructure

The following Fig 2 shows different layers occurs in the big data analytics. (Praveena and

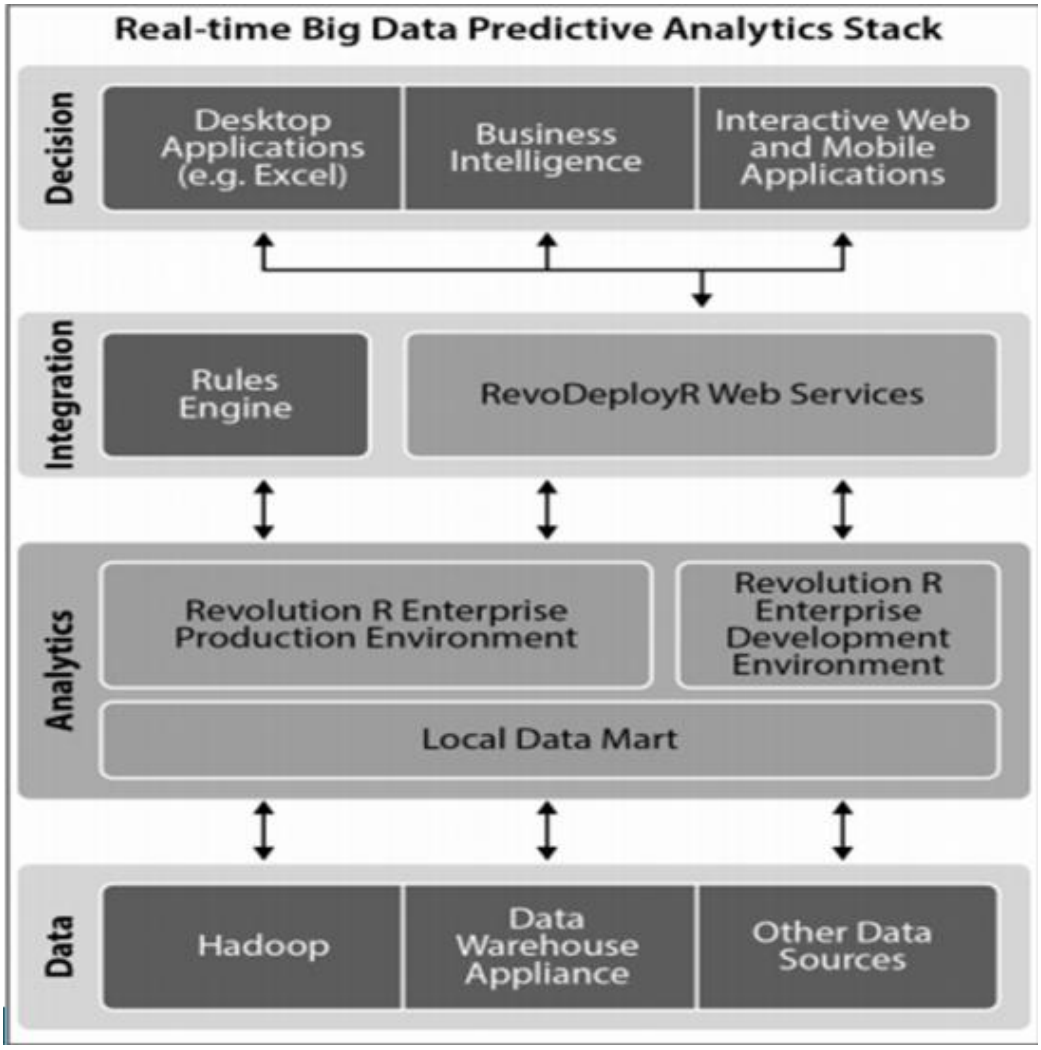


Fig2 Big Data Analytics layers

Bharathi, 2017)

Data Layer: This layer has RDBMS structured based such as Oracle, MySQL or semi-structured based like MongoDB or unstructured based like streaming data from the web world.

Analytics Layer: This layer provides the environment for implementing and building models for dynamic data analytics and deploy real time values.

Integration Layer: The end user applications and analytical engine are located in this layer.

Decision Layer: This is the layer where clients interact with system. It can be consisted of end user applications such as mobile apps, web applications, desktop applications and so on.

2.2 Recommendation Systems

It has been noted that a recommendation system which belongs to the class of Information retrieval, datamining and machine learning to be quite necessary in the world of E-commerce. Historical data from visitors and customers can be used to analyze the users needs and product preferences (Vaidya and Khachane, 2017) and also handling some business problems, such as goods promotion strategy and correlation product recommendation. (Raghani, 2019) Recommendation system is an information filtering technique,provides users with information, which he/she may be interested in. This project work revolves around recommending a product using Apriori algorithm. In the same context, Karandeep Singh (Abhishek Oraganti et al., 2019) implemented a useful recommendation technique using Apriori algorithm. This technique aims to detect association rules and examines frequently purchased items and those items that are then suggested as a recommendation to the customer.

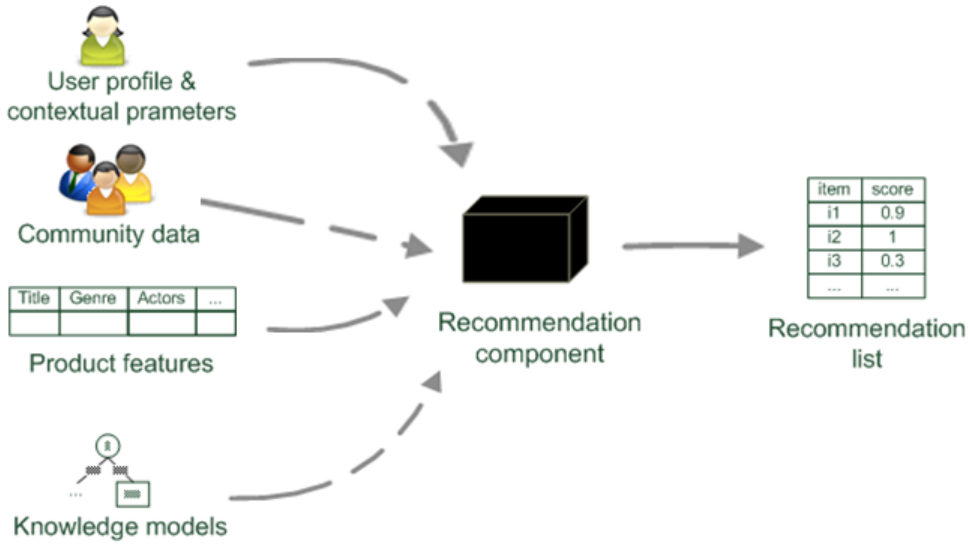


Fig3 : Recommendation Paradigm(Jannach et al., 2010)

One recent research on mining association rules (Ali Anaissi, and Madhu Goyal, 2015) raises the title of the SVM-Based Association Rule for Knowledge and Classification. The ultimate goal is to find an existing relation between product sales that can help retailers to identify new opportunities to sell their products to customers. They aim to find the patterns of knowledge in

big data which are not obvious and can help to generate more understanding for data holders and also more opportunities for crucial tasks including strategic planning and decision making. “The main purpose of this paper is to find connections existing among the items that can assist retailer with distinguishing new open doors for cross-offering their items to clients.”(Anaissi and Goyal, 2015)

Several projects related to basket analysis have been carried out by Kaggle. A kernel entitled Market Basket Analysis - Association Rule Mining(ARM) - Apriori by (Camilo-Data Scientist at Late-Nights Miami) is one of them. The purpose is to make some business-oriented analysis to discover insights hidden within the data and also determine sell-up opportunities. (Kaggle.com, 2018)

Another study about Association Rules has been done by Michael Hahsler with the title of “A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules” which contains a collection of commonly used measures of significance for association rules and item sets.(Hahsler, 2015)

Chapter 3: Background

3.1 Requirement Collection

The process of collecting system requirements in this project covers the need for online store related information, data needs from problem identification, problem-solving, database retrieval, data processing planning taken from database in real time so that it will be processed into data mining. Data mining is an automated process for existing data and data to be processed in the form of very large data, and the purpose of data mining is to obtain relationships or patterns that may provide useful indications.

3.2 Data Retrieval

The initial process of system design analysis is to collect customer purchase history data. A portion of data utilised for the analysis was collected from the two separate eCommerce platforms that helps online retailers run their business (Linnworks, SoldEazy) and the rest was collected from the collection of the Company's eight different accounts on Amazon and eBay.

3.2.1 Orders Collection

Order details was easy as datasets were available and ready for user download in Linnworks platform. Prior to any data manipulation, it is essential to extract the data and transform it into a format that can be easily used in the processing stage. Since the datasets were in CSV format, they were represented as a Pandas DataFrame.The dataset obtained is as follows:

Full_customer_data.csv: Containing the details of historical 486262 rows data purchase transactions on Online Store between 2015 and 2019 in their eBay and Amazon channels. Features: "OrderId", "FullName", "EmailAddress", "ReceievedDate", "ProcessedOn", "PostCode", "Source", "SubSource", "Country", "Address1", "Address2", "Address3", "Town", "BuyerPhoneNumber", "OrderItemNumber", "SKU", "ItemTitle", "OriginalItemTitle", "Quantity". Below is a sample of what the main dataset consists of.

nOrderId	cFullName	cEmailAddress	dReceievedDate	dProcessedOn	cPostCode	Source	SubSource	cCountry	Addres
166057	Miss Michelle Wright	t4sprhfpbpvvc7s@marketplace.amazon.co.uk	17/10/2015 19:46:00	19/10/2015 10:40:43	TS5 7LS	AMAZON	kidoloop	United Kingdom	Hackn W
166593	Abigail Sadler	43ql3t7hdm6r5k1@marketplace.amazon.co.uk	21/10/2015 06:25:00	21/10/2015 09:31:28	LS25 3EQ	AMAZON	Electronic Component	United Kingdom	Home Fa Cotta
188732	paul mcbrien	6l1ck29tfnv0k39@marketplace.amazon.co.uk	25/12/2015 10:33:28	28/12/2015 12:33:02	HP79RX	AMAZON	Electronic Component	United Kingdom	149 chilt heig
189051	KAtherine Cooper	0hmcs43nlbm4jfl@marketplace.amazon.co.uk	26/12/2015 16:04:08	28/12/2015 12:56:31	RG26 3PP	AMAZON	Electronic Component	United Kingdom	11 Gle Clc
189068	Elaine Ainsworth-Jones	4qvss7y1th0x1zg@marketplace.amazon.co.uk	26/12/2015 16:47:41	28/12/2015 10:12:33	BN3 5RL	AMAZON	Electronic Component	United Kingdom	1 St Phil Me

