LUISS

Machine Learning: NTT

Final Presentation

NTT DATA

Our Team

Balkiss Tekaya

Tunisian
Background in:
Business Administration
Business Analytics

Giulia Di Martino

Italian
Background in:
Finance

Martina Bozzi

Italian
Background in:
Political Science

Sirine El Feki

Tunisian
Background in:
Business Administration
Business Analytics

Tasnim Tekaya

Tunisian
Background in:
Business Administration
Business Analytics

NTT DATA

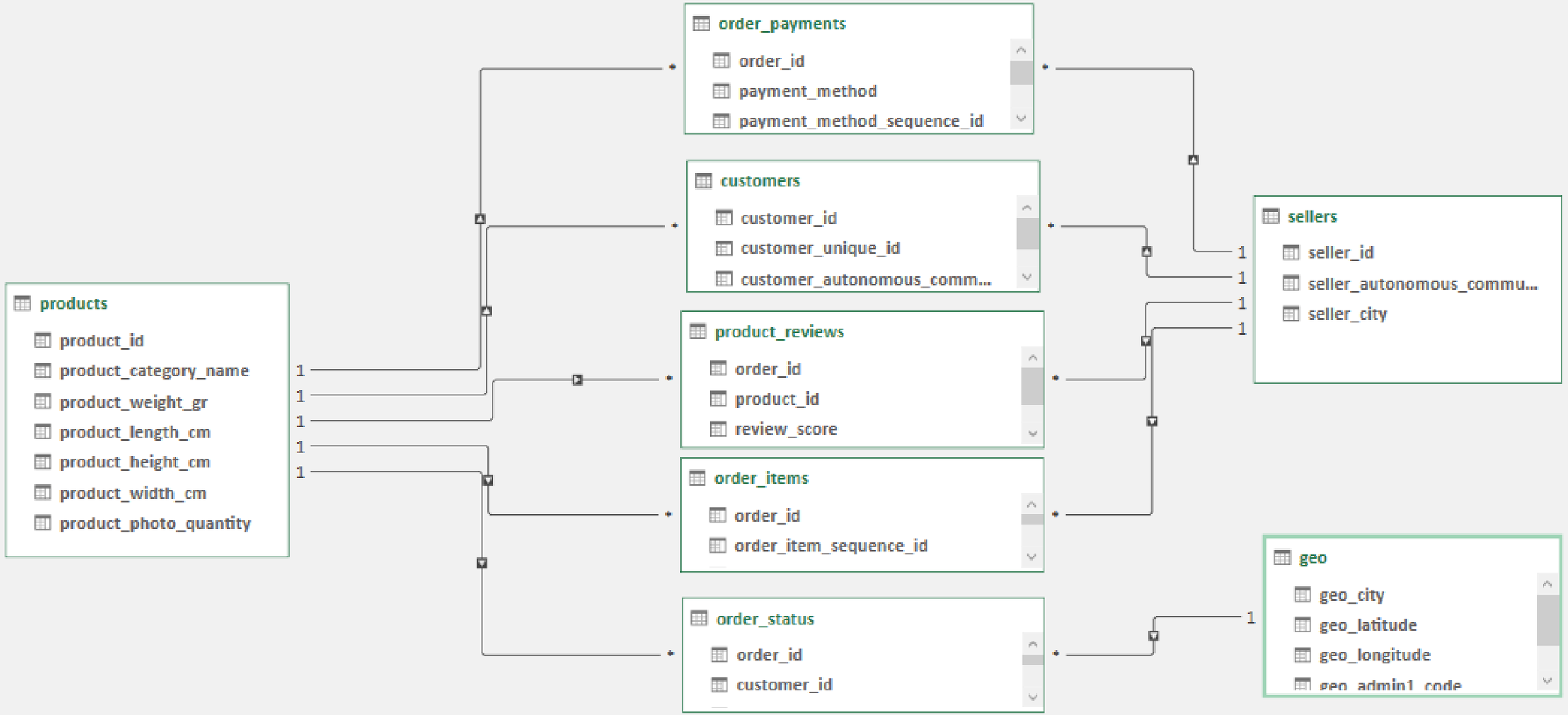
LUISS

Outline

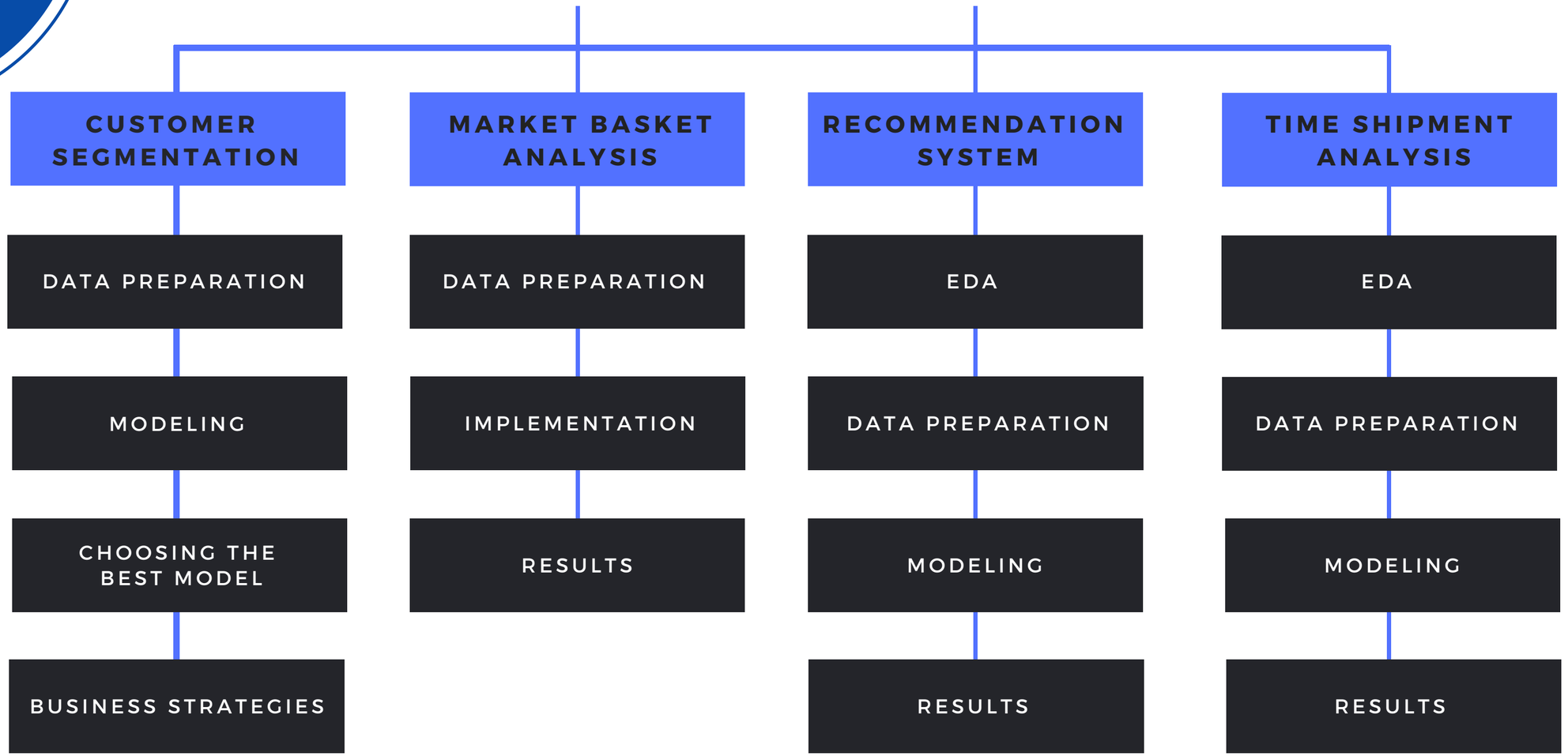
1. Introduction
2. Customer Segmentation
3. Market Basket Analysis
4. Recommendation System
5. Time Shipment Analysis

ENTITY RELATIONSHIP MODEL

Representation of the active relationships between variables of the provided datasets



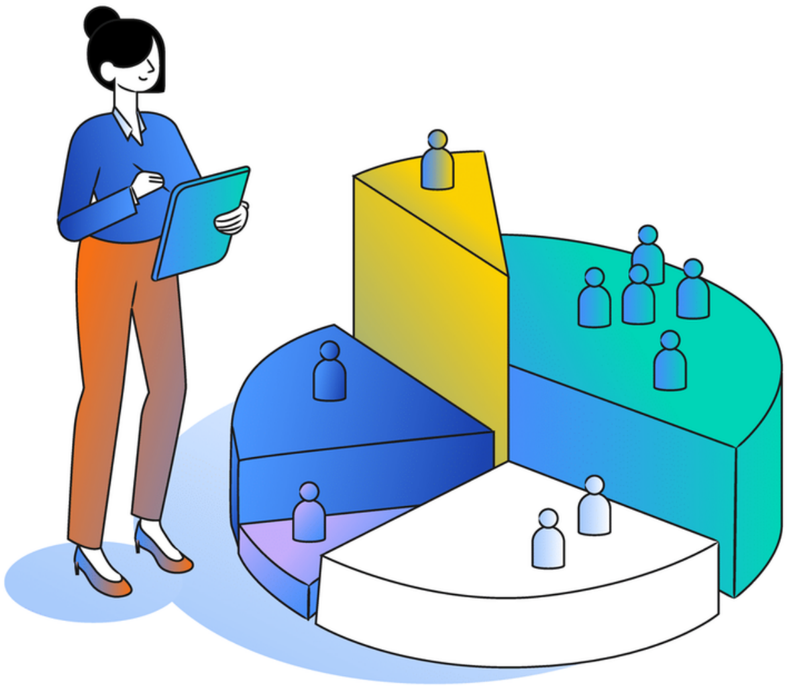
WORK PROCESS



Customer Segmentation

Main goal of the project

Aim: Customer analysis that aims at identifying groups of users with similar purchasing behaviors



CUSTOMER SEGMENTATION

DATA PREPARATIONS

MODELING

CHOOSING THE BEST MODEL

BUSINESS STRATEGIES

Datasets

For visualizations

Customers.csv

Sellers.csv

geo.csv

order_status.csv

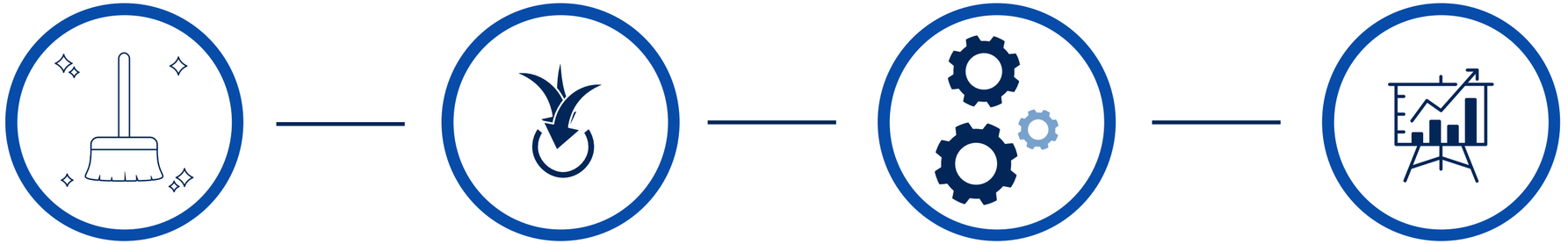
order_items.csv

order_payments.csv

products.csv

products_reviews.csv

Data Preparations



Data Cleaning

Only delivered orders
97% share

Cleaned and grouped each dataset before merging

Datasets Merging

Data that is directly dealing with customers (excluding sellers)

Feature Engineering

Data Visualizations

CUSTOMER SEGMENTATION

DATA PREPARATIONS

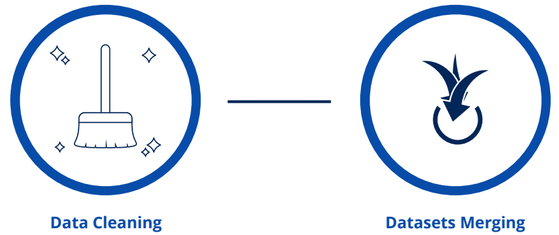
MODELING

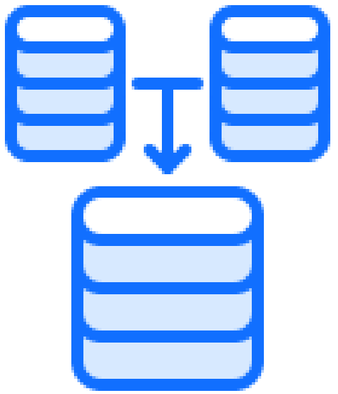
CHOOSING THE BEST MODEL

BUSINESS STRATEGIES

Data Preparations

Dataset	Data Preparation	Group By
order_items.csv	* drop missing order_id (5) * drop columns 'max_shipping_seller_date' & 'seller_id' we don't need them in this analysis	Group by: order_id order_item_sequence_id: max() to get number of items per order product_id: join by "," if the products are different price: sum, shipping_cost: sum
order_status.csv	* only kept delivered orders * convert the variables to their appropriate type (float & date) * deleted ts_order_delivered_carrier & 'ts_order_approved * dropped NA	-
order_payments.csv	* remove NAs (only 12) * merged with a column with the unique customer ids	Group by: unique_customer_id 'payment_installments_quantity': 'max', 'payment_method': join by "," if the payments are different 'transaction_value': 'sum', 'payment_method_sequence_id': 'max'
product_reviews.csv	* merged with a column with the unique customer ids	Group by: unique_customer_id 'review_score': count() to get the number of reviews each customer made
products.csv	* we created a new category where we grouped further the categories	-
customers_csv	* only kept the variables 'product_category_name','category_new','product_id'	Group by: unique_customer_id





merge our datasets to get our final, cleaned dataset

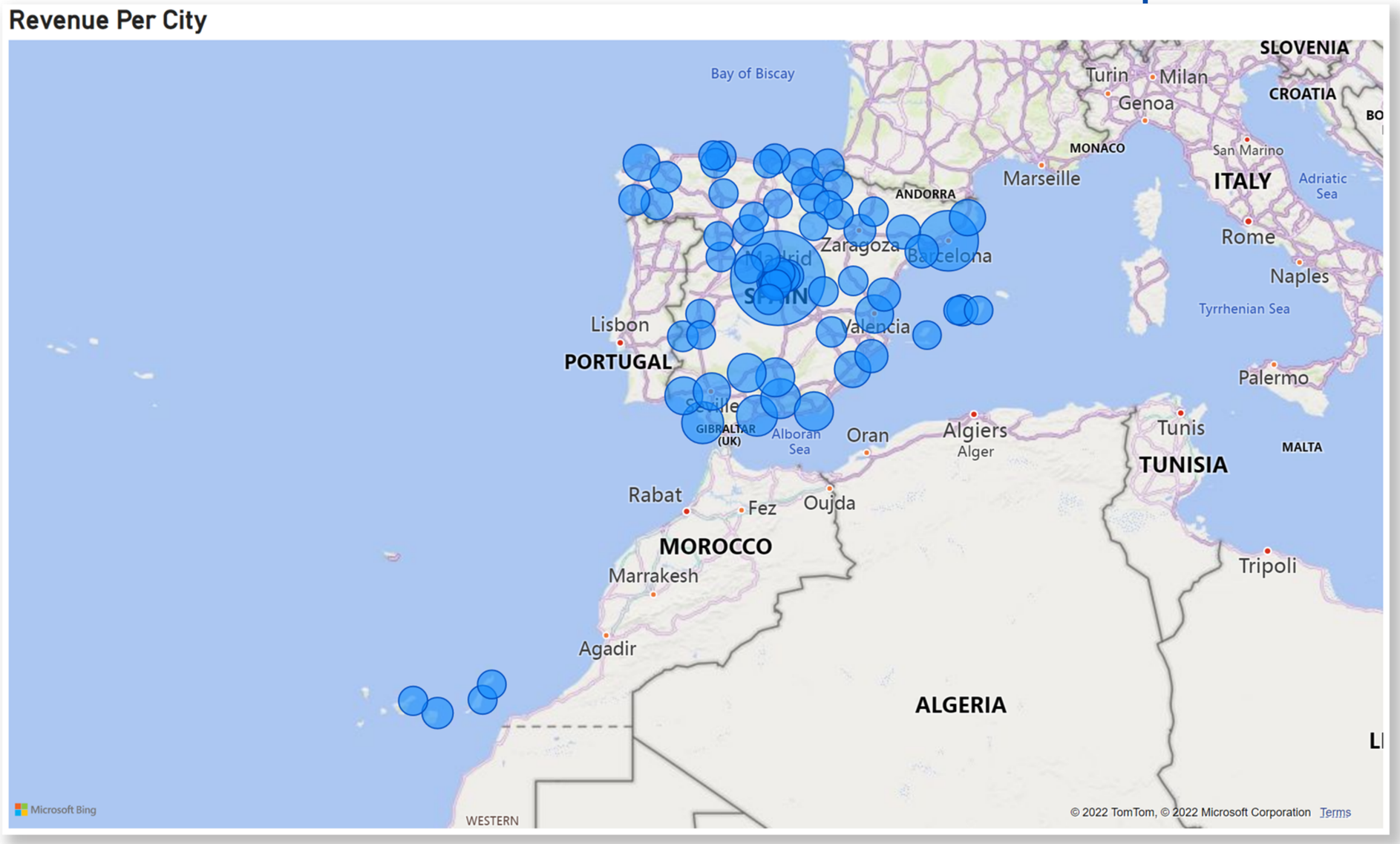
Data Preparations

Feature Engineering

Features	Description
Frequency	Number of orders per customer_unique_id
max_shipping	Maximum shipping cost is the maximum total shipping cost a unique customer spent per order
Tenure	Tenure is how long a customer used the app/website to purchase orders
Recency	How latest issue an X customer did, this is computed by taking the (latest date in the dataset)+1 and subtracting it by the latest date of each customer_unique_id
tot_ship/tot_spend	Total shipment spent per customer_unique_id over the Total_spending cost per customer_unique_id Intuition: the share of shipping cost compared to price of products purchased
new_diff	It is the maximum duration of estimated delivery from the each estimated wait per order_id (for a specific customer_unique_id)
Installment_payment	A binary variable; 0 for not paying is installment & 1 otherwise
del/rev_mean	The average time it takes for a customer to review bought product since the day of actual delivery
Payment_method	The payment method used per customer_unique_id; in this we can find a combination of method (e.g; Voucher, Credit Card)
tot_order_item_count	Total number of items bought by a unique customer
rev_count/tot_items_mean	Average number of reviews to the total items bought per unique customer