Fig 4 : Full_Customer_data.csv data sample

Column Name	Description
OrderId	This is the primary key for each order.
FullName	This shows the name of customer
EmailAddress	This is the customer's Email Address
Received Date	This is the date that order submitted
ProcessedOn	This shows the date that order shipped
PostCode	This is the postcode of shipping address
Source	This indicates the source of order(Amazon or eBay)
SubSource	This is the name of channel that order submitted
Country	This shows the country of shipping
Address1,Address2,Address3	These three columns are shipping address

Town	This is customer's town
BuyerPhoneNumber	This is the customer's phone number
OrderItemNumber	This shows the number of order
SKU	This is the unique item number for a product
ItemTitle	This is the title given to the SKU
Quantity	This shows how many of the products has been ordered
OriginalItemTitle	This shows the original title of the product

Table1: Full_Customer_data.csv columns description

Customer.csv: Including the details of the 9306 customers who did transaction between 2015 and 2019. Features: "Customer email address", "Customer buyer phone number", "Shipping customer name", "Shipping address 1", "Shipping address 2", "Shipping address 3", "Shipping town", "Shipping region", "Shipping postcode", "Shipping country", "Shipping country code", "Order total", "Currency".

Customer email address	Customer buyer phone number	Customer company	Shipping customer name	Shipping address 1	Shipping address 2	Shipping address 3	Shipping town	Shipping region	Shipping postcode
c.scrivener@yahoo.com	1270614146	NaN	Claire Scrivener	Air Products Plc	2 Millennium Gate, Westmere Drive	NaN	Crewe	Cheshire	CW1 6AP
t74wcrpw40cwrd0@marketplace.amazon.co.uk	1803835656	NaN	Elaine M Swailes	4	Oakwood Close	NaN	Dartmouth	Devon	TQ6 9GA
l522y03sbztdf3v@marketplace.amazon.co.uk	7947612689	NaN	Mrs Jennifer Mcandrew	9 FOREST CLOSE	NaN	NaN	LONDON	NaN	NW6 7BF
crazy_tiger321@hotmail.co.uk	7731717548	NaN	Natasha Baddeley	129 High Street	Silverdale	NaN	Newcastle Under Lyme	Staffordshire	St5 6lr
eunice_kormi@hotmail.co.uk	7474876176	NaN	Eunice Kormi	18 Westover Close	NaN	NaN	Belmont	Surrey	Sm2 6ub
83sammy@live.co.uk	7835287115	NaN	Samantha Winterbottom	33 Wildbrook Crescent	Fitton Hill	NaN	Oldham	Greater Manchester	ol82nt

Fig 5 : Customer.csv data sample

Column Name	Description
Customer email address	This is the customer's Email Address
Customer buyer phone number	This is the customer's phone number
Shipping customer name	This shows the name of customer
Shipping address 1,Shipping address 2,Shipping address 3	These three columns are shipping address
Shipping town	This is customer's town
Shipping region	This is customer's region
Shipping postcode	This is the postcode of shipping address
Shipping country	This is the postcode of shipping country
Shipping country code	This shows the abbreviation of the country
Order total	This is the total amount of their order
Currency	This indicates the currency of transaction

Table2: Customer.csv columns description

3.2.2 Inventory Collection

Inventory is about the details of all stock products. These details were collected from 6 separate channels (**coolbitz**,**coolbitzit**,**precious-deal**,**s-max2009)** in eBay and (**Coolbitz Trading Group**, **Kids Craft Toys**) in Amazon. The datasets collected is as below:

coolbitz_listing.csv: Containing the details of 851 products which are listed in "coolbitz" Channel. Features: "dsheet_row_id", "dsheet_shop_initial", "dsheet_sitecode", "dsheet_currency", "dsheet_var_parent_sku", "dsheet_product_code", "dsheet_duration".Below is a sample of what the main dataset consists of.

dsheet_row_id	dsheet_shop_initial	dsheet_sitecode	dsheet_currency	dsheet_format	dsheet_duration	dsheet_var_parent_sku	dsheet_product_code
6972329	TA	UK	GBP	BIN	GTC	BENA1130	BENA1130
6973881	TA	UK	GBP	BIN	GTC	NA1001	NA1001
6976450	TA	UK	GBP	BIN	GTC	142979640851-P	unicornheadband-Rainbow
6976533	TA	UK	GBP	BIN	GTC	UW1228659	UW1228659
7001115	TA	UK	GBP	BIN	GTC	UW1223101-P	UW1223101-Green
7001119	TA	UK	GBP	BIN	GTC	UW1221585	UW1221585
7001122	TA	UK	GBP	BIN	GTC	UW1221586	UW1221586
7001124	TA	UK	GBP	BIN	GTC	UW1209333	UW1209333
7001127	TA	UK	GBP	BIN	GTC	UW1209334	UW1209334
7001129	TA	UK	GBP	BIN	GTC	UW1209355	UW1209355

Fig 6: eBay coolbitz channel items

coolbitzit_listing.csv: Containing the details of 531 products which are listed in "coolbitzit" Channel. Features: "dsheet_row_id", "dsheet_shop_initial", "dsheet_sitecode", "dsheet_currency", "dsheet_var_parent_sku", "dsheet_product_code", "dsheet_duration".Below is a sample of what the main dataset consists of.

dsheet_row_id	dsheet_shop_initial	dsheet_sitecode	dsheet_currency	dsheet_format	dsheet_duration	dsheet_var_parent_sku	dsheet_product_code
7033280	COZIT	UK	GBP	BIN	GTC	UW3368367	UW3368367
7036052	COZIT	UK	GBP	BIN	GTC	UW3368336	UW3368336
7041119	COZIT	UK	GBP	BIN	GTC	LaceRemover-P	LaceRemover-2Pc
7131278	COZIT	UK	GBP	BIN	GTC	UW1216753	UW1216753
7131358	COZIT	UK	GBP	BIN	GTC	UW1131557	UW1131557
7131386	COZIT	UK	GBP	BIN	GTC	UW1082521	UW1082521
7149004	COZIT	UK	GBP	BIN	GTC	6022631-JungleRubble	6022631-JungleRubble
7150316	COZIT	UK	GBP	BIN	GTC	6024232	6024232
7162747	COZIT	UK	GBP	BIN	GTC	UW1136122	UW1136122
7162762	COZIT	UK	GBP	BIN	GTC	UW1149160	UW1149160

Fig 7: eBay coolbitzit channel items

precious-deal_listing.csv: Containing the details of 1918 products which are listed in "Precious deal" Channel. Features:"dsheet_row_id", "dsheet_shop_initial", "dsheet_sitecode", "dsheet_currency", "dsheet_var_parent_sku", "dsheet_product_code", "dsheet_duration". Below is a sample of what the main dataset consists of.

dsheet_row_id	dsheet_shop_initial	dsheet_sitecode	dsheet_currency	dsheet_format	dsheet_duration	dsheet_var_parent_sku	dsheet_product_code
6952402	Preciuos	UK	GBP	BIN	GTC	6038301	6038301
6952407	Preciuos	UK	GBP	BIN	GTC	UW1150220	UW1150220
6952428	Preciuos	UK	GBP	BIN	GTC	6043983	6043983
6952465	Preciuos	UK	GBP	BIN	GTC	UW1147573	UW1147573
6952471	Preciuos	UK	GBP	BIN	GTC	6044084	6044084
6952500	Preciuos	UK	GBP	BIN	GTC	6043960	6043960
6952503	Preciuos	UK	GBP	BIN	GTC	UW1152538	UW1152538
6964901	Preciuos	UK	GBP	BIN	GTC	Hookah-RM1200	Hookah-RM1200
6965146	Preciuos	UK	GBP	BIN	GTC	GA359	GA359
6965147	Preciuos	UK	GBP	BIN	GTC	UW1223189	UW1223189

Fig 8: eBay precious deal channel items

s-max2009_listing.csv: Containing the details of 1264 products which are listed in "s-max2009" Channel. Features: "dsheet_row_id", "dsheet_shop_initial", "dsheet_sitecode", "dsheet_currency", "dsheet_var_parent_sku", "dsheet_product_code", "dsheet_duration".

dsheet_row_id	dsheet_shop_initial	dsheet_sitecode	dsheet_currency	dsheet_format	dsheet_duration	dsheet_var_parent_sku	dsheet_product_code
6952490	SMax	UK	GBP	BIN	GTC	6044770	6044770
6965477	SMax	UK	GBP	BIN	GTC	UW-TPH237	UW-TPH237
7001241	SMax	UK	GBP	BIN	GTC	UW1223445	UW1223445
7001327	SMax	UK	GBP	BIN	GTC	UW1220436	UW1220436
7034704	SMax	UK	GBP	BIN	GTC	UW3368336-2	UW3368336-2
4305677	SMax	UK	GBP	BIN	GTC	ADIDAS 5IN SHORT XS(6)	ADIDAS 5IN SHORT XS(6)
4305916	SMax	UK	GBP	BIN	GTC	SUPREMECASERED	SUPREME-SUP-IPHONEX
4306790	SMax	UK	GBP	BIN	GTC	IPHONECHARGING	IPHONECABLE_BLK
4408606	SMax	UK	GBP	BIN	GTC	UW1223303	UW1223303
4446439	SMax	UK	GBP	BIN	GTC	Sunglasses-P	Sunglasses-Car-Black

Fig 9: eBay s-max2009 channel items

Column Name	Description
dsheet_row_id	This is the unique key for each row.
dsheet_shop_initial	This column will show you the name of channel that SKU is listed in it.
dsheet_sitecode	This is the origin of product.

dsheet_currency	This column shows the currency.
dsheet_var_parent_sku	This column shows all SKUs (the unique item number for a product) that apply to the current search terms or filtering, including variation parents.
dsheet_product_code	This is the number given to the SKU.
dsheet_duration	This indicates the time that product will be listed. After this period it is going to be removed from the listed items.

Table3 : eBay products data sheet description

Coolbitz_Trading_Group.csv: Containing the details of 865 products which are listed in "Amazon Coolbitz" Channel. Features: "ItemNumber", "ListingPrice", "ItemTitle", "Channel SKU", "BarCode", "Level", "Source", "SubSource", "CategoryName", "ListingTitle".

ItemNumber	ItemTitle	CategoryName	Source	SubSource	ListingPrice	ChannelSKU	ListingTitle
2134	Pillow Pets Comfy Panda Hat	Default	AMAZON	Amazon Coolbitz	14.99	2134	Pillow Pets Comfy Panda Hat
2229	Pillow Pets Moshi Monster Poppet	Default	AMAZON	Amazon Coolbitz	24.99	2229	Pillow Pets Moshi Monster Poppet
2F-8EAP-X6XA	Coo11 UV Reactive Putty Stress Reliever Putty ...	Default	AMAZON	Amazon Coolbitz	9.99	2F-8EAP-X6XA	Coo11 UV Reactive Putty Stress Reliever Putty ...
3960	Bkids Wacky Bebee funny monkey with colourful ...	Bkids	AMAZON	Amazon Coolbitz	14.20	3960	B kids BeBee Wacky Monkey
NC679	100pcs Ladies Nail Tips Full Nails 10 Shape Ac...	Default	AMAZON	Amazon Coolbitz	2.99	NC679	100pcs FRENCH MANICURE NAILS Full Cover Beige ...
NC680	100pcs Ladies Nail Tips Full Nails 10 Shape Ac...	9. Sara Beauty	AMAZON	Amazon Coolbitz	2.99	NC680	100pcs FRENCH MANICURE NAILS Full Cover Transp...
NC701	20pcs Women Triangle Sponge Foundation Wedges ...	Default	AMAZON	Amazon Coolbitz	2.49	NC701	20PCS Beauty Makeup Cosmetic Nail Triangle Fou...
UW1247089	Magic Sand 1000g - 500g Bags Mould & Play Magi...	Default	AMAZON	Amazon Coolbitz	12.95	UW1247089	Motion Moving Sand 1000g Sand Motion Beach Pla...
UW1247089	Magic Sand 1000g - 500g Bags Mould & Play Magi...	Default	AMAZON	Amazon Coolbitz	12.95	UW1247089.3	Motion Moving Sand 1000g Sand Motion Beach Pla...
UW1247089	Magic Sand 1000g - 500g Bags Mould & Play Magi...	Default	AMAZON	Amazon Coolbitz	12.95	UW1247089/Coo11	Motion Moving Sand 1000g Sand Motion Beach Pla...

Fig 10: Amazon coolbitz channel items

Kids_Craft_Toys.csv: Containing the details of the product which are listed in "Amazon Kidoloop" Channel. Features: "ItemNumber", "ListingPrice", "ItemTitle", "Channel SKU", "BarCode", "Level", "Source", "SubSource", "CategoryName", "ListingTitle".

ItemNumber	ItemTitle	CategoryName	Source	SubSource	ListingPrice	ChannelSKU	ListingTitle
2318	DREAM LITES Friendly Frog Plush	Default	AMAZON	kidoloop	14.99	2318	DREAM LITES Friendly Frog Plush
4657	Bkids Pull-n-Rattle Stroller Toy Bebee	Bkids	AMAZON	kidoloop	8.99	4657	Bkids Pull-n-Rattle Stroller Toy Bebee
4664	Bkids Travel Discovery Gym	Bkids	AMAZON	kidoloop	39.99	4664	Bkids Travel Discovery Gym
NC678	100pcs Ladies Nail Tips Full Nails 10 Shape Ac...	Default	AMAZON	kidoloop	2.99	NC678	100 Pieces Translucent Full Cover False Nail D...
NC679	100pcs Ladies Nail Tips Full Nails 10 Shape Ac...	Default	AMAZON	kidoloop	2.99	NC679	100Pcs 10 SIZES False Fake Nails Full Nail acr...
NC680	100pcs Ladies Nail Tips Full Nails 10 Shape Ac...	9. Sara Beauty	AMAZON	kidoloop	2.99	NC680	100Pcs 10 SIZES False Fake Nails Full Nail acr...
NC701	20pcs Women Triangle Sponge Foundation Wedges ...	Default	AMAZON	kidoloop	2.49	NC701/KCT	20PCS Beauty Makeup Cosmetic Triangle Blender ...
UW1247071-Purple	Magic Sand 1000g - 500g Bags Mould & Play Magi...	Default	AMAZON	kidoloop	5.95	UW1247071-Purple	Coo11 Magic Motion Moving Play Sand Pack 500g ...
UW1247071-Purple	Magic Sand 1000g - 500g Bags Mould & Play Magi...	Default	AMAZON	kidoloop	5.95	UW1247071-Purple.1	Coo11 Magic Motion Moving Play Sand Pack 500g ...
UW1247071-Yellow	Magic Sand 1000g - 500g Bags Mould & Play Magi...	Default	AMAZON	kidoloop	5.95	UW1247071-Yellow	Coo11 Magic Motion Moving Play Sand Pack 500g ...

Fig 11: Amazon kidoloop channel items

Column Name	Description
ChannelSKU	This column shows all SKUs (the unique item number for a product) that apply to the current search terms or filtering, including variation parents.
ItemNumber	This is the number given to the SKU.
ItemTitle	This is the title given to the SKU.
BarCode	This is the barcode that has been assigned to the stock item.
ListingPrice	This is the default retail price of the SKU.
ListingTitle	This is the listed title given to the SKU which is visible to the customers.
Level	This is the physical stock holding without adjustment for stock in open orders.
CategoryName	This is the category assigned to the SKU. Linnworks categories are completely internal to the system and have no effect on your listings. Categories are simply an easy way to group similar stock items.

Source	This shows the Source of SKU which can be Amazon or eBay.
SubSource	This column will show you the name of channel that SKU is listed in it.

Table 4 : Amazon products data sheet description

3.4 Data Cleaning

To perform the analysis effectively, various data cleaning techniques were carried out. This process is the backbone of the present analysis and also a stimulating learning experience as the datasets provided in modern data analysis are likely to be corrupted with various errors such as incorrect, inconsistent, missing or duplicated data. Moreover, databases can have constructions, arrangements and formatting that may not best suit the purpose of this analysis. In addition, using machine learning methodologies with dirty data can prove troublesome to debug and can degrade the model.

The Data cleaning procedure we adopted for this analysis consists of the following steps:

- Removing Duplicates
- Dealing with Missing Values
- Replacing Values and Transforming Data
- Modifying Column Data Across Tables
- Renaming Columns
- Modifying DataTypes

3.4.1 Removing Duplicates

First, all DataFrames are checked for duplicate entries as it is the most common human error when dealing with populating datasets. Duplicated values can be an obstacle in the modelling process and lead to poor targeting and decision making. As the listed products from 4 eBay channels can have duplicates after joining them, the function drop_duplicates() was used to remove duplicate entries.

3.4.2 Dealing with missing values

We check for missing values using isnull(). Missing values are problematic in that there are variations in the ways and reasons data are missing that require different approaches to handle them accordingly. Data could be missing at random, or not at random. It is also troublesome in that missing values could be dependent on some other variables' values. Therefore, it is important to take the correct measures. Here, we believe that it is safe to remove some records with missing values depending upon their occurrences. However, since it is retail-related, the fact that many entries have missing data could have its own significance, therefore, removing observations can produce a bias in the model.

- In **Orders** DataFrame: 'cEmailAddress' column has 2249, 'dProcessedOn' column has 4770, 'cPostCode' column has 776, 'Address2' column has 250557, 'Region' column has 57357, 'BuyerPhoneNumber' column has 8818, 'SKU' column has 16 and 'ItemTitle' column has 12 null values.
    - Since the 'cEmailAddress', 'Address2', 'cPostCode' and 'Region' columns are categorical variable, the missing values are filled with 'unknown' using fillna() function.
    - Since 'BuyerPhoneNumber' is a numerical column and shows the phone number of customers, so there are several ways to deal with it. In this case, missing values in the 'BuyerPhoneNumber' columns were filled in with the '0000000000'.
    - The 'SKU' and 'ItemTitle' columns appear to have several None values. As SKU and Title of products are crucial for our statistics, it would be best to simply drop all the entries which do not have SKU and Title specified by dropna() function.
    - The 'dProcessedOn' column Nan values are filled in by default value of '01/12/2019 00:00:00'.

- In **Customers** DataFrame : 'Customer buyer phone number' column has 120, 'Customer company' column has 9304, 'Shipping address 2' column has 5010, 'Shipping address 3' column has 9280, 'Shipping region' column has 1177, 'Shipping postcode' column has 8 and 'Item title' column has 88 null values.
    - As 'Customer company', 'Shipping address 2', 'Shipping address 3', 'Shipping region' and 'Shipping postcode' columns are categorical variables and not really necessary in this dataset, the missing values are filled in with 'unknown' using fillna() function.

- The 'Item title' column appears to have several None values. As Title of products are crucial for our statistics, it would be best to simply drop all the entries which do not have Title specified by dropna() function.

- The **ebay_SMax_2009_Listed, ebay_precious_Listed, ebay_coolbitzit_Listed, ebay_coolbitz_Listed, ebay_SMax_2009_Listed** DataFrames did not have any NaN values. Rather, the missing values were encoded as the string '-' , so records with such values were dropped.

3.4.3 Replacing Values and Transforming Data

Categorical variables were sometimes represented in a single category in multiple ways. Replacement is then needed to convert the representations to one consistent method. For instance, the following command looks into the Orders DataFrame and lists all the different categorical variables for the 'SubSource' column.