Visualizations

98K unique customers

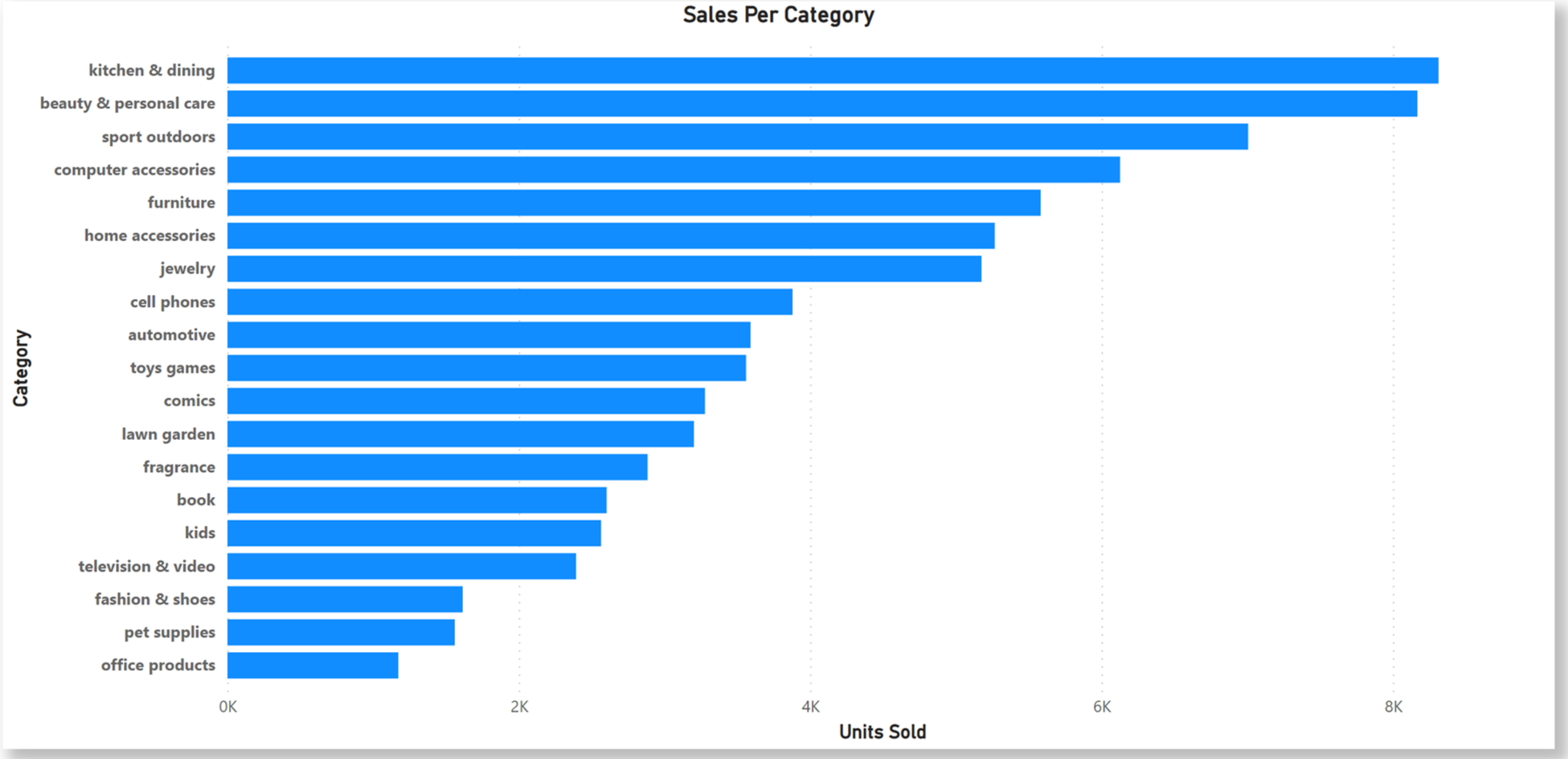


Data Visiualizations

Power BI

Visualizations

Units Sold Per Category

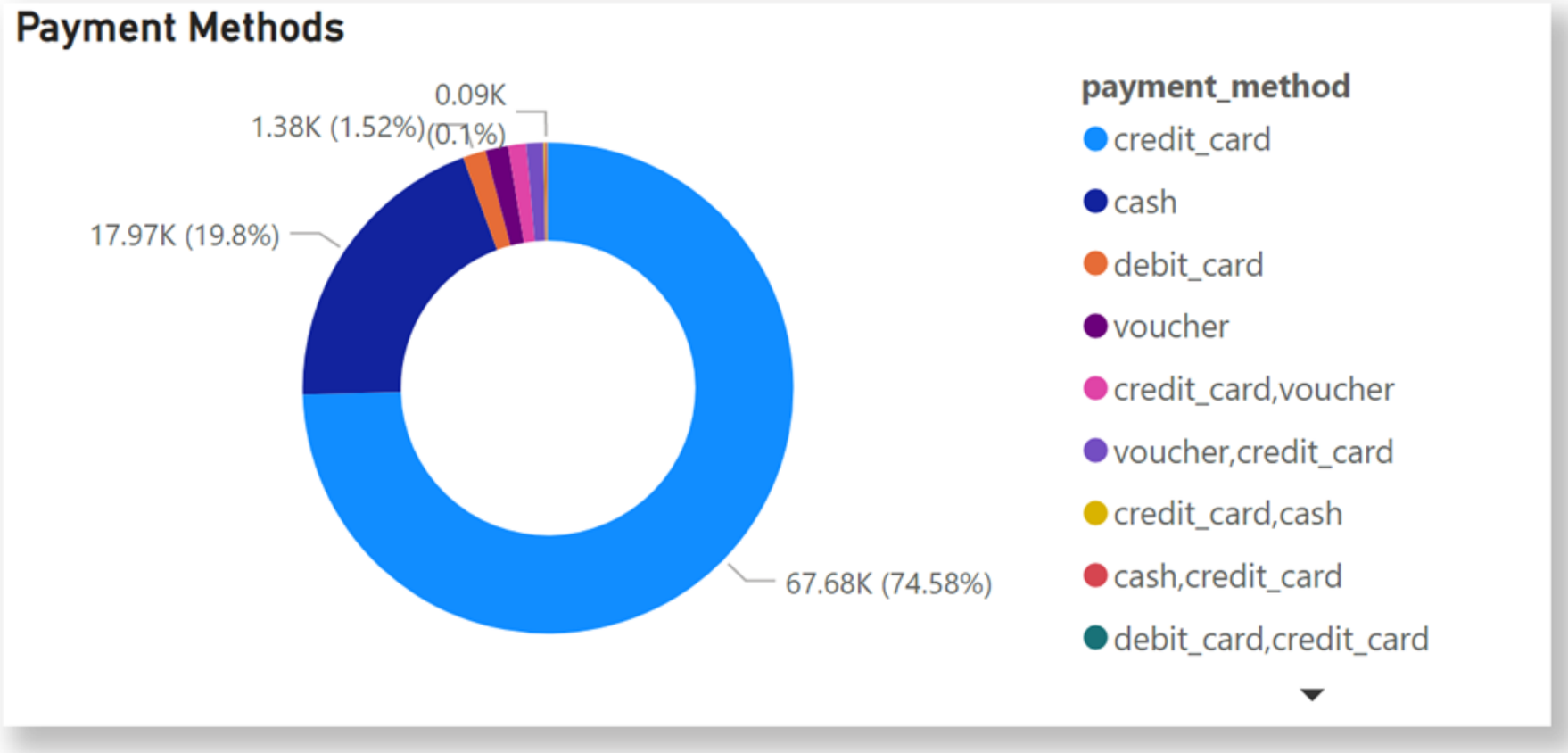


Data Visiualizations

Power BI

Visualizations

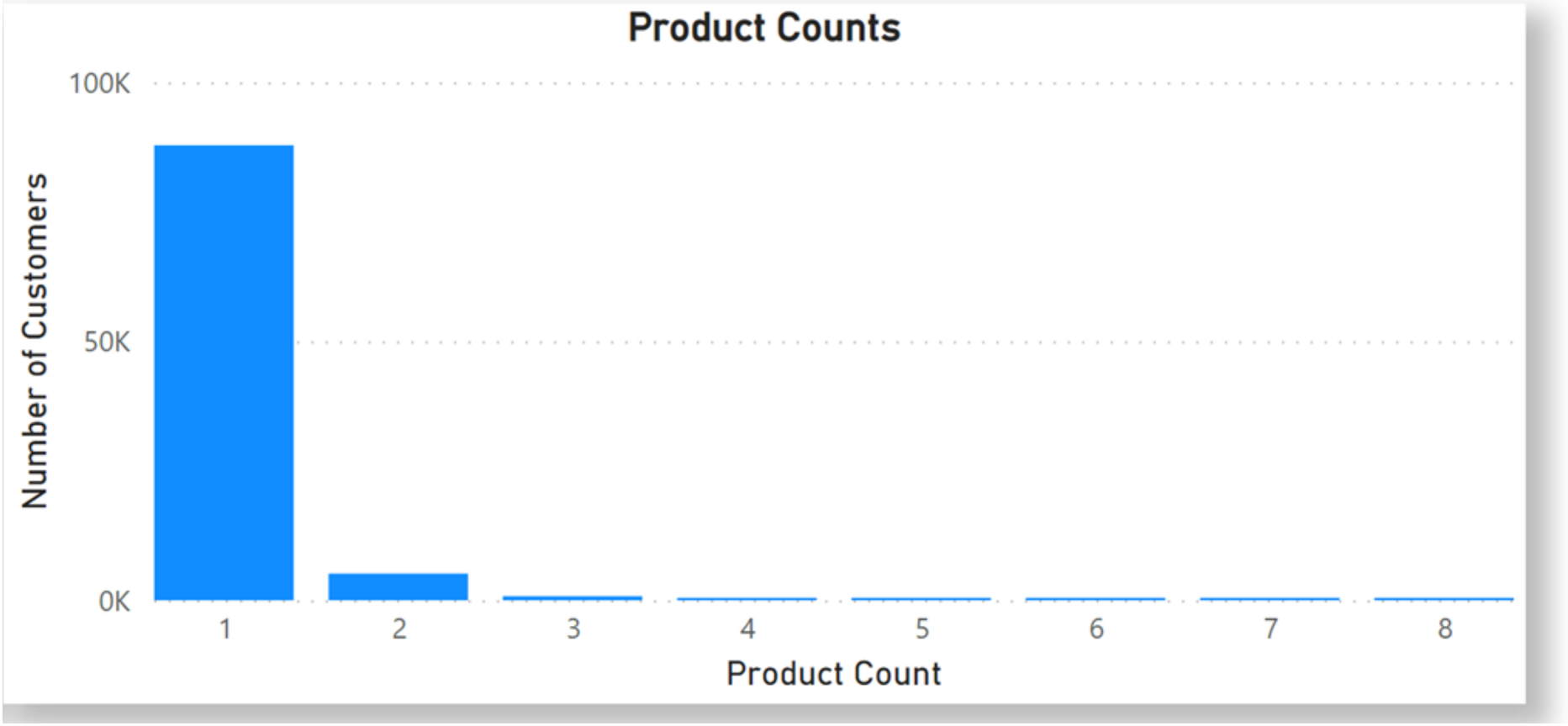
Data Visiualizations



Power BI

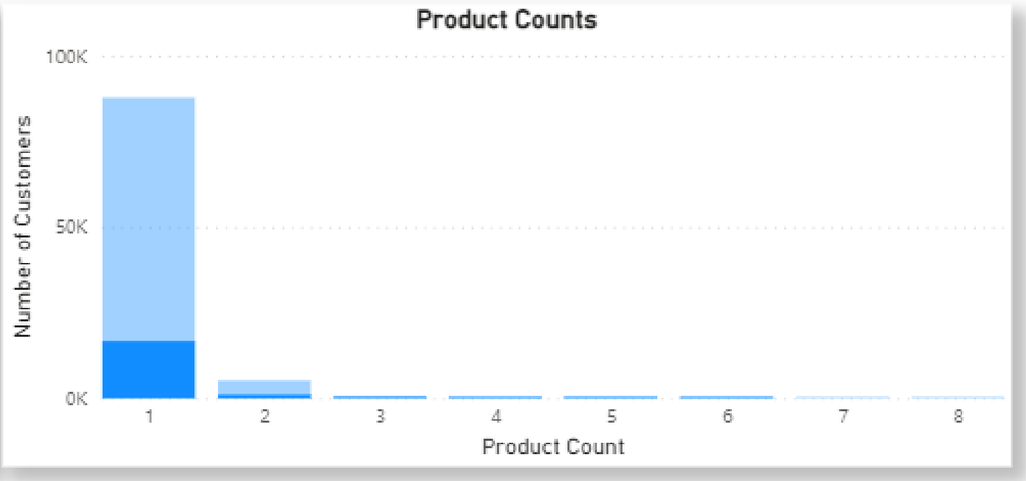
Visualizations

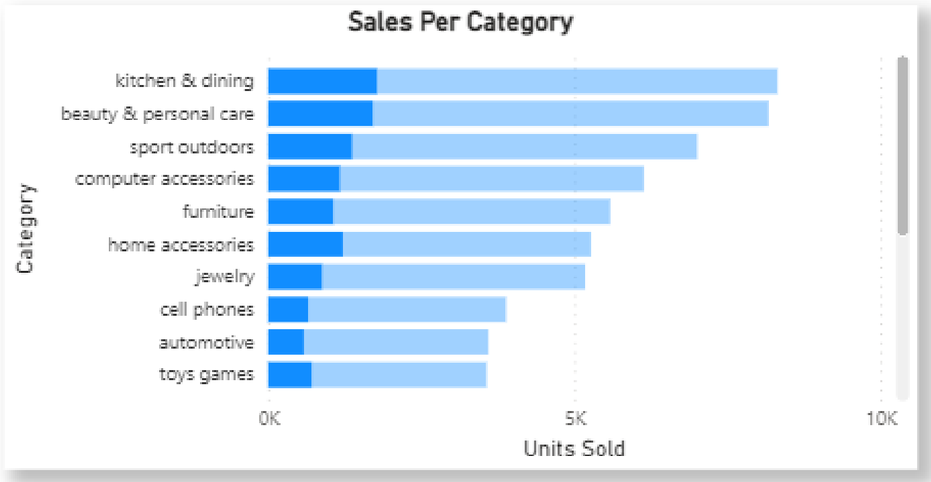
Data Visiualizations

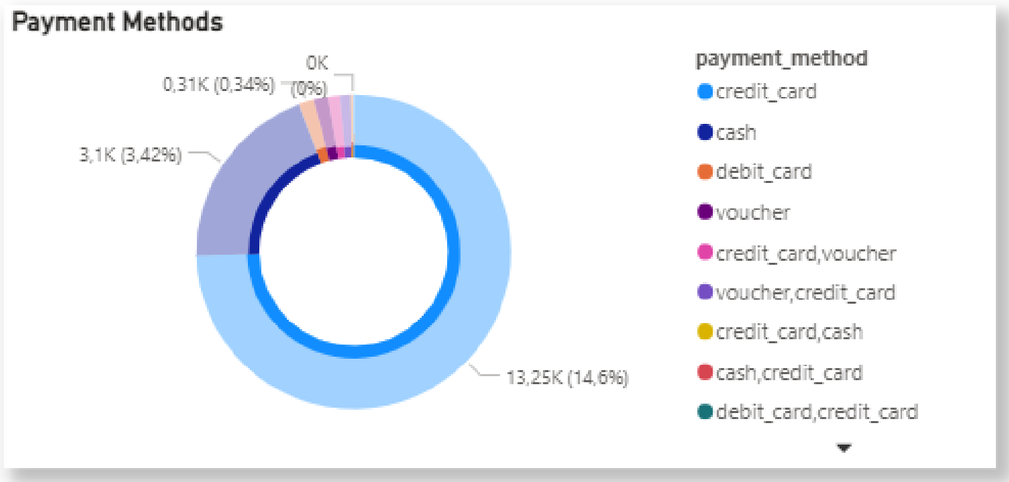


Power BI

Visualizations







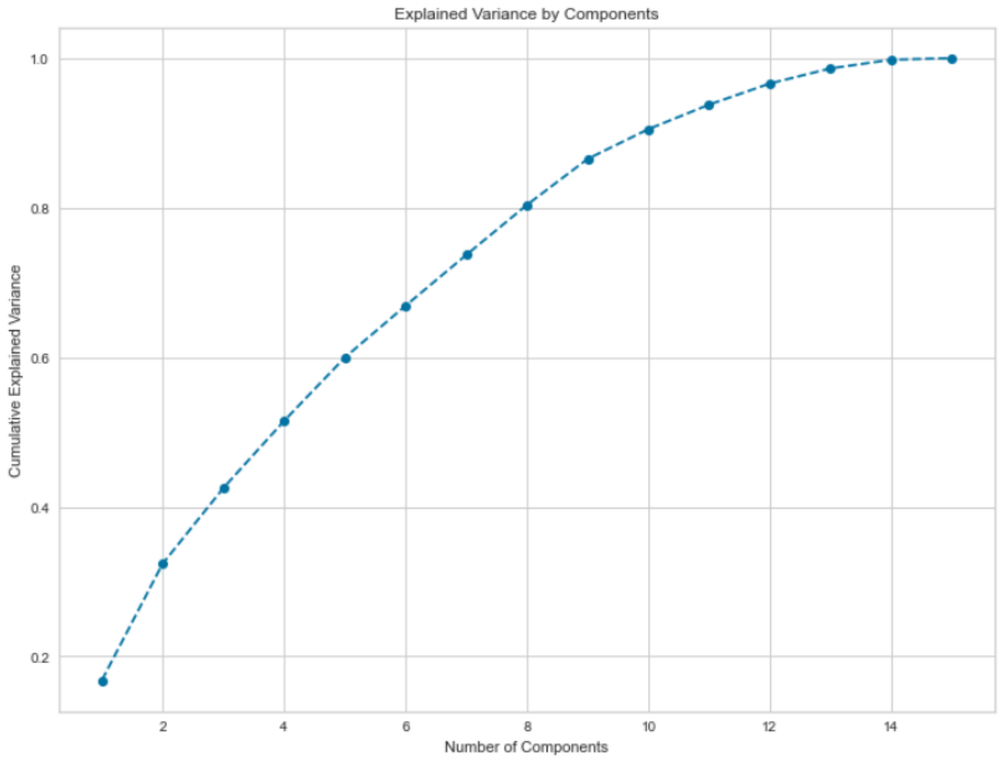
Data Visiualizations

Power BI

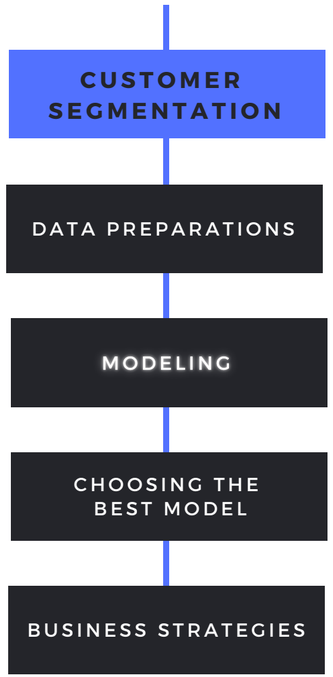
Modeling

Dimensionality Reduction: Principal Component Analysis

14 variables > We can reduce the dimensionality of our dataset before doing clustering

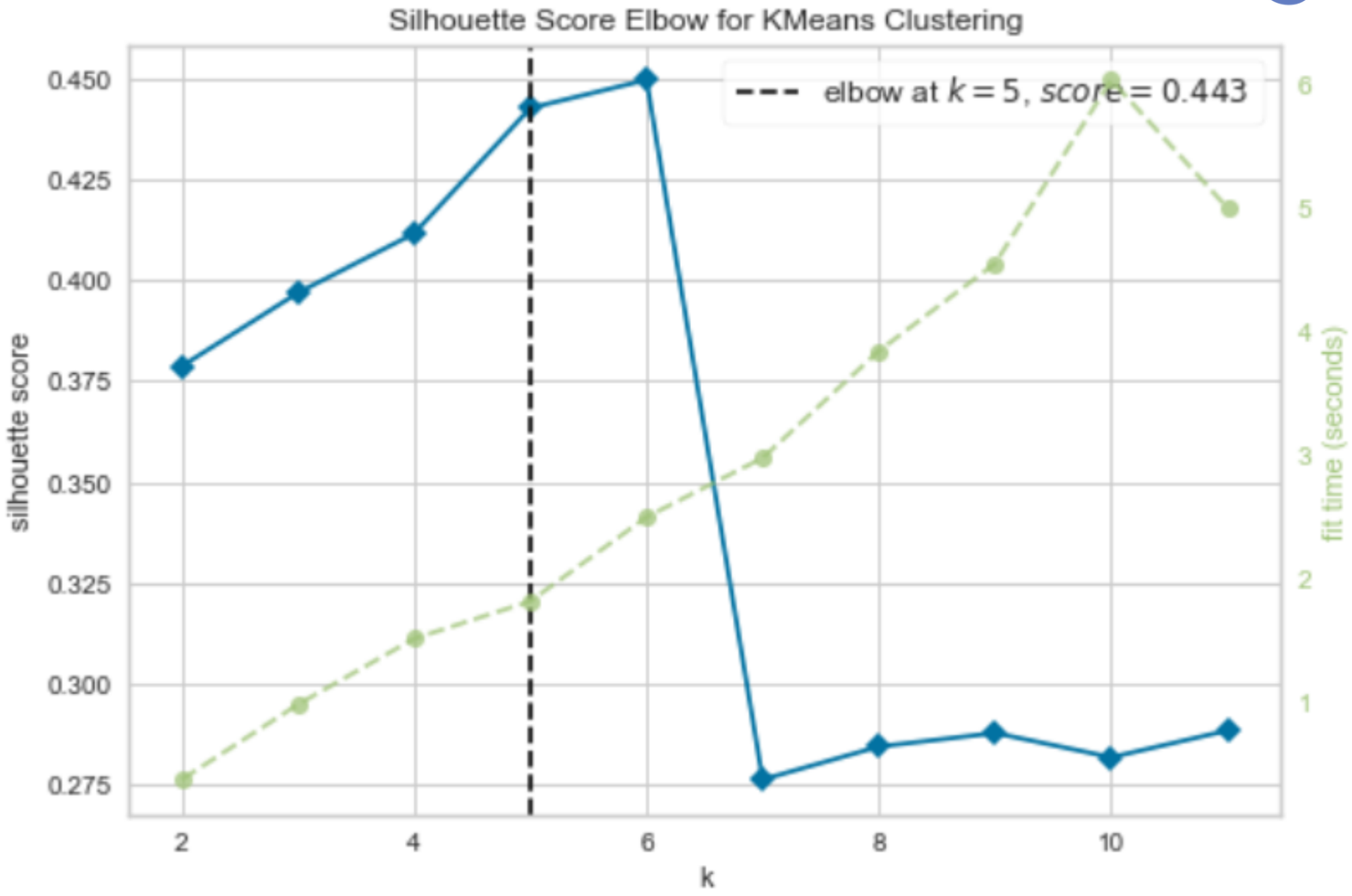


From 14 variables, we can work with
8 Principal Components

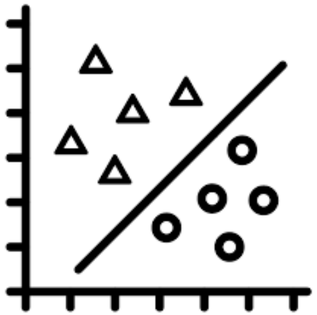


Modeling

Determining the best number of clusters (segments)