As it can be seen, there are many inconsistent single value representations (for example 'EBAY0' and 'coolbitz' are the same). These were all replaced and made consistent using mask() function.

```
In [198]: Orders.SubSource.unique()

Out[198]: array(['kidoloop', 'Electronic Component', 'EBAY2', 'Amazon Coolbitz',
                 'http://kidoloop.com', 'EBAY9', 'EBAY1', 'Coolbitz', 'EBAY',
                 'EBAY12', 'giftandgadget', 'EBAY3', 'coolbitz es', 'AMAZON',
                 'EBAY16', 'coolbitz FR', 'coolbitz it', nan, 'coolbitz de',
                 'Coolbitz Germany', 'coolbitz', 'EBAY0', 'Coolbitz Italy',
                 'DIRECT', 'kidscrafttoys', 'Coolbitz Spain', 'EBAY10', 'Amazon',
                 'Amazon-KIDO-FR', 'Amazon-KIDO-ES', 'http://thelondonpartners.com',
                 'Amazon-Kido-DE', 'ebay', 'Stationary-CF0014A5'], dtype=object)
```

3.4.4 Modifying Column Data Across Tables

Many strings in each of the DataFrames needed to be formatted correctly and consistently. For instance, the Source and SubSorce (channel) name are written differently across the tables (lowercase, uppercase, with ending such as FBA). Below is an illustration of how the SubSource name in Orders is made consistent.

```
Orders.SubSource= Orders.SubSource.str.upper()
for index,row in Orders.iterrows():
    tempSubSource = row["SubSource"]
    SubSourcesplit = tempSubSource.split(" ")
    if(SubSourcesplit[-1] == "FBA" or SubSourcesplit[-1] == "INT"):
        del SubSourcesplit[-1]
    tempstring = " ".join(SubSourcesplit)
    Orders.set_value(index,"SubSource",tempstring)
```

3.4.5 Renaming Columns

In order to obtain DataFrames that accurately encode one-one, one-many relations, columns were named the same to establish the linkage and improve readability. Moreover, ambiguous column names are uncooperative during data analysis.

```
Orders.columns = ['OrderId','FullName','EmailAddress','RecievedDate','ProcessedOn','PostCode','Source','SubSource','Country',
                  'Address1','Address2','Address3','Town','Region','PhoneNumber','Company','ItemNumber','SKU','ItemTitle',
                   'OriginalItemTitle','Quantity']
```

3.4.6 Modifying DataTypes

As a final step towards having tidy data ready for analysis, it is crucial to ensure that each column has only one sort of datatype and that datatype should be appropriate for that variable. This includes ensuring that all names were lowercase for continuity using `str.lower()` function, ‘date’ column data is of a date-time datatype, and all rates and numbers are of a float data type using `to_numeric()` function.

3.5 Data visualization and exploration

Prior to any modeling, a great amount of attention was devoted to data exploration.
The significance of this step is in identifying underlying trends and observing remarkable correlations, stimulating questions regarding sales and proper decisions. Therefore, extensive exploration of the data was carried out prior to directly feeding the data into a Black Box and wait for the results.

Exploration and visuals are beneficial for understanding the data and to grasp critical information that could have been easily missed - information that supported current analysis in the long run.

3.5.1 Columns Exploration

- The total number of unique active products in channels (eBay, Amazon) is 14535 item.
- 252685 orders are from eBay, 225701 from Amazon, 6193 from Direct, 980 from Magento, 205 from CDiscount and 29 from Flubit.
- Between all eBay channels the highest sale belong to EBAY2 with 111674 orders.
- Between all Amazon channels the 'Amazon Coolbitz' channel contains the highest demand with 57096 orders.

3.5.2 top10 countries based on number of orders

This part explores the number over purchases over countries. As it can be seen, the top 10 countries with the highest number of orders are United Kingdom, Unknown, Ireland, Italy, France, Spain, Germany, United States, Sweden and Denmark.

	cCountry
United Kingdom	466531
UNKNOWN	3693
Ireland	2503
Italy	2404
France	1961
Spain	1596
Germany	1462
United States	702
Sweden	479
Denmark	435

Fig 12 : countries with highest total number of orders

3.5.3 Top 20 selling items

The dataset of Orders separates SKUs (products) into 14535 unique items. Displaying the total number of selling items on a bar chart as follows. This is intuitive as product with SKU of 81000 make the largest sale in the warehouse, followed by Stationary-RFS10356, UW20170503 and UW1072881.

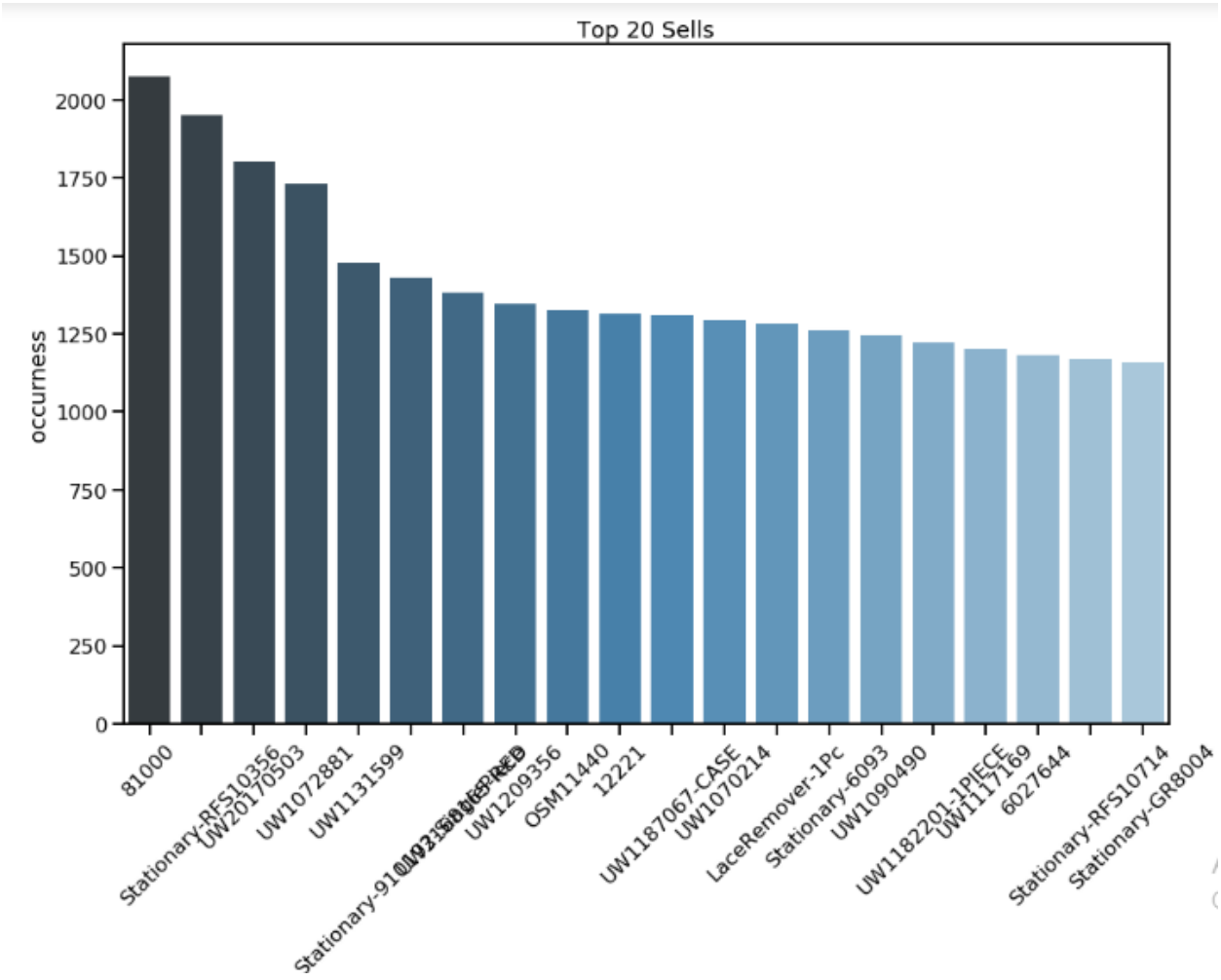


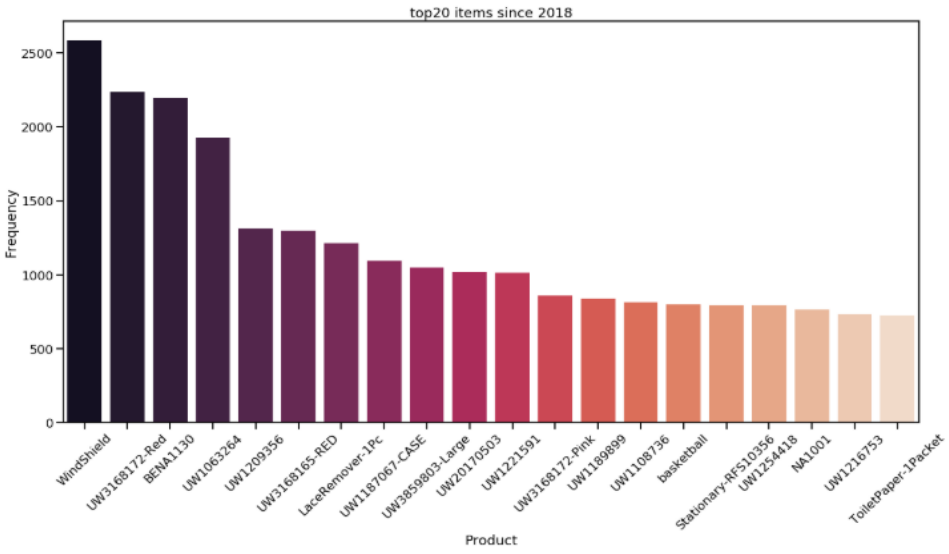
Fig 13: top 20 items in terms of frequency

	SKU	ocurness
1	81000	2074
481	Stationary-RFS10356	1950
1093	UW20170503	1805
2550	UW1072881	1732
5276	UW1131599	1477
1322	Stationary-910192-SinglePiece	1431
937	UW3168165-RED	1382
1349	UW1209356	1348
3611	OSM11440	1328
621	12221	1315
2304	UW1187067-CASE	1312
84	UW1070214	1295
880	LaceRemover-1Pc	1283
1370	Stationary-6093	1264
1150	UW1090490	1245
3556	UW1182201-1PIECE	1223
1008	UW1117169	1203
2930	6027644	1183
2819	Stationary-RFS10714	1172
2657	Stationary-GR8004	1158

Table 14: top 20 items in terms of frequency

3.5.4 Top 20 Items since 2018

Because of the change in their pattern from 2018, the Company required the provision of the best items since 2 years ago. This is clear that product with SKU of WindShield make the largest sale in the retail, followed by UW3168172-Red, BENA1130 and UW1063264.



3.5.5 Last Year Sale By Week

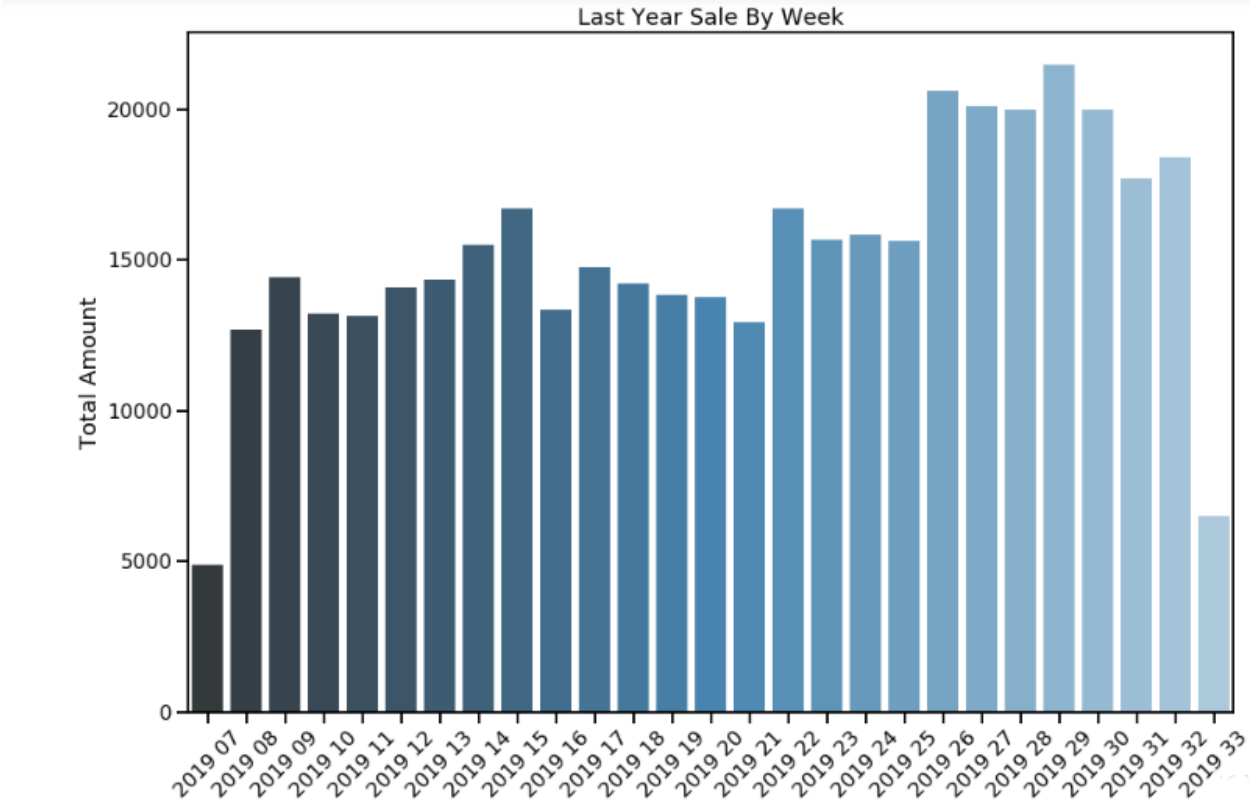


Fig 16: Last year week Sale

Due to the change of retail buying plan, Koolbitz tried to purchase new products since the beginning of 2019. As a result, they required analysis of the data of the sales based on each week. Therefore, the above examined results are the data pertinent to the sales of each week. As can be seen, the highest amount belongs to the week 29 of 2019 in total value of £21486 and the lowest is for week 7 in total value of £4898.

3.5.6 Last year profit by week

Figure 5 below shows the profit amount of each week in 2019. If the findings in Figure 5 and Figure 4 is compared, it can be seen that the result is the same, meaning that the highest profit belongs to the week with the highest sale and lowest profit is related to the week with lowest sale. Furthermore, it seems that there is a direct relation between sales and profits.

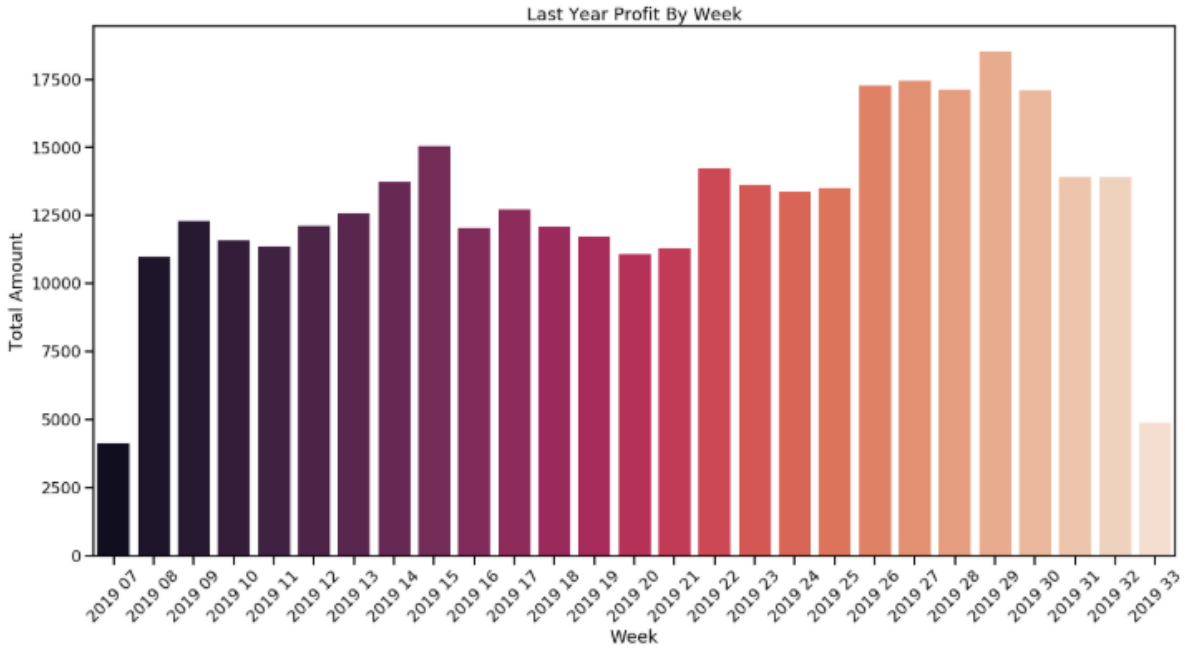


Fig 17: Last year profit by week

3.6 Modeling

3.6.1 Apriori Algorithm

Market container examination has been carried out by many organizations for the sole purpose of recovering item sets that are visited and are purchased simultaneously by the clients and further to this they have identified the clients purchasing habits. Apriori algorithm is a widely used technique. it is usually used to find those combinations of itemsets on a database containing a large number of transactions. Apriori works on bottom-up manner. it works on 2 steps scan and prune steps. In the scanning step it generates candidate itemset and in pruning step it reduces the size of candidate itemset by reducing itemsets whose support below the minimum_support threshold. Apriori is known as bottom-up breadth first search method. Apriori algorithm is iterative approach level-wise search, where k-itemset is used to generated (k+1) item set. Apriori algorithm uses Prior knowledge of frequent itemset property.

3.6.2 Key Concepts

Association Rules: Let I = $\{i_1, i_2, \ldots, i_n\}$ be a set of n attributes called items and D = $\{t_1, t_2, \ldots, t_n\}$ be the set of transactions. It is called database. Every transaction, t_i in D has a unique transaction ID, and it consists of a subset of itemsets in I. A rule can be defined as an implication, X⟶Y where X and Y are subsets of I (X,Y⊆I), and they have no element in

common. X and Y are the antecedent and the consequent of the rule, respectively. As an example for a rule in current scenario would be:

{Necklace, Earring} =>{Nail Polish}

Which means that if Necklace and Earring are bought, customers also buy Nail Polish.
There are multiple rules possible, so in order to select the interesting ones, constraints are implemented on various measures of interest and significance. Below some of these useful measures such as Support, Confidence and Lift are provided.

Support: Support is the default popularity of any item. We calculate the Support as a quotient of the division of the number of transactions containing that item by the total number of transactions. The range of Support is between 0 and 1. As an example:

Support(Nail Art Sticker) = (Transactions involving Nail Art Sticker) / (Number of all Transactions)

Confidence: Confidence is the likelihood that customer bought some products together. Dividing the number of transactions that include those products by the total number of transactions will give the Confidence figure.The range of Support is between 0 and 1. Example for Confidence:

Confidence (Christmas Hanging Santa Claus Elk Snowman Sock, Christmas Light 20 LED) = (Transactions involving both) / (Number of all Transactions)

Lift: Lift is the increase in the ratio of the sale of one item when you sell another one. This signifies the likelihood of the itemset Y being purchased when item X is purchased while taking into account the popularity of Y. For example, if the lift of {Nail Polish, Earring} is 5 it means the likelihood of a customer buying both Nail Polish and Earring is 5 times more than the chance of purchasing Nail Polish alone. If the Lift value is less than 1 it entails that the customers are unlikely to buy these together. Greater the value of Lift, the better combination of products.