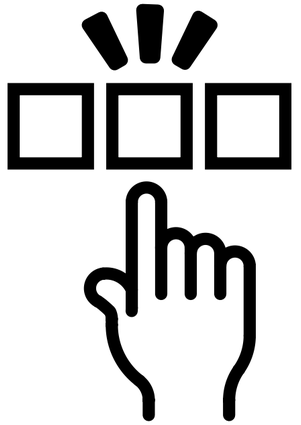
Modeling



3 chosen Clustering Algorithms were tested for this analysis

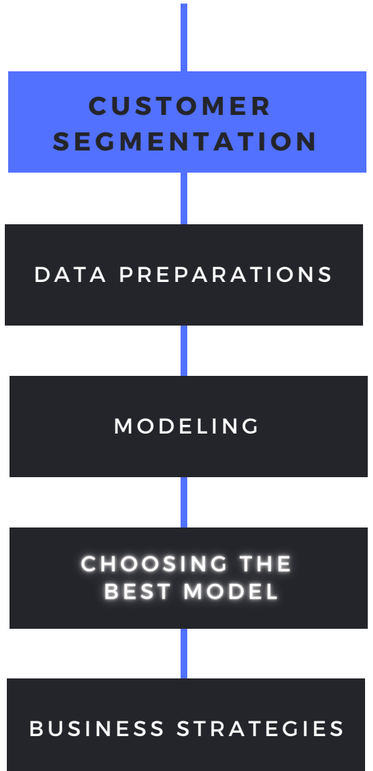
Choosing the best model? > **Silhouette Score**

Choosing The Best Model



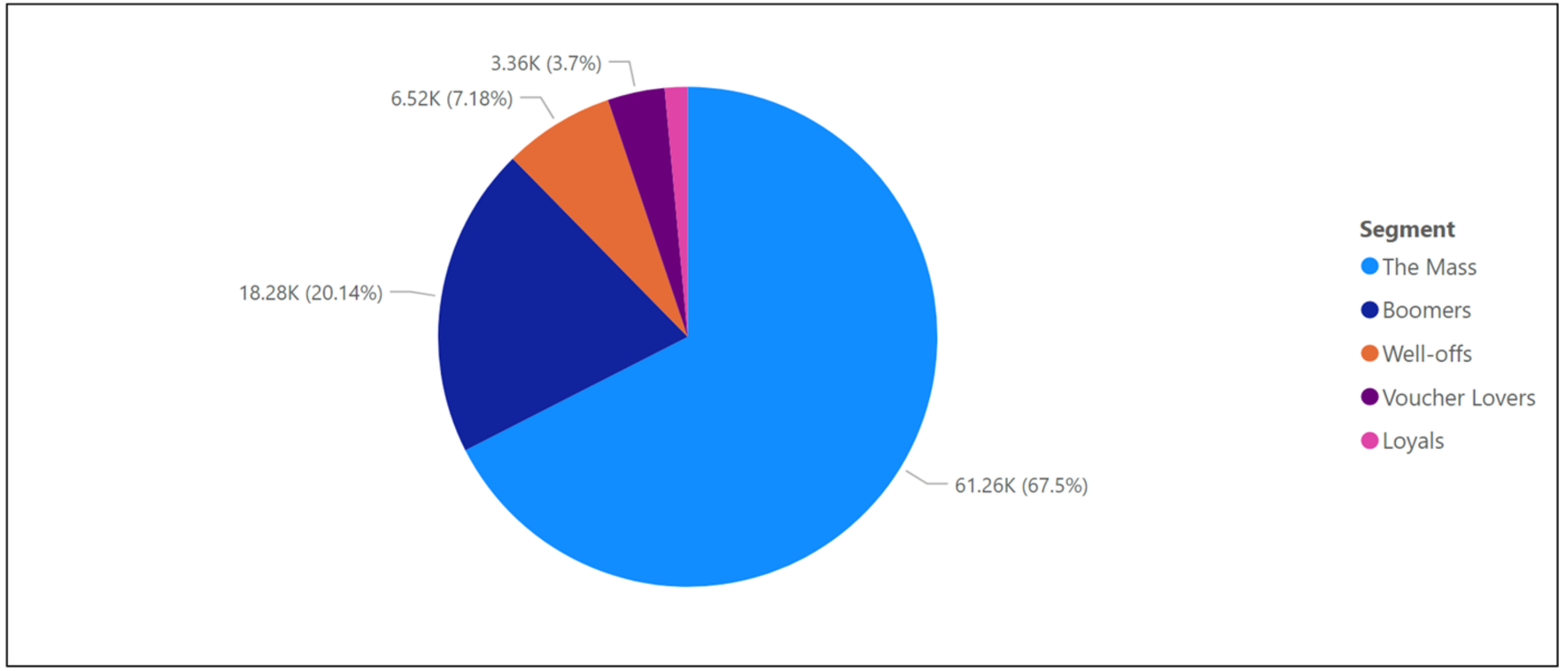
Choosing the best model: Silhouette Score

```
==========================================================================
Clustering    : KMeans++_pca   : silhouette score :   0.4426569994264603
==========================================================================
Clustering    : GMM_pca   : silhouette score :   0.39106239637145135
==========================================================================
Clustering    : BIRCH_pca   : silhouette score :   0.4079401826082869
==========================================================================
```