Minimum Support and Confidence: These are user-specified parameters for Association Rule generation. These are used to exclude rules in the result that have a support or confidence lower than minimum support and minimum confidence respectively.

Frequent Itemsets: The sets of items which has a support value equal or greater than the minimum support.

Anti-monotonicity of the support measure: It assumes that (1) A subset of a frequent itemset must also be a frequent itemset, For example if {XY} is a frequent itemset, both X and Y should be a frequent itemset. (2) Similarly, for any infrequent itemset, all its supersets must be infrequent too.

3.6.3 Stages

The general processes undertaken in the Apriori method for obtaining frequent itemsets can be divided into two steps: (1) **Join** (merge): This process is done by combining items with other items until no more combinations can be established; and (2) **Prune**: The trimming process is the result of a merged item then trimmed by using the minimum support. This approach of extending a frequent itemset one at a time is called the "bottom up" approach.

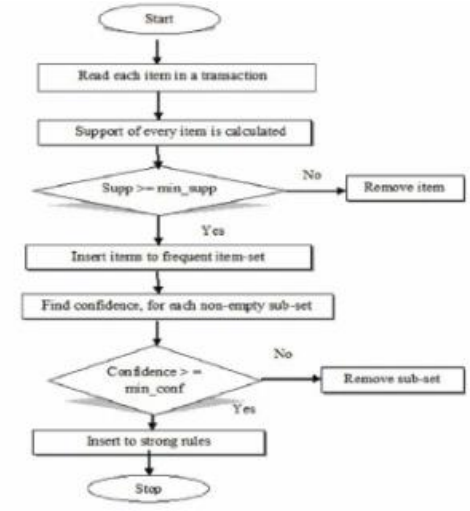


Fig 18: Apriori Process(Kalpana, 2017)

For Mining Association Rules step we can use a way which is called the two-step approach:
Step1 - Frequent Itemset Generation: Find all itemsets for which the support is greater than the minimum support threshold.

Step2 - Rule Generation: Create rules from each frequent itemset using the binary partition of frequent itemsets and look for the ones with high confidence. These rules are called candidate rules.

3.6.4 Apriori Pseudo-code

- **Join step:** C_k generated by joining L_{k-1} with itself
- **Prune step:** any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

C_k = Candidate itemset of size k

L_k = Frequent itemset of size k

L_1: {frequent items};

for(k=1; L_k != ∅; k++) do begin

C_{k+1} = candidates generated from L_k;

for each transaction t in database do

Increment the count of all candidates in C_{k+1};

That are contained in t

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\cup_k$ L_k (Khalil Jumaa, Al-Janabi and Abedlqader Ali, 2014, 65)

3.6.5 Simple Example Of Apriori Algorithm

Let us have a look at a simple example of algorithm. Before beginning the process we must set the support threshold to 22%. Consider a database containing 9 transactions:

Transaction ID	List of Items
1	I1,I2,I5
2	I2,I4
3	I2,I3
4	I1,I2,I4
5	I1,I3
6	I2,I3
7	I1,I3
8	I1,I2,I3,I5
9	I1,I2,I3

Table4: Transactions of the example

Step1: Generate the frequency table of all items that occur in all transactions.

Item	Frequency
I1	6
I2	7
I3	6
I4	2
I5	2

Table5: 1-itemset frequency

Step2: We know that only those products are significant which contain the support equal or greater than minimum support (0.22). So we should select the products which occur 2 or more. The table below indicates the single items which are frequent.

Item	Frequency
I1	6
I2	7
I3	6
I4	2
I5	2

Table6: Frequency table with minimum support

Step3: The next step is to create all possible pairs of frequent items (2-itemsets).

Item	Frequency
{I1,I2}	4
{I1,I3}	4
{I1,I4}	1
{I1,I5}	2
{I2,I3}	4

{I2,I4}	2
{I2,I5}	2
{I3,I4}	0
{I3,I5}	1
{I4,I5}	0

Table7: Frequency of all possible 1-frequent itemset pairs

So our frequent 2-itemsets are: {I1,I2} - {I1,I3} - {I1,I5} - {I2,I3} - {I2,I4} - {I2,I5}

Step4: The next step is to generate 3-itemset frequent pattern. Inorder to find the set of candidates we need to join Table 7 with itself. Our possible pairs are: {I1,I2,I3} - {I1,I2,I5} - {I1,I3,I5} - {I2,I3,I4} - {I2,I3,I5} - {I2,I4,I5}. Between all possible pairs we should pick those which contain minimum support, so our frequent 3-itemset are: {I1,I2,I3} - {I1,I2,I5}.

Note: In terms of Apriori Property all subsets of a frequent itemset must also be frequent.

Step5: At this stage we need to generate association rules from frequent itemsets. First for each frequent itemset 'I' we must generate all nonempty subsets of 'I'. Next for every nonempty subset 'S' of 'I', output the rule " S→(I - S)" if support_count(I) / support_count(s) >= minimum_confidence .

We had L={{I1},{I2},{I3},{I4},{I5},{I1,I2},{I1,I3},{I1,I5},{I2,I3},{I2,I4},{I2,I5},{I1,I2,I3},{I1,I2,I5}}. Let's take I = {I1,I2,I5}. Its all nonempty subsets are {I1,I2},{I1,I5},{I2,I5},{I1},{I2},{I5}. We assume that the minimum confidence threshold is 0.7. The resulting association rules are shown below:

- R_1: I1,I2 → I5
  - Confidence: S_C(I1,I2,I5) / S_C(I1,I2) = 2/4 = 0.5
  - $R_1$is rejected
- R_2: I1,I5 → I2
  - Confidence: S_C(I1,I2,I5) / S_C(I1,I5) = 2/2 = 1
  - $R_2$is selected.
- R_3: I2,I5 → I1
  - Confidence: S_C(I1,I2,I5) / S_C(I2,I5) = 2/2 = 1
  - $R_3$is selected
- R_4: I1 → I2,I5

    - Confidence: S_C(I1,I2,I5) / S_C(I1) = 2/6 = 0.3
    - $R_4$is rejected
- R_5: I2 → I1,I5
    - Confidence: S_C(I1,I2,I5) / S_C(I2) = 2/7 = 0.29
    - $R_5$is rejected
- R_6: I5 → I2,I1
    - Confidence: S_C(I1,I2,I5) / S_C(I5) = 2/2 = 1
    - $R_6$is selected

So the output of operating the algorithm on database D, is 3 association rules:

1. I1,I5 → I2
2. I2,I5 → I1
3. I5 → I1,I2

Chapter 4: Implementation

4.1 Data representation

`Python3` was chosen as the main language to conduct our analysis as it has very useful and powerful libraries and packages for our analytical and modeling needs. Specifically, we made use of the following libraries:

● **Pandas**: a prominent open source data manipulation library. It simplifies the retrieval of data from external sources and provides high performance, easy-to-use data structures and data analysis tools. For this project, the main Pandas functionalities utilised are the DataFrame and Series objects. The DataFrame object is 2-dimensional labeled data structures with columns consisting of one or different data types structured in a way similar to spreadsheets or SQL tables. The Series object is a one-dimensional object that can hold any data type and is used when setting DataFrame rows.

● **Numpy**: A fundamental package for scientific computing. It provides support for large multidimensional arrays with high-level efficient math functions for operations on these arrays. In particular, many machine learning algorithms can be expressed as a sequence of operations on arrays using NumPy.

● **Matplotlib**: A basic Python package with a variety of functions for data plotting and visualisation such as plots, maps, charts, etc. It greatly complements Pandas by providing visuals to better explore and understand the data.

● **Mlxtend**: An outstanding open source Python package that provides solid implementations of a range of machine learning and datamining algorithms and utilities for commonly used tools like classification, feature selection, regression and clustering such as random forests and k-means..etc.(Raschka, 2018)

4.2 Feature Selection

To create an accurate predictive model I chose features that will give us better accuracy whilst using less data. Benefits of Feature selection for our model:

● Reduces the complexity of our models.
● Improves the accuracy
● Reduces Overfitting

I remove unneeded and irrelevant attributes such as PostCode, Source, SubSource, Country, Address1 etc to keep only the best subset. (`Orders_selected_columns` DataFrame)

FullName	RecievedDate	SKU
Michael trigger	2017-11-03 14:09:39	6026411-SILVER
Georgina Sinclair	2016-11-24 14:18:36	OSM67601
Samantha Haley	2016-07-19 07:05:40	Vtech-164203
Brenda Manning	2019-12-02 11:05:52	UW1187067-CASE
Anne Easton	2018-04-03 17:01:36	6036215

Fig 19: Orders_selected_columns DataFrame sample

4.3 Models

4.3.1 Recommendations using Association Rules

First Mlxtend.frequent_patterns.apriori has been used to fetch frequent itemsets for association rule mining. Feeding it with current dataset containing transactions and features, the algorithm will come up with a set of indices of the items which are frequent and can be useful for association rule mining. Next, mlxtend.frequent_patterns have been used to generate association rules based on our frequent itemsets. To evaluate association rules one of the Confidence, Lift or Conviction metrics was selected and a threshold for that was set up.

The confidence of a rule indicates the probability of appearing the consequent in a transaction given that it also contains the antecedent. The lift metric is used to show how much more often the antecedent and consequent of a rule occur together than in their independent shape. Conviction shows how much consequent depends on its antecedent. The higher value of conviction would amount to higher dependency.

API:

apriori(df, min_support=0.0045, use_colnames=False)→ Get frequent itemsets from a one-hot DataFrame.

`df`: Pandas DataFrame in one-hot encoded format.

`min_support`: a float variable between 0 and 1 with the default value of 0.5 for minimum support of the itemsets returned.