Our Segments

Proportions of Our Segments



CUSTOMER SEGMENTATION

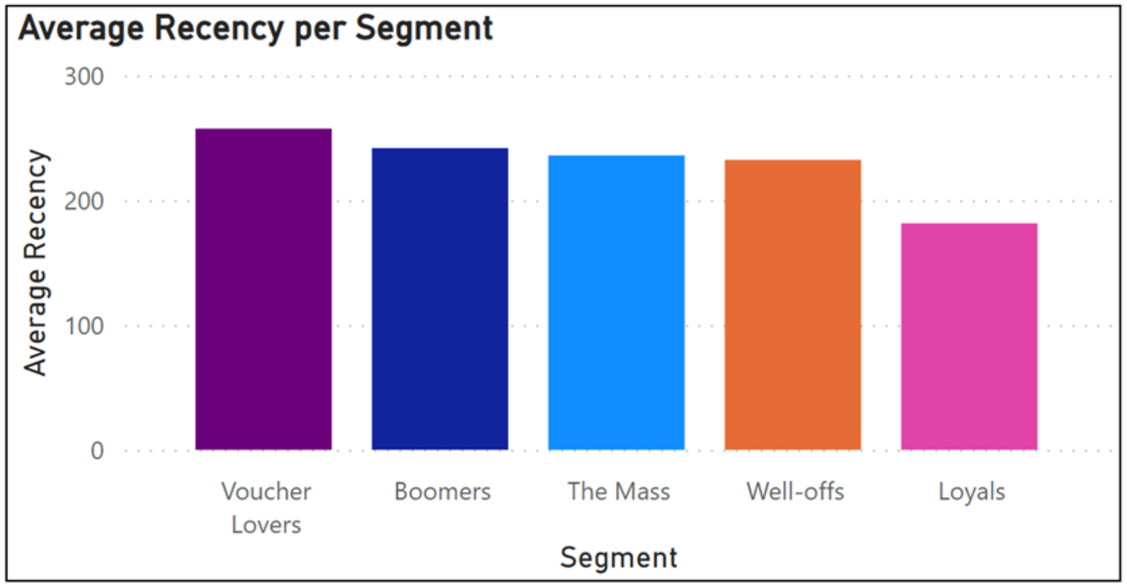
DATA PREPARATIONS

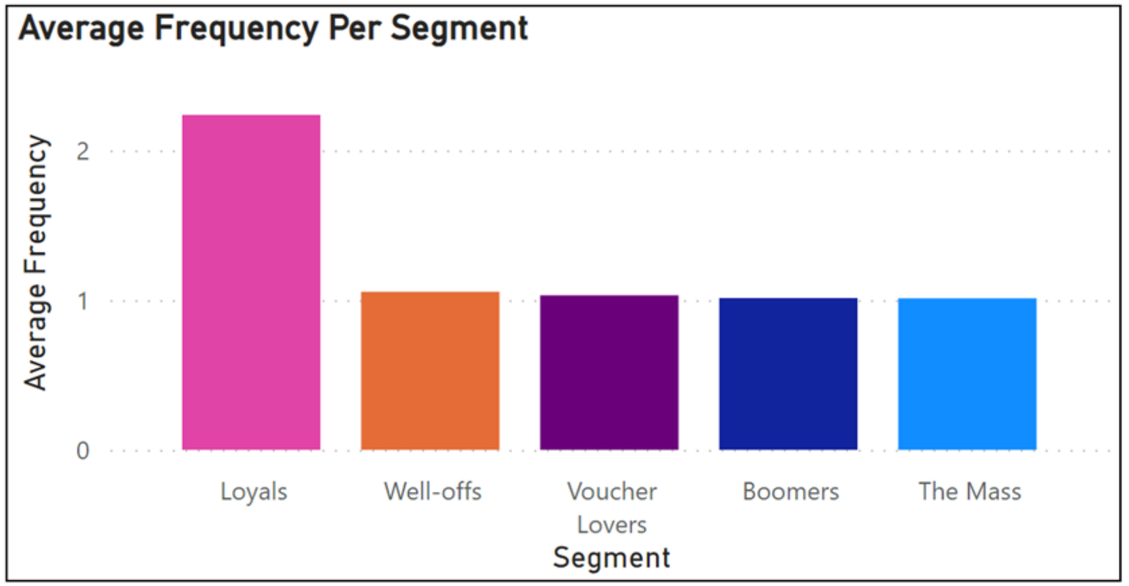
MODELING

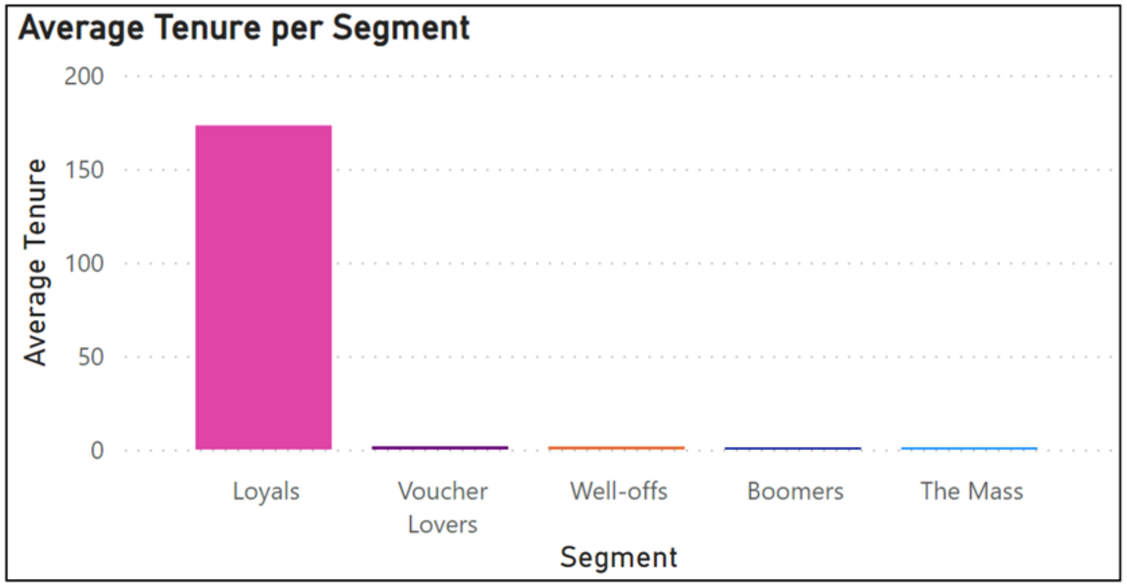
CHOOSING THE BEST MODEL

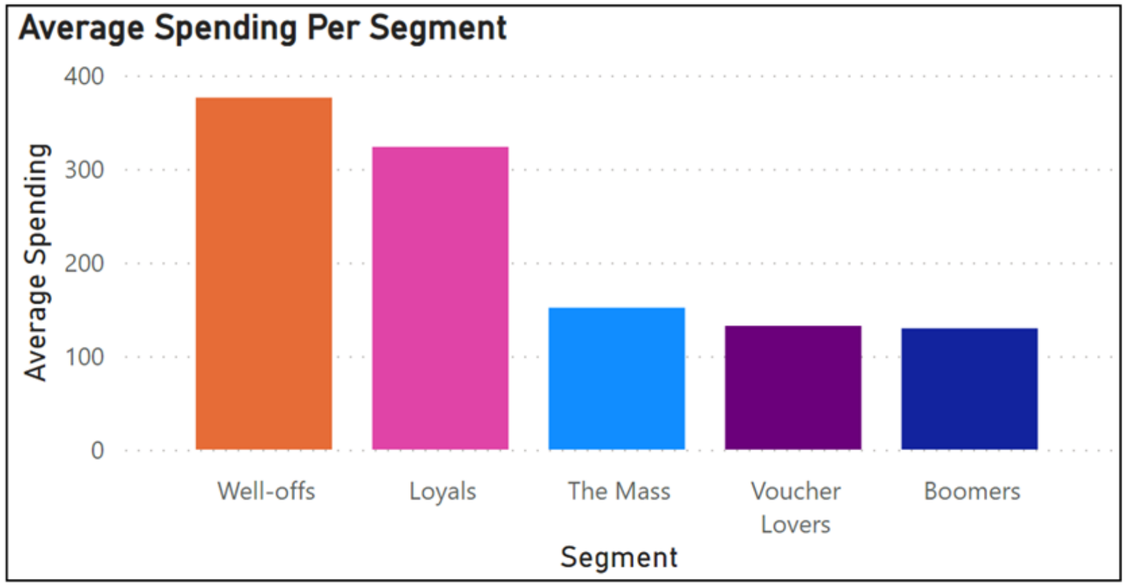
BUSINESS STRATEGIES

Our Segments

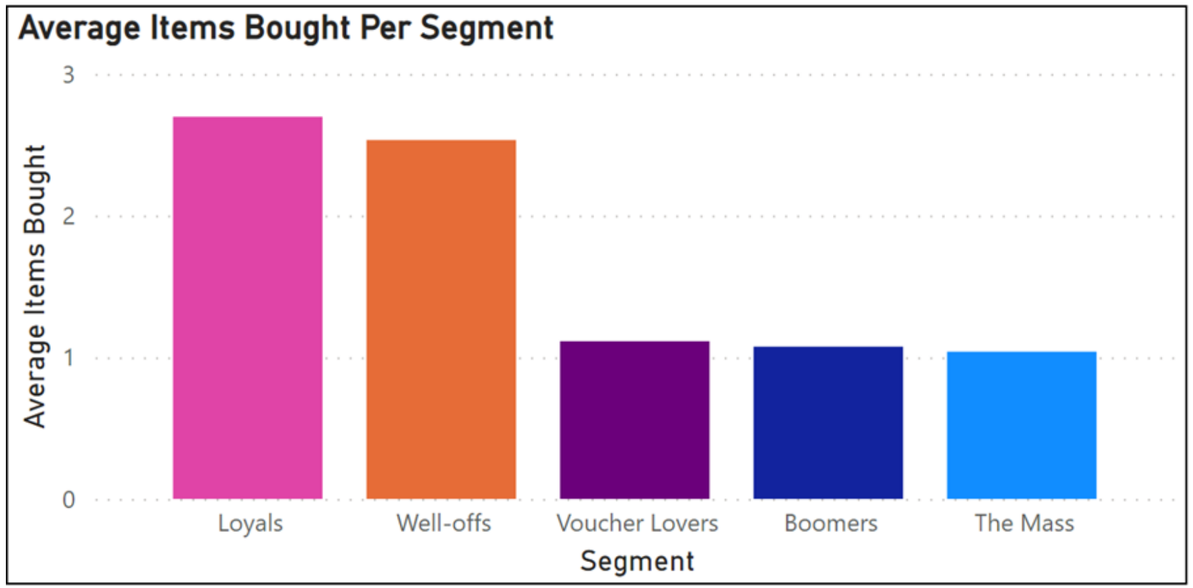


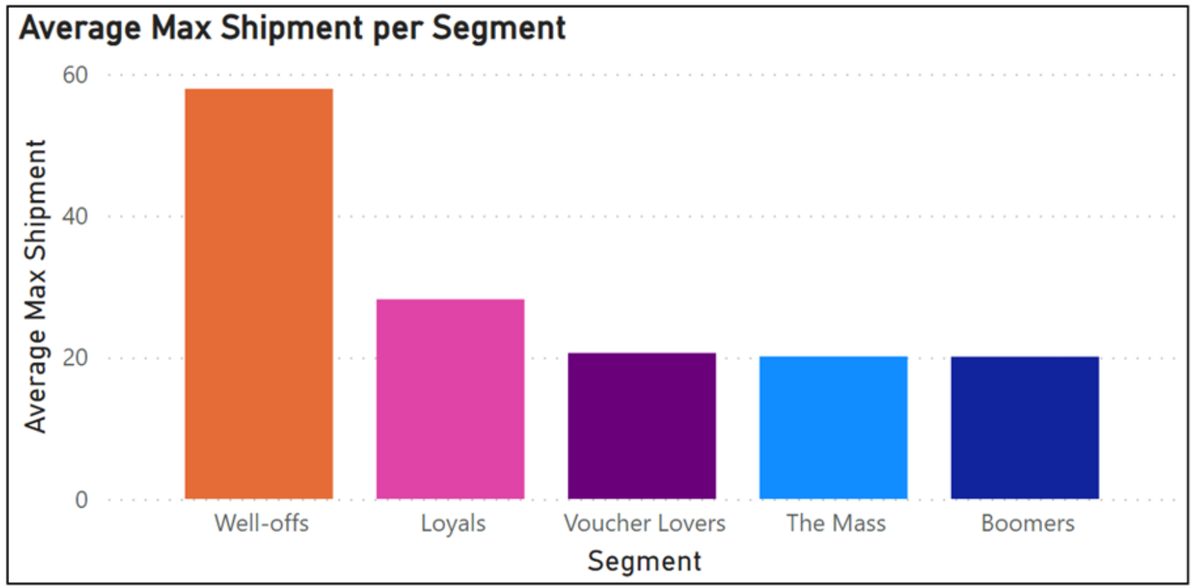


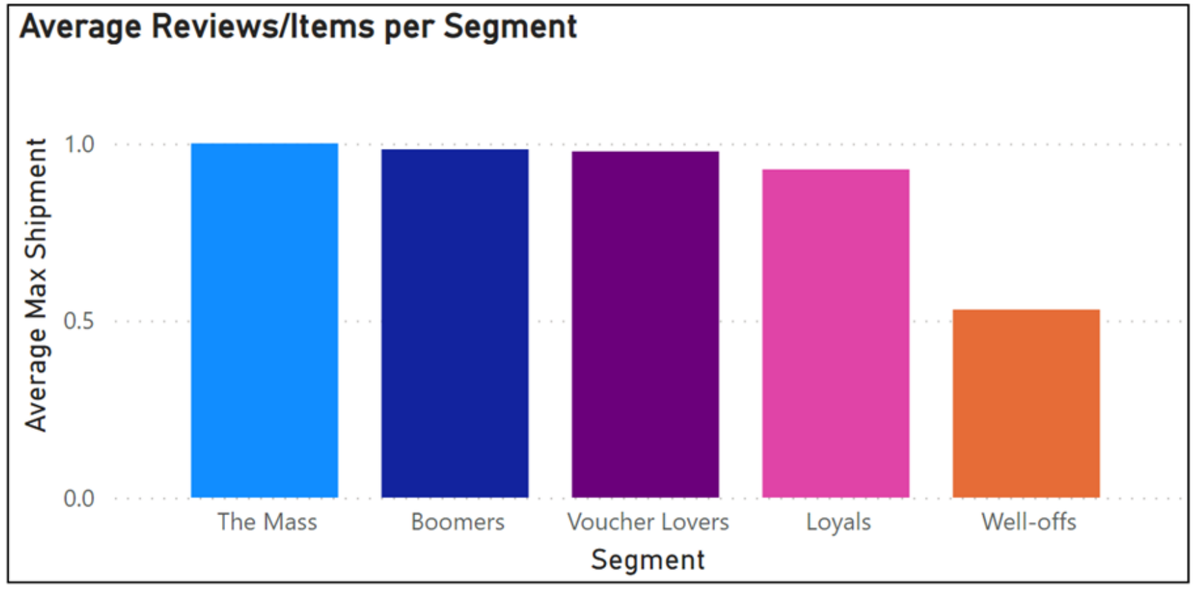




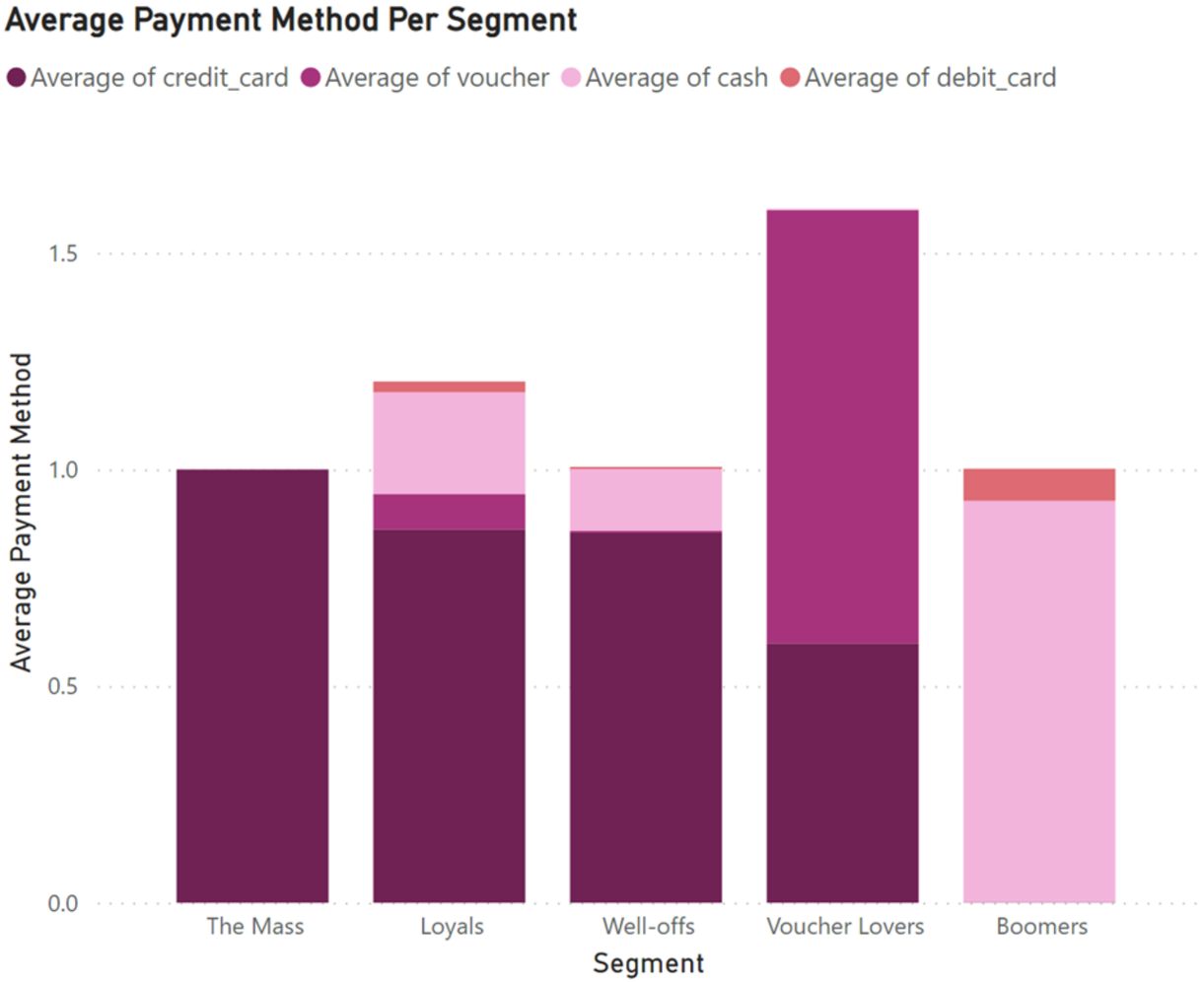
Our Segments

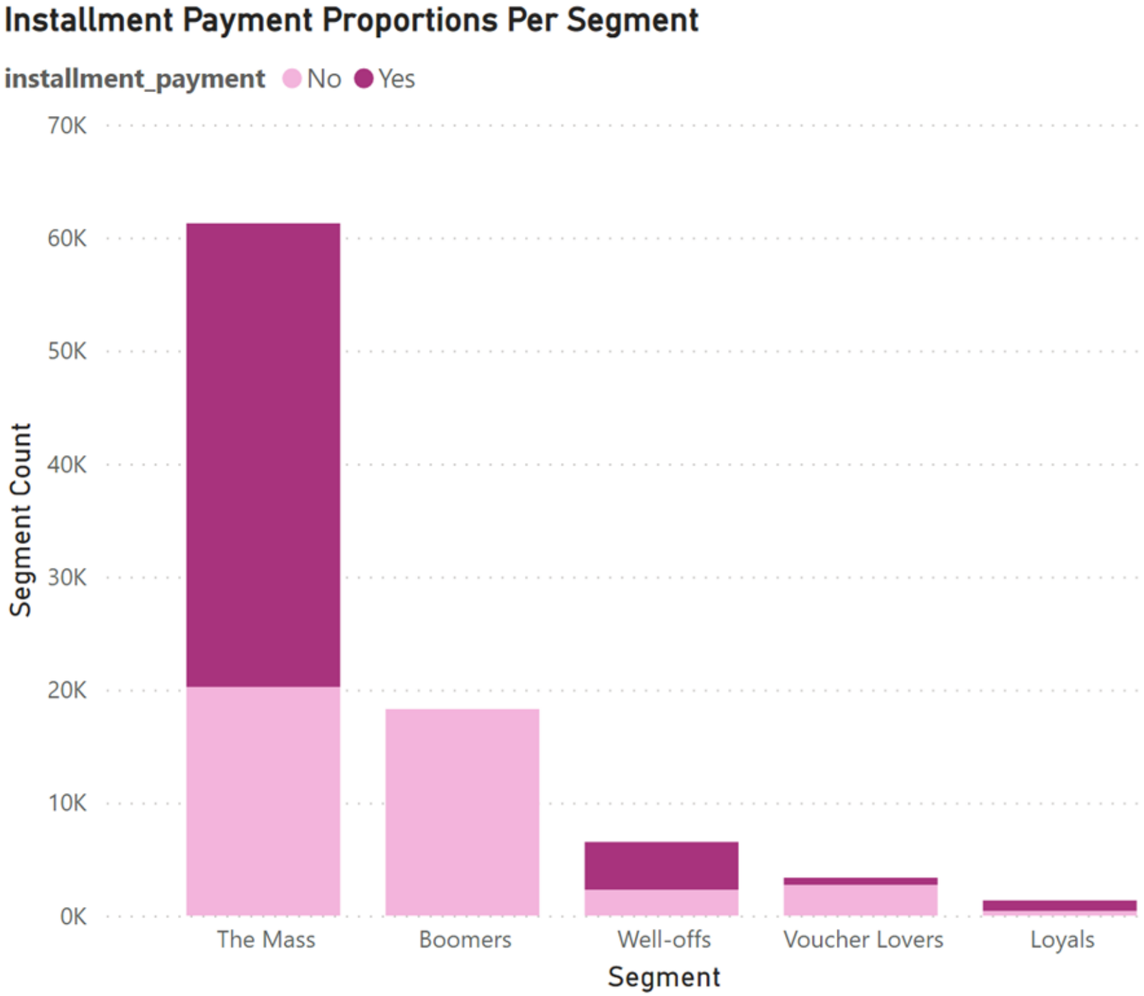






Our Segments





The segments' profiles: In Summary

Well-offs

- Most pay in installments (35% do not)
- They spend the most (also for the shipping)
- Not active in reviewing

Boomers

- Least to spend
- They only pay in cash or debit
- No installment payments
- Sensitive to shipment costs

Loyals

- Highest frequency & Recency (170)
- Pay in installments
- high TS
- Active reviewers
- They wait the most for the delivery
- pay in CC & cash

The Mass

- Bought 1 item on average
- Pay in installments
- Only use credit card
- Sensitive to shipping cost
- Active reviewers

Voucher lovers

- Bought 1 item
- They do not buy frequently
- They buy using vouchers
- The oldest in terms of recency

Business Recomendations

Well-offs

- Automated email with a special voucher /discount code to use
- special greetings through popups
- exclusive offers for, say, $40 orders or more.