`use_colnames`: a boolean variable with the default value of False. we can set `use_colnames=True` to convert indices values into the respective item names.

rules =(association_rules(frq_items, metric ="Confidence", min_threshold = 1)→ Generating Rules

`frq_items`: results of operating apriori function on our dataset

`metric`: indicates the metric we want to utilise in the model

`min_threshold`: declare the minimum threshold for the metric

Model Implementation:

I started by changing Orders_selected_columns dataset to the suitable form which is transactions in order to utilize it in my model.

```
In [34]: basket = (Orders_selected_columns.groupby(['FullName','SKU'])['totalbuying'].sum()
                   .unstack().reset_index()
                   .fillna(0).set_index('FullName'))
         def encoding(x):
             if(x<= 0):
                 return 0
             if(x>= 1):
                 return 1
         basket_encoded = basket.applymap(encoding)
         basket.head()
```

Out[34]:

	FullName	Hookah-RM1213	Hookah-RM1222-Clear	Stationary-ZB012-4packet	Stationary-ZB013-4packet	Stationary-ZB014-4packet	UW1209334	SMALLARROWS-10	UW1222875-1Pc	UW1222874-1Pc	UW1072915	UW1069911
0	FAO Manager Kathryn Normie	1	1	0	0	0	0	0	0	0	0	0
1	Isabel Sayers	0	0	1	1	1	0	0	0	0	0	0
2	.simon beaumont	0	0	0	0	0	1	1	0	0	0	0
3	01495442650	0	0	0	0	0	0	0	1	1	0	0
4	10, Walters Close	0	0	0	0	0	0	0	0	0	1	1

Fig 20: basket DataFrame

Then `apriori` model have been applied. Supposing that only rules for those items that are purchased at least once a day, or 7 times in one week are required, since the present dataset is for 4 year-week time period and in total 52710 transactions are available, The support for those items can be calculated as (52*7*4)/52710 = 0.03. After operating model the frequent itemsets can be seen. Below is a sample of our frequent itemsets:

```
{'UW31161-Red', 'UW31161-Green'})
{'UW3168172-Pink', 'UW3168172-Red'})
{'UW3268209', 'UW3168196'})
{'UW3268230', 'UW3268209'})
{'UW3268261', 'UW3268209'})
{'UW3559505-0', 'UW3559505-1'})
{'Vtech-160603', 'Vtech-160703'})
{'WF910', 'WF909'})
{'WF911', 'WF909'})
{'WF910', 'WF911'})
{'unicornheadband-Pink', 'unicornheadband-Rainbow'})
{'unicornheadband-Pink', 'unicornheadband-Silver'})
{'unicornheadband-Pink', 'unicornheadband-Turquoisepurple'})
{'unicornheadband-Rainbow', 'unicornheadband-Silver'})
{'unicornheadband-Rainbow', 'unicornheadband-Turquoisepurple'})
{'unicornheadband-Turquoisepurple', 'unicornheadband-Silver'})
{'6035619-Rocky', '6035619-Marshall', '6035619-Chase'})
{'6035619-Marshall', '6035619-Skye', '6035619-Chase'})
{'6035619-Marshall', '6035619-Chase', '6035619-Zuma'})
```

Fig 21: Frequent itemsets

Next stage was about applying association_rules model and collecting the inferred rules in a list. Implemented measures are confidence and lift. Rules derived from the frequent itemsets with level of confidence above the 20 percent threshold (min_threshold=0.2)Then that list was iterated to see the association rules. As a result, i have got 186 association rules.

```
====================================
Rule: UW-BC2290 -> UW-BC2291
Support: 0.030493265035097705
Confidence: 0.3780821917808219
Lift: 57.94266869609335
====================================
Rule: UW-BC2290 , 6037169 -> UW-BC146205
Support: 0.030303547713906278
Confidence: 0.43529411764705883
Lift: 84.94762288477034

====================================
Rule: Hookah-RM1222-Blue , KL708-Pink -> Hookah-RM1225-Red
Support: 0.030284575981787135
Confidence: 0.4205882352941177
Lift: 89.44004524886878

====================================
Rule: OSM11440 , unicornheadband -> 6035619-Marshall
Support: 0.03070195408840827
Confidence: 0.8829268292682928
Lift: 162.76891405043747
```

Fig 22: Association rules of Apriori algorithm

While the association rules for our dataset is presented in figure 22. For instance, based on the last Rule, most of the customers who bought products with SKU OSM11440 and unicorn headband, also bought the item with SKU 6035619-Marshall.

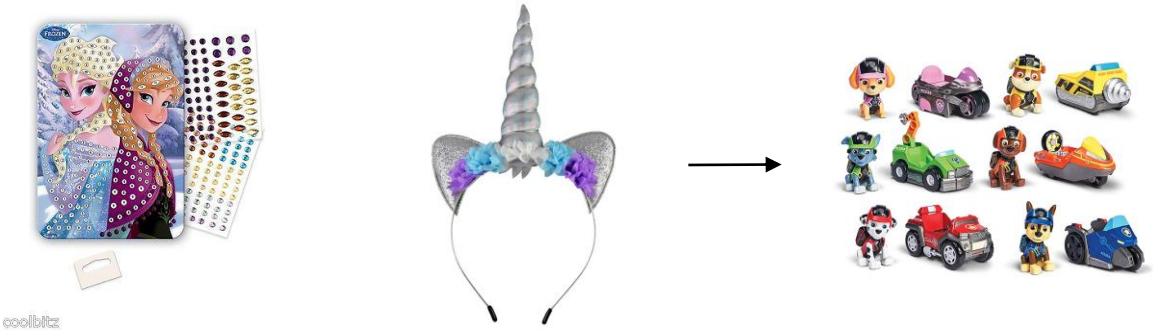


Fig 23: Association rules of three product

4.3.2 Recommendations using K-Nearest-Neighbor

K-Nearest Neighbor is known as a non-parametric method of pattern classification. It classifies the data based on similarities or distance to the stored training tuples.(Peterson, 2009) it is learnt by comparing a specific test tuple with a set of training tuples that are similar to it. It is classified based on the class of their closest neighbors.(Adeniyi, Wei and Yongquan, 2016) "K" indicates the number of neighbors were taken into account in determining the class. (Han, Kamber and Computer, 2012, 285 - 300) The only issue of using this model on present dataset is that we need the rate of products which is taken from the customers. In order to deal with this issue a rate of 1 for every transaction is assumed. Simplifying the pseudo-code:

1. Load the training and test data
2. Choose the value of K
3. For each point in test data:

    - find the Euclidean distance to all training data points
    - store the Euclidean distances in a list and sort it
    - choose the first k points
    - assign a class to the test point based on the majority of classes present in the chosen points

4. End (Li et al., 2012)

Model Implementation:

To implement KNN algorithm first we should import the NearestNeighbors library from sklearn Python package. Then i have used the one-hot encoding DataFrame which was already created in 3.3.1 section and also sparse matrix representation was used to convert the mentioned DataFrame to a sparse matrix. For the purpose of finding the nearest neighbors we must determine the KNN model and then fit that model with the dataset.

API:

NearestNeighbors(*n_neighbors=5, radius=1.0, algorithm='brute', metric='cosine', p=2*)→ to implement neighbor searches.

`n_neighbors`: number of neighbors

`radius`: range of parameter space to use

`algorithm`: Algorithm used to compute the nearest neighbors. We are allowed to use ‘’ball_tree” to use BallTree, kd_tree to use KDTree and “brute” to use brute-force search.

`Metric`: declare the method for distance computation.cosine method is selected so our algorithm will calculate the cosine similarity.

The output of fitting model is the distance and indice of the K products which are more similar to the selected item.

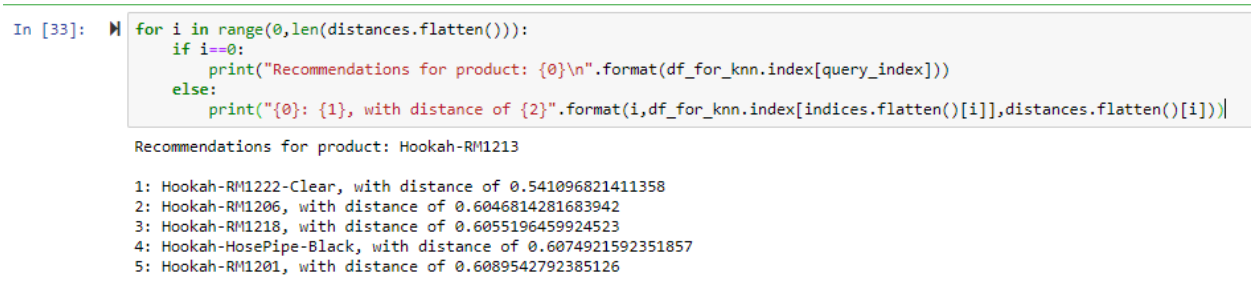


Fig 24: Nearest neighbors for a product

As can be seen in figure24, most similar items to Hookah-RM1213 includes Hookah-RM1222-Clear, Hookah-RM1206, Hookah-RM1218, Hookah-HosePipe-Black, Hookah-RM1201.

Hookah-RM1213

Hookah-RM1222-Clear

Hookah-RM1218

Hookah-RM1206

Hookah-HosePipe-Black

Hookah-RM1201

Fig 25: Similar items fetched by KNN

Chapter 5: Evaluation

Throughout the project, the main finding was that the items in our association rules that were frequently occurred, were simultaneously in the same category. For instance, 42 out of 186 of our association rules (equivalent to 22%) that were identified were also including cosmetics products as well.

{NC680,Q094} → NC811

Fig 26: Sample of an association rule

Further, after the last year's weekly sale was plotted, it was deduced that there was a soar in their sale from week 25 to date. It was realized, after obtaining the information from Koolbitz, that they have listed again the products that were sold less than others. The reason behind listing this type of products was stated that they were appearing in the front pages of sales platforms.

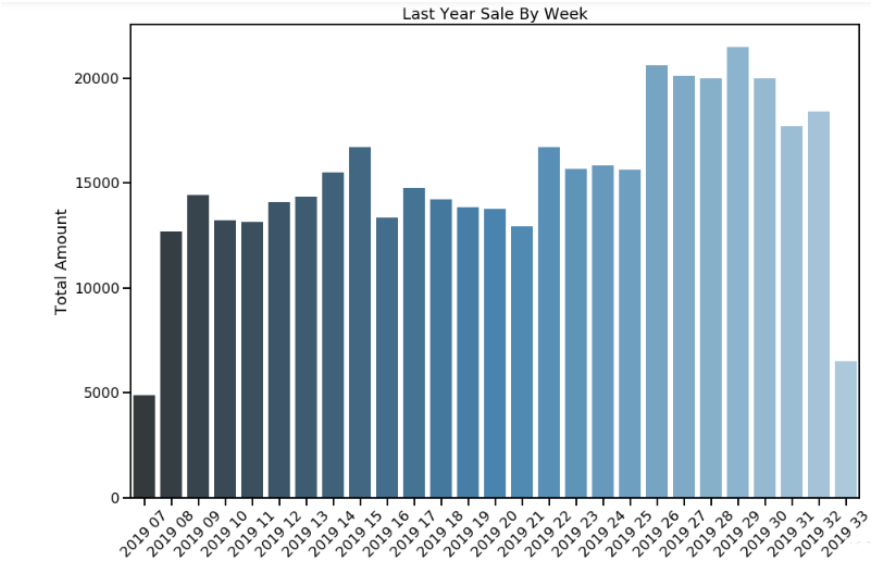


Fig 27: Weekly sales

It is also worth mentioning that with attention to the top sold items, the price average of the 20 items with highest sale is £4.66 which means cheaper items gained higher sales.

After analysing Koolbitz sales data it was found that every year sales prior to Christmas (the last two month of every year), the sales rate has experienced a significant rise. It is thus recommended that the attendance and consequence items identified from the association rules, should be listed as Gift Boxes to be shown as suggested to the visitors.

Comparing Models:

The results of two models for a product with SKU Hookah-RM1213 shall be compared. The index of product in our one-hot encoding DataFrame is needed to fetch the corresponding association rules via Apriori algorithm. The results of both models is as follows:

```
KNN Recommendations for product: Hookah-RM1213

1: Hookah-RM1222-Clear, with distance of 0.541096821411358
2: Hookah-RM1206, with distance of 0.6046814281683942
3: Hookah-RM1218, with distance of 0.6055196459924523
4: Hookah-HosePipe-Black, with distance of 0.6074921592351857
5: Hookah-RM1201, with distance of 0.6089542792385126
```

Fig 28: KNN Recommendations

```
Apriori Recommendations for product: Hookah-RM1213

1: [Hookah-RM1206, Hookah-HosePipe-Black] with lift of 3.624592745206416
2: [Hookah-RM1206, TUW288308, ] with lift of 3.3471258932712369
3: [Hookah-RM1206, Hookah-RM1222-Clear] with lift of 3.2453333834834230
4: [Hookah-HosePipe-Black, Hookah-RM1203-Silver] with lift of 3.2946816
```

Fig 29: Apriori Recommendations

As the figures clearly show, the results of both algorithms seem to be similar and also they are showing related products.

The K-NN is often referred to as “Lazy learner” in the sense that it simply stores the given training tuples and waits until it is given a test tuple, on the other hand, K-NN takes care of new data, meaning that when new data is needed to be added we do not need to run the model again from the beginning. But Apriori is not robust with updating data and with any newly added data the model must be run again.

Chapter 6: Conclusion

6.1 Conclusion

Based on the work of this project it is observed that data analysis and datamining tools can be vital for the companies that are operating in E-commerce space. Unclear patterns and useful information can be retrieved to help online shop providers in making decisions and coming up with new ideas about their business. One of these useful patterns is the understanding customers behaviour in purchasing. A good insight of customers behaviour and similarity between the products can be acquired by utilizing Recommendation Systems. I have used two widely used algorithms for mining association rules on a real-world dataset belongs to koolbitz ltd group. mlxtend which is an open source machine learning python package provided me to apply the Apriori algorithm with the 0.03 and 0.2 minimum threshold for support and confidence metrics respectively on my dataset. The results included 186 association rules that show us the certain relations between the products which are known as frequent items.
Further i have implemented a recommendation system using K Nearest Neighbors algorithm. Cosine metric and Brute-force algorithm has been employed to computing similarity between items and search operation. I also used sparse matrix to represent my dataset. Then in order to check the result validity of two models a comparison has been done and almost similar outcome was observed. Moreover it was noticed that 22% of the frequent itemsets in association rules belong to Beauty items category and 14% are related to the toys category.
Besides data explorations and visualisations has been carried out. It has been deduced by listing the products with low sales rate again and linking the unlinked products to the channels they can enhance sales significantly and also higher sales can fetch more profit to the seller.
After the Company was provided with the findings it appeared that they are interested in them. Also they said these observes and rules can change their business models and decisions.

6.2 Future Investigation

As prospective works first we can use hash based apriori algorithm that provides better results in less amount of time as compared to apriori. Further we can cluster the customers based on their purchasing behaviour to predict the suitable products for new customers. Moreover after having communication with Company we decided to add a new feature as a rate to their new website. In the more clear way every customer rank the item after purchasing to indicate his/her satisfaction. Then we can implement another recommendation system that would operate more accurate.

Reference list

[**Cited as:** Abhishek Oraganti et al., 2019] Abhishek Oraganti, Karandeep Singh Talwar, Ninad Mahajan and Pravin Narsale (2019). Recommendation System using Apriori Algorithm. *International Journal for Scientific Research and Development*, [online] 3(1), pp.183–185. Available at: http://ijsrd.com/Article.php?manuscript=IJSRDV3I1107 [Accessed 20 Aug. 2019].

[**Cited as:** Adeniyi, Wei and Yongquan, 2016] Adeniyi, D.A., Wei, Z. and Yongquan, Y. (2016). Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. *Applied Computing and Informatics*, 12(1), pp.90–108.

[**Cited as:** Anaissi and Goyal, 2015] Anaissi, A. and Goyal, M. (2015). SVM-based association rules for knowledge discovery and classification. *2015 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*. [online] Available at: https://ieeexplore.ieee.org/document/7476236?section=abstract.

[**Cited as:** Fatoni, Utami and Wibowo, 2018] Fatoni, C.S., Utami, E. and Wibowo, F.W. (2018). Online Store Product Recommendation System Uses Apriori Method. *Journal of Physics: Conference Series*, [online] 1140, p.012034. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/1140/1/012034 [Accessed 20 Aug. 2019].

[**Cited as:** Hahsler, 2015] Hahsler, M. (2015). *A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules*. [online] Hahsler.net. Available at: https://michael.hahsler.net/research/association_rules/measures.html [Accessed 20 Aug. 2019].

[**Cited as:** Han, Kamber and Computer, 2012] Han, J., Kamber, M. and Computer, P. (2012). *Data mining : concepts and techniques*. 3rd ed. Amsterdam ; Boston: Elsevier/Morgan Kaufmann.

[**Cited as:** Jannach et al., 2010] Jannach, D., Zanker, M., Felfernig, A. and Friedrich, G.

(2010). *Recommender systems : an introduction*. Cambridge: Cambridge University Press.

[**Cited as:** Kaggle.com, 2018] Kaggle.com. (2018). *Assoc. Rule Discovery! {Cookies <-> Hot Chocolate}* | *Kaggle*. [online] Available at: https://www.kaggle.com/kalash04/assoc-rule-discovery-cookies-hot-chocolate [Accessed 21 Aug. 2019].

[**Cited as:** Kalpana, 2017] Kalpana, D. (2017). DATA MINING APRIORI ALGORITHM IMPLEMENTATION USING R. *International Research Journal of Engineering and Technology (IRJET)*, [online] 4(11). Available at: https://www.irjet.net/archives/V4/i11/IRJET-V4I11328.pdf.

[**Cited as:** Khalil Jumaa, Al-Janabi and Abedlqader Ali, 2014] Khalil Jumaa, A., Al-Janabi, S.T.F. and Abedlqader Ali, N. (2014). Hiding Sensitive Association Rules over Privacy Preserving Distributed Data Mining. *kirkuk university journal for scientific studies*, [online] 9(1), pp.59–72. Available at: https://www.iasj.net/iasj?func=article&aId=89609 [Accessed 20 Aug. 2019].

[**Cited as:** Li et al., 2012] Li, N., Zeng, L., He, Q. and Shi, Z. (2012). Parallel Implementation of Apriori Algorithm Based on MapReduce. *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. [online] Available at: https://ieeexplore.ieee.org/document/6299286 [Accessed 21 Aug. 2019].

[**Cited as:** Peterson, 2009] Peterson, L. (2009). K-nearest neighbor. *Scholarpedia*, [online] 4(2), p.1883. Available at: http://www.scholarpedia.org/article/K-nearest_neighbor [Accessed 21 Aug. 2019].

[**Cited as:** Praveena and Bharathi, 2017] Praveena, M.D.A. and Bharathi, B. (2017). A survey paper on big data analytics. *2017 International Conference on Information Communication and Embedded Systems (ICICES)*. [online] Available at: https://ieeexplore.ieee.org/document/8070723 [Accessed 20 Aug. 2019].

[**Cited as:** Raghani, 2019] Raghani, V. (2019). *Recommender Systems using Apriori –*

An Application in Retail using Python. [online] SogetiLabs. Available at: https://labs.sogeti.com/recommender-systems-using-apriori/ [Accessed 20 Aug. 2019].

[**Cited as:** Raschka, 2018] Raschka, S. (2018). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *Journal of Open Source Software*, [online] 3(24), p.638. Available at: https://joss.theoj.org/papers/10.21105/joss.00638 [Accessed 20 Aug. 2019].

[**Cited as:** Schafer, Konstan and Riedi, 1999] Schafer, J.B., Konstan, J. and Riedi, J. (1999). Recommender systems in e-commerce. *Proceedings of the 1st ACM conference on Electronic commerce - EC '99*. [online] Available at: https://dl.acm.org/citation.cfm?doid=336992.337035 [Accessed 20 Aug. 2019].

[**Cited as:** Vaidya and Khachane, 2017] Vaidya, N. and Khachane, A.R. (2017). Recommender systems-the need of the ecommerce ERA. *2017 International Conference on Computing Methodologies and Communication (ICCMC)*. [online] Available at: https://ieeexplore.ieee.org/abstract/document/8282616 [Accessed 20 Aug. 2019].