Boomers

- Reduce shipping cost & add to price
- Facilitate installments using cash
- Make convenient products easily available

Loyals

- Strenghen the relationship
- Customized anniversary emails
- occasional special promos/gift card

The Mass

- Reduce shipping cost & add to price
- Make payment in installments possible
- Promote products with good reviews

Voucher lovers

- Activate them with vouchers
- Discount incentives
- Lotteries to win big vouchers
- Add "points" as feature

ISSUES WE DETECTED

ONLY 1.48% LOYAL CUSTOMERS

MOST CUSTOMERS MADE ONLY ONE PURCHASE AND HAVEN'T AGAIN IN OVER 200 DAYS

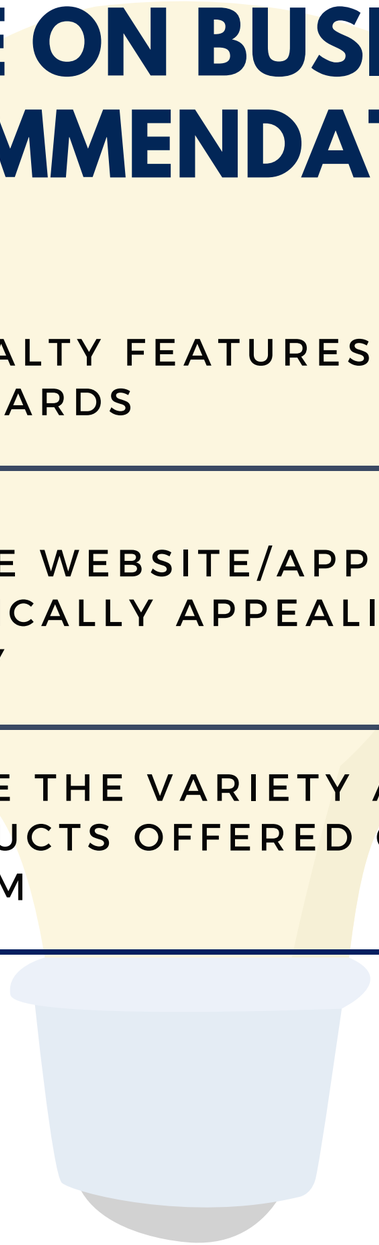
ONLY 3000 PRODUCTS OFFERED ON THE PLATFORM

MORE ON BUSINESS RECOMMENDATIONS

ADD LOYALTY FEATURES LIKE FIDELITY POINTS/CARDS

MAKE THE WEBSITE/APP AESTHETICALLY APPEALING & USER FRIENDLY

INCREASE THE VARIETY AND NUMBER OF PRODUCTS OFFERED ON THE PLATFORM



Market Basket Analysis

Uncover relationships and associations between products and categories of products

MARKET BASKET ANALYSIS

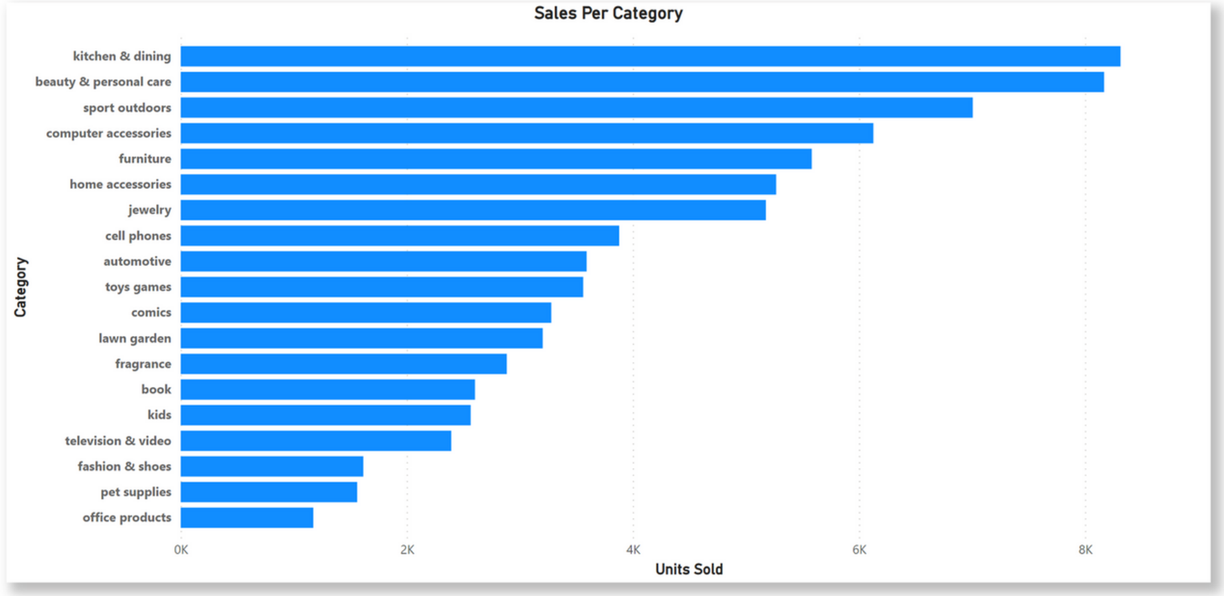
DATA PREPARATION

IMPLEMENTATION

RESULTS

Market Basket Analysis

Market basket analysis is a data mining technique used by retailers to increase sales by better understanding customer purchasing patterns. It involves analyzing large data sets, such as purchase history, to reveal product groupings, as well as products that are likely to be purchased together.



Which of these categories are being purchased together?

We decided to do this analysis on categories instead of products because they are interpretable

Market Basket Analysis

Data Preparation

1

order_id	product_category_name	order_item_sequence_id	product_id	price	shipping_cost
002f98c0f7efd42638ed6100ca699b42	toys games,videogame console	2	880be32f4db1d9f6e2bec38fb6ac23ab,d41dc2f2979f5...	53.89	39.73
005d9a5423d47281ac463a968b3936fb	kids,toys games	3	4c3ae5db49258df0784827bdacf3b396,fb7a100ec8c7b...	99.97	45.28
014405982914c2cde2796ddcf0b8703d	fragrance,sport outdoors	2	6782d593f63105318f46bbf7633279bf,e95ee6822b66a...	49.23	29.20
01b1a7fdae9ad1837d6ab861705a1fa5	kitchen & dining,home accessories	2	3fae92f8d0ebb3317991934a6d717c47,9b02b650be0a3...	108.99	26.85
01cce1175ac3c4a450e3a0f856d02734	book,lawn garden	2	9d0aa87e8df1bdbe0f79353520a2d538,415dfa57292b8...	96.23	27.51

- Grouped by order ID
- Combined product ids & category names
- Sum of price & shopping cost

2

	Other	automotive	beauty & personal care	bedroom decor	book	business office	cd vinyl	cell phones	coffee machines	comics	...	sport outdoors	television & video	tools home improvement	toys games	videogame
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	True	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	True	0
2	0	0	0	0	0	0	0	0	0	0	...	True	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
4	0	0	0	0	True	0	0	0	0	0	...	0	0	0	0	0

- One hot encoding
- create a matrix where cell = True if two categories have been purchased together

Market Basket Analysis

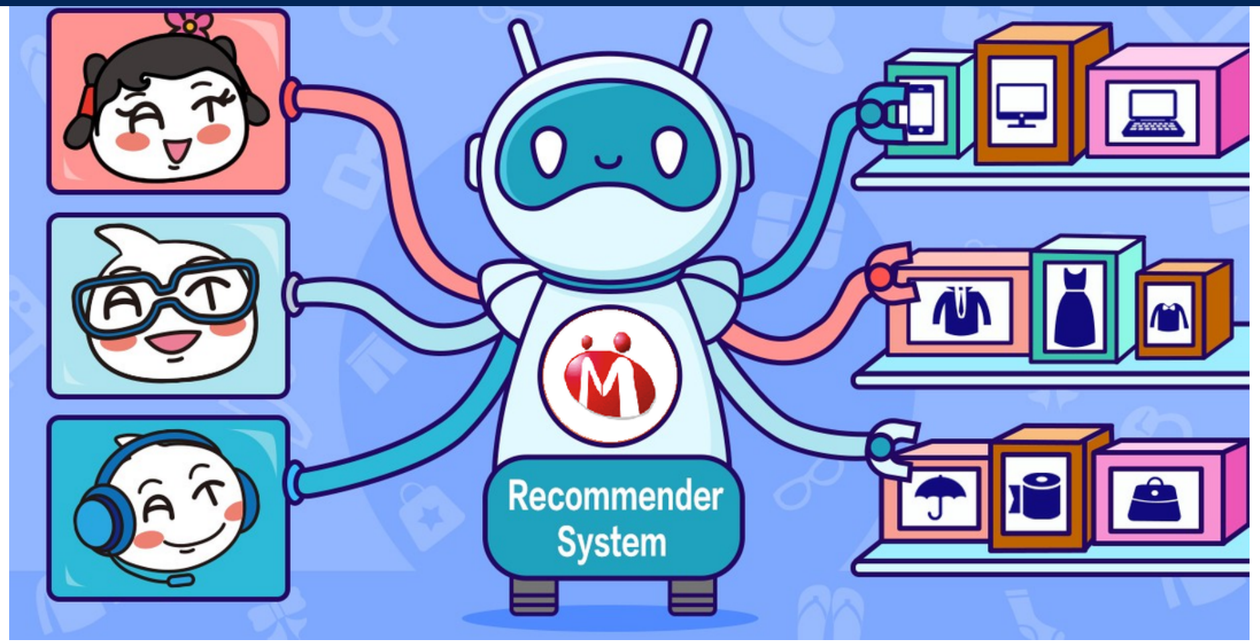
Results

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(fashion & shoes)	(Other)	0.025543	0.079183	0.011494	0.450000	5.683065	0.009472	1.674213
1	(Other)	(home accessories)	0.079183	0.148148	0.017880	0.225806	1.524194	0.006149	1.100309
2	(fragrance)	(beauty & personal care)	0.035760	0.091954	0.015326	0.428571	4.660714	0.012037	1.589080
3	(kids)	(comics)	0.118774	0.085568	0.025543	0.215054	2.513240	0.015379	1.164961
4	(comics)	(kids)	0.085568	0.118774	0.025543	0.298507	2.513240	0.015379	1.256216
5	(home accessories)	(furniture)	0.148148	0.263091	0.030651	0.206897	0.786408	-0.008325	0.929147
6	(furniture)	(kitchen & dining)	0.263091	0.260536	0.089400	0.339806	1.304255	0.020855	1.120070
7	(kitchen & dining)	(furniture)	0.260536	0.263091	0.089400	0.343137	1.304255	0.020855	1.121862
8	(lawn garden)	(furniture)	0.093231	0.263091	0.021711	0.232877	0.885158	-0.002817	0.960614
9	(tools home improvement)	(furniture)	0.063857	0.263091	0.030651	0.480000	1.824466	0.013851	1.417133
10	(kids)	(toys games)	0.118774	0.065134	0.024266	0.204301	3.136622	0.016529	1.174899
11	(toys games)	(kids)	0.065134	0.118774	0.024266	0.372549	3.136622	0.016529	1.404454
12	(wellness & relaxation)	(kitchen & dining)	0.063857	0.260536	0.054917	0.860000	3.300882	0.038280	5.281883
13	(kitchen & dining)	(wellness & relaxation)	0.260536	0.063857	0.054917	0.210784	3.300882	0.038280	1.186169

Recommendation System

Second Goal

Aim: able to provide to each user a set of products related to their interests, in order to maximize the probability of purchase

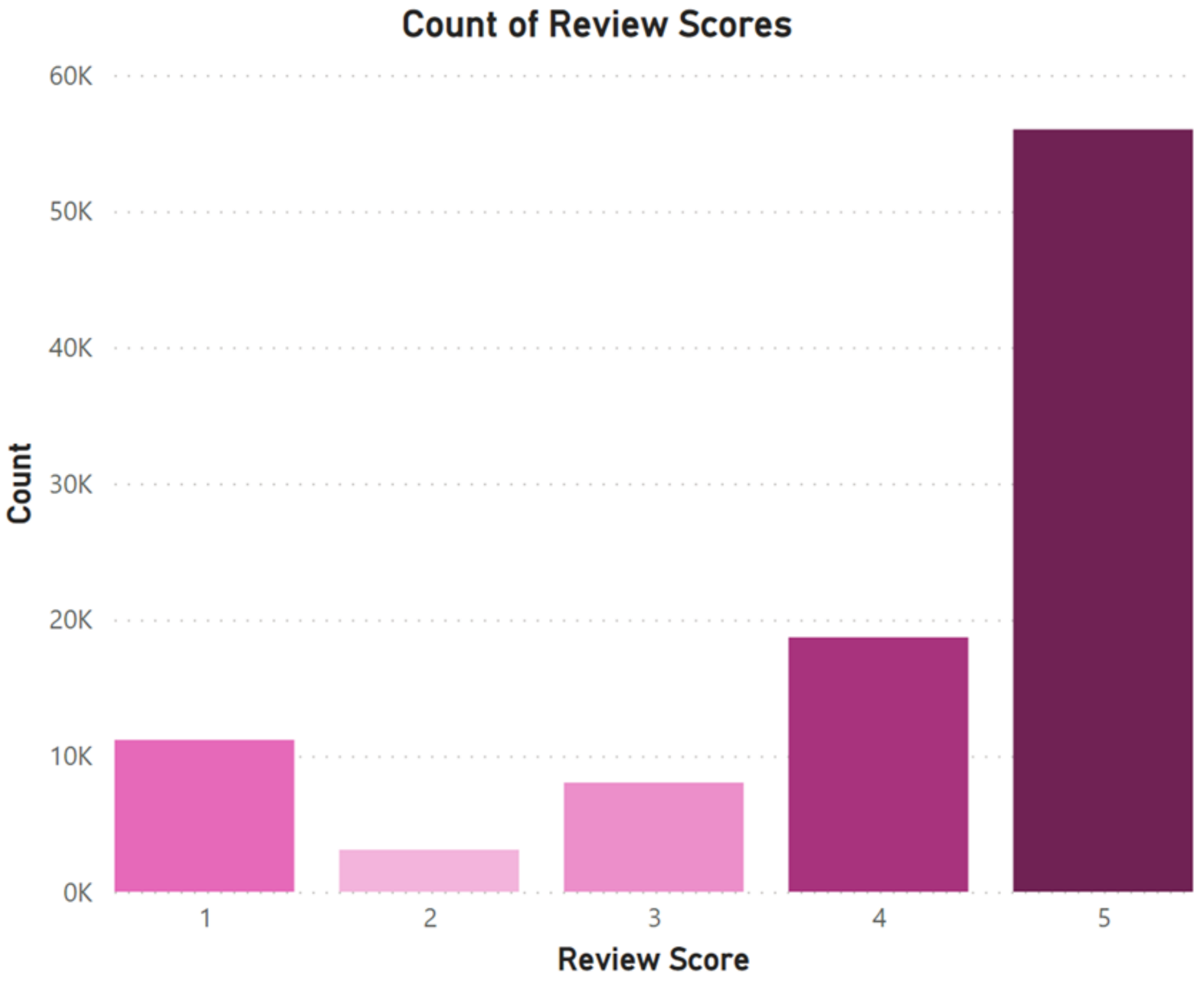


RECOMMENDATION SYSTEM

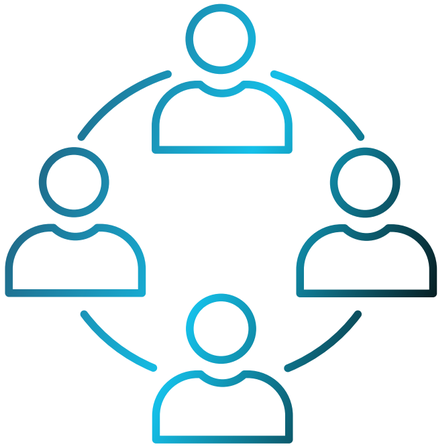
DATA PREPARATION

MODELING

RESULTS


Count of Review Scores
60K
50K
40K
30K
20K
10K
0K
Count
1
2
3
4
5
Review Score

Data Preparation

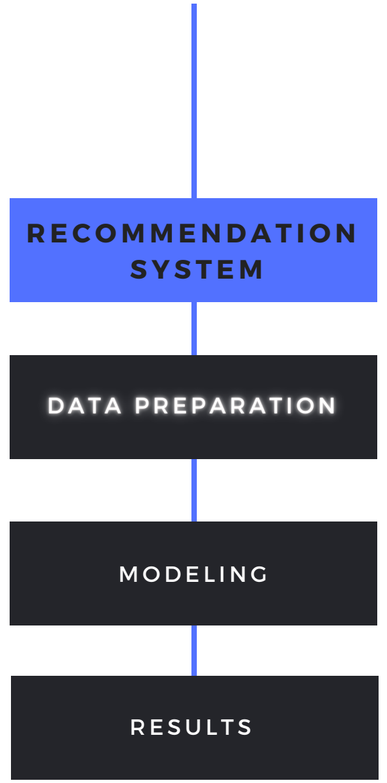


We kept customers who rated at least 2 products & products that have been rated by at least 5 customers
We create our matrix:

```
reviewmatrix.head()
```

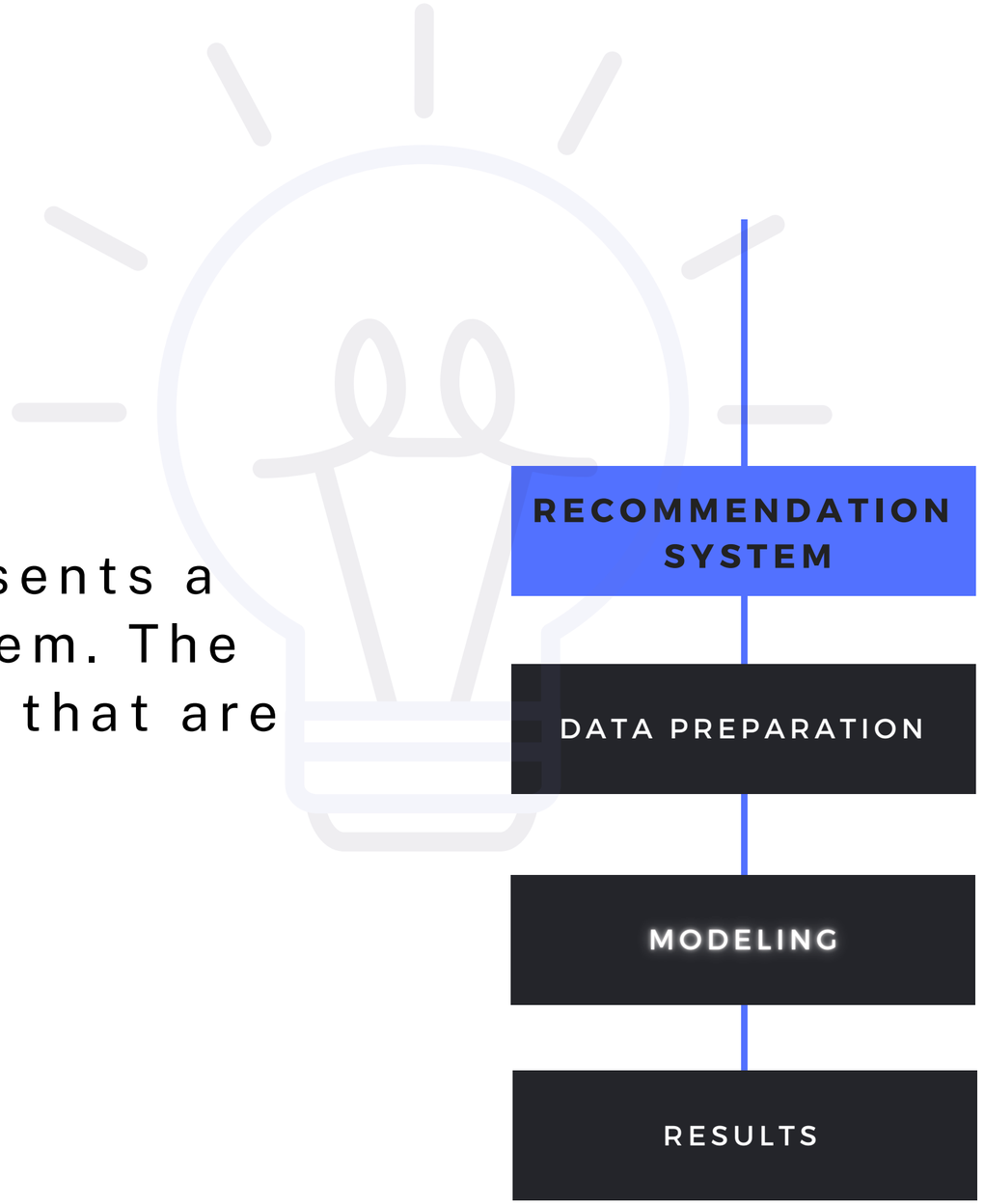
product_id customer_unique_id	001b72dfd63e9833e8c02742adf472e3	005030ef108f58b46b78116f754d8d38	008cff0e5792219fae03e570f980b330	009af1277432f1a(
0025795df7a7d077c4c90162fa820085	0.0	0.0	0.0	
005754cb59f8ee5cdbb5cda9d138757e	0.0	0.0	0.0	
00a39521eb40f7012db50455bf083460	0.0	0.0	0.0	
00ae50eb5e1d2514f694dee1dcbbd5ae	0.0	0.0	0.0	
00ec23a308504080697e5204d3dbcb2c	0.0	0.0	0.0	

3624 rows × 2225 columns



Modeling

- Singular Value Decomposition (**SVD**)
- Uses Collaborative filtering
- Matrix structure where each row represents a user, and each column represents an item. The elements of this matrix are the ratings that are given to items by users.



Results

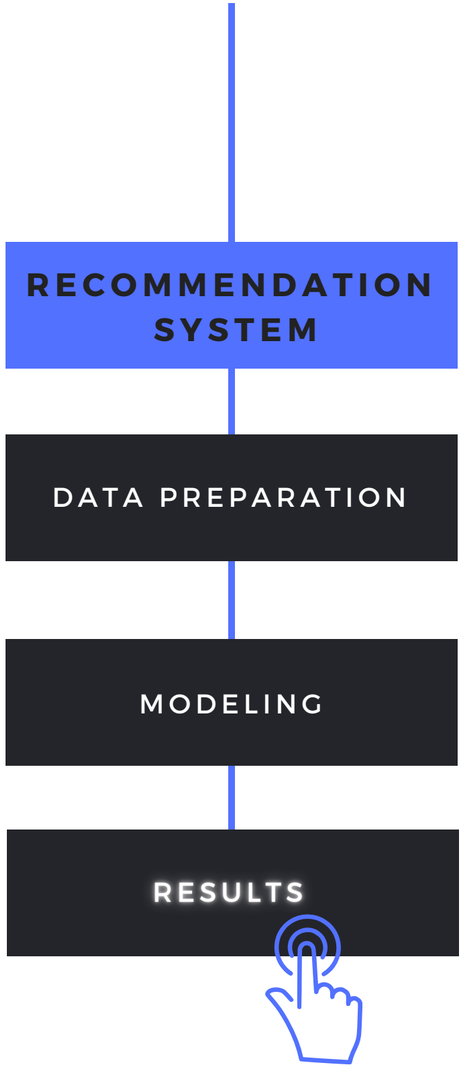
```
# Find the highest similarity
def cosine_similarity(v,u):
    return (v @ u)/ (np.linalg.norm(v) * np.linalg.norm(u))

highest_similarity = -np.inf
highest_sim_col = -1
for col in range(1,vh.shape[1]):
    similarity = cosine_similarity(vh[:,0], vh[:,col])
    if similarity > highest_similarity:
        highest_similarity = similarity
        highest_sim_col = col

print("Column %d (item id %s) is most similar to column 0 (item id %s)" %
        (highest_sim_col, reviewmatrix.columns[col], reviewmatrix.columns[0]))
```

```
Column 1 (item id ffd4bf4306745865e5692f69bd237893) is most similar to column 0 (item id 001b72dfd63e9833e8c02742adf472e3)
```

- We try to understand more the items by seeing under which category they belong

```
In [55]: rec[rec["product_id"]=="ffd4bf4306745865e5692f69bd237893"].product_category_name.unique() , rec[rec["product_id"]=="00210e41
```

```
Out[55]: (array(['fashion & shoes'], dtype=object),
          array(['beauty & personal care'], dtype=object))
```



EXTENSIONS

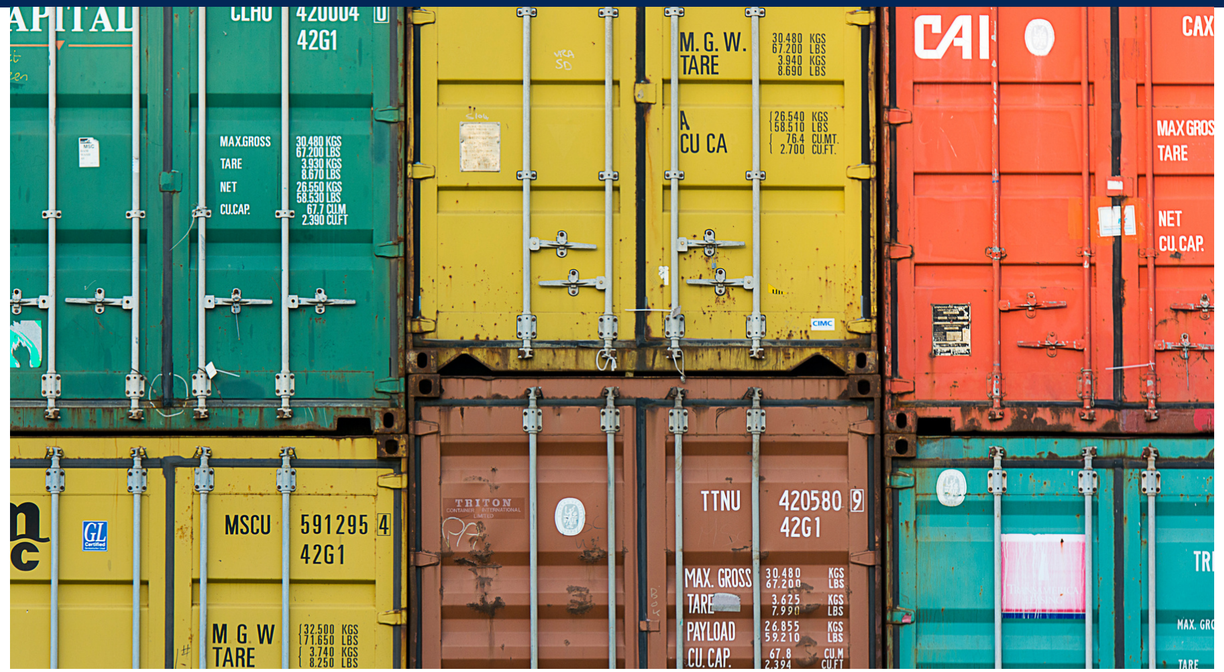
NATURAL LANGUAGE PROCESSING

Possible approaches for recommender system, use the names of previous multiple products purchases we recommend the next item via NLP (semantic relation) of the items

Time Shipment Analysis

Third Goal

Aim: predict an accurate date of delivery and to reduce delays, in order to improve the quality of the shipping service (Delivery Time Prediction)



TIME SHIPMENT ANALYSIS

EDA

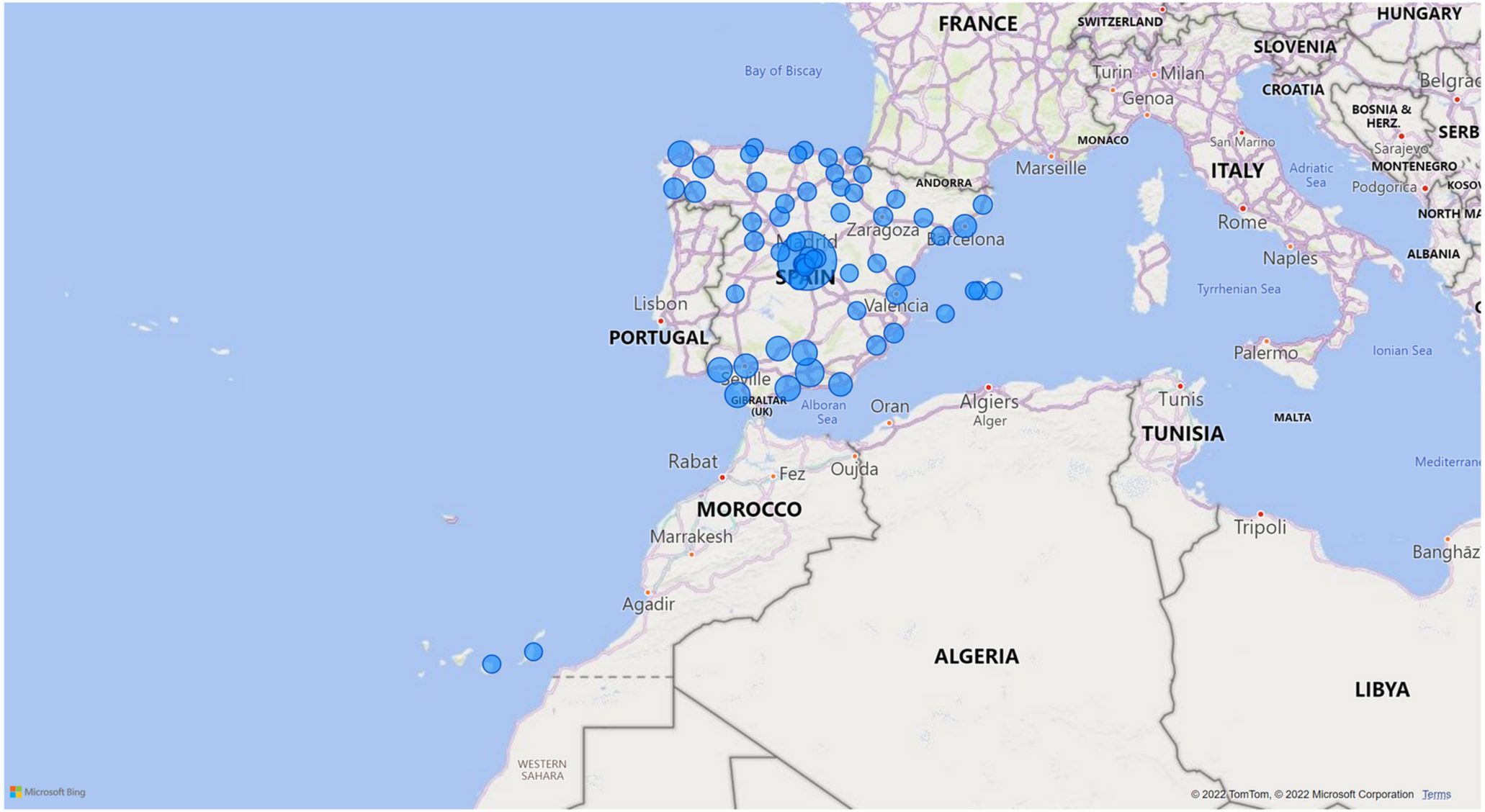
DATA PREPARATION

MODELING

RESULTS

Exploratory Data Analysis

Geographical Location of Sellers



Data Preparations

Dataset used for the analysis:

- customers
- geo
- sellers
- items
- order status

Merged in one Dataset "time_shipment" in which new variables were computed

1. Checked for null values,
2. All characters converted to lower case
3. Converted all letters to latin alphabet
4. Since geo_autonomous_community and seller_autonomous_community are the same, replace the name to seller_autonomous community for merging
5. Same thing for variable geo_city and seller_city
6. Merging of customer and geo dataset
7. Merged the previous dataset and order status using seller_id as key
8. Creation of a final dataset called "time_shipment" with the informations about both customer and sellers and the orders made

Found garbage data while cleaning:
Removed negative values when computing for the duration carrier_date - purchase_date

New variables computed :

1. "actual_wait" : delivery date to customer - day of the purchase
2. "estimated _wait": estimated delivery date- day of the purchase
3. "diff": the difference between estimated and actual wait to get an insight of the over/under estimations of the latter
4. "distance": computed using latitude and longitude of customers and sellers with the Harvesine formula
5. "op_carrier": difference between when the order was delivered to the carrier, and when it was purchased.

What emerged?

- On average the time for delivery is of 12 days
- The company's estimated wait is of 23 days
- 11 days over-estimation

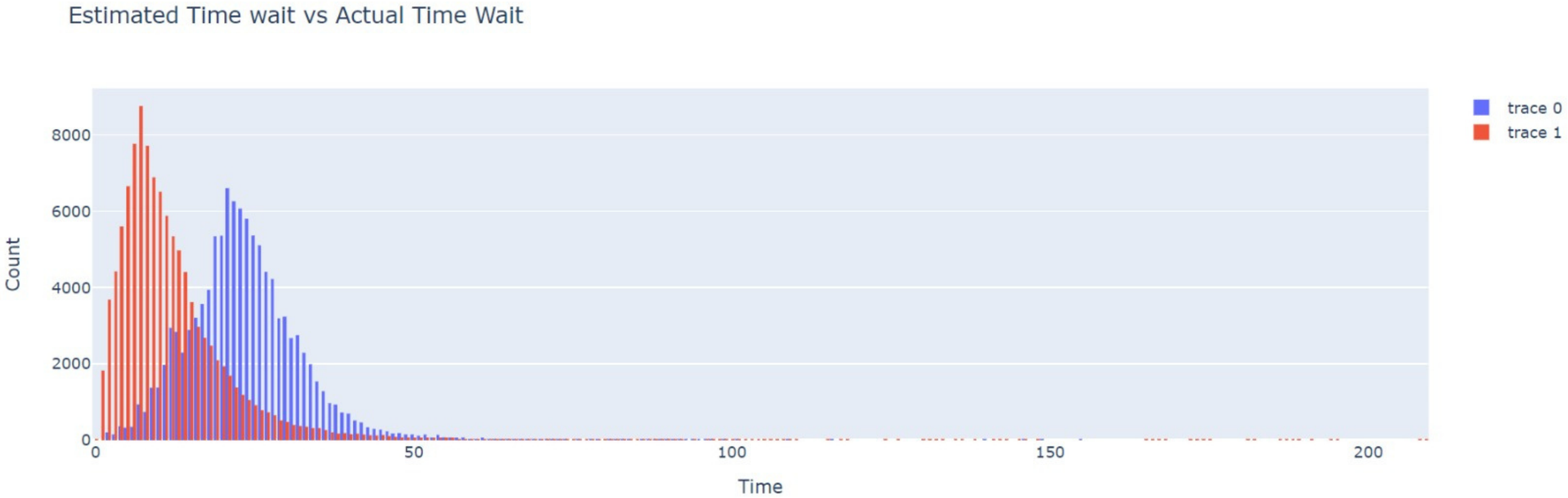
```
#the real wait for orders in on average of 12 days
time_shipment['actual_wait'].mean()
```

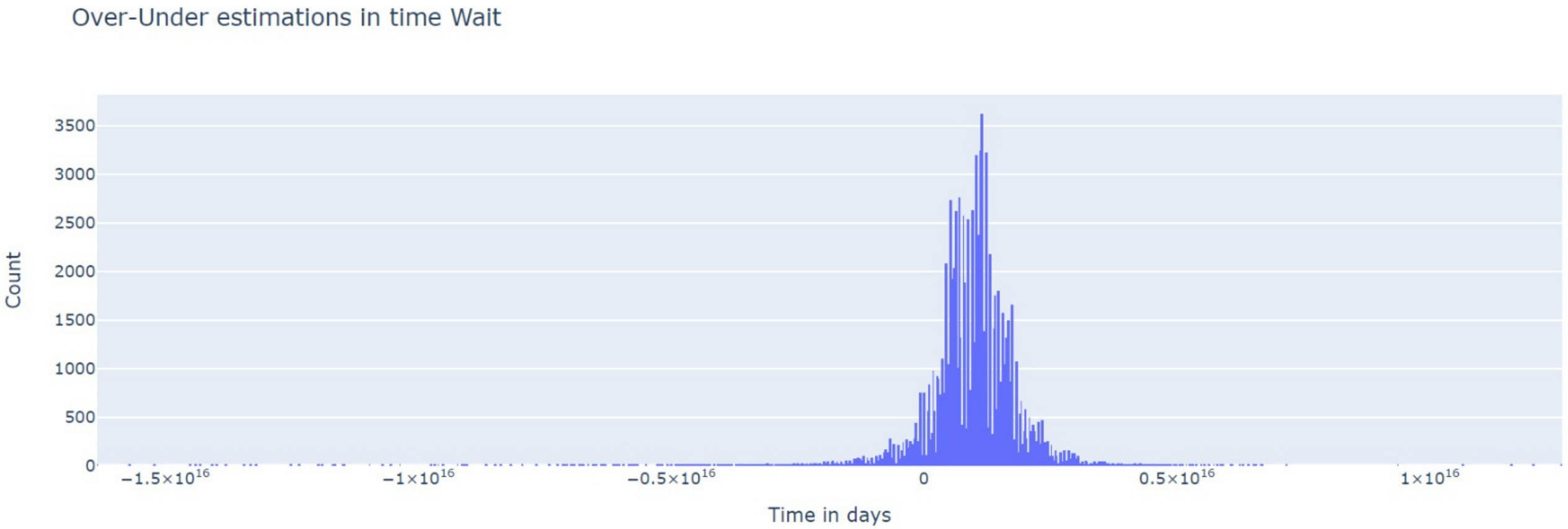
```
Timedelta('12 days 11:19:58.466215357')
```

```
#the communicated time for delivery is of 23 days
time_shipment['est_wait'].mean()
```

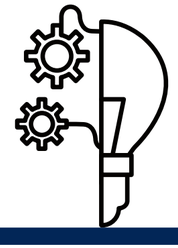
```
Timedelta('23 days 19:19:02.149429250')
```

Visually...


Estimated Time wait vs Actual Time Wait
Count
8000
6000
4000
2000
0
0
50
100
150
200
Time
trace 0
trace 1


Over-Under estimations in time Wait
Count
3500
3000
2500
2000
1500
1000
500
0
−1.5×10^16
−1×10^16
−0.5×10^16
0
0.5×10^16
1×10^16
Time in days

Data Preparation



Creation of a new dataset with just the variables needed for the model

Outcome variable:
actual wait

Independent variables:

- number of items,
- number of products,
- number of sellers,
- maximum distance
- product weight
- op_carrier

number_items	number_products	number_seller	max_distance	op_carrier	product_weight_gr	mean_distance	actual_wait
1	1	1	9144.266345	6.0	650.0	9144.266345	7.0
1	1	1	10513.888384	8.0	30000.0	10513.888384	16.0
1	1	1	2136.253098	1.0	3050.0	2136.253098	7.0
1	1	1	5056.786406	2.0	200.0	5056.786406	6.0
1	1	1	11329.586369	11.0	3750.0	11329.586369	25.0
...	...	...	...	...	...	...	...
1	1	1	6180.268603	10.0	1050.0	6180.268603	16.0
1	1	1	651.543193	1.0	10150.0	651.543193	17.0
1	1	1	11437.146886	2.0	8950.0	11437.146886	9.0
1	1	1	11437.146886	2.0	967.0	11437.146886	4.0
1	1	1	9068.190113	1.0	600.0	9068.190113	5.0

Just one variable between maximum and mean distance to choose)

Modeling & Results

Three models were fitted for the shipment time analysis:

1

K-nearest neighbors

- MAE = 6.4 (missing of 8 days)
- RMSE = 10.39%

2

eXtreeme Gradient Boosting

- MAE = 5.60 (missing of 6 days)
- RMSE = 10%

3

Random Forest

- MAE = 5.5 (missing of 5.5 days)
- RMSE = 9%

The model misses of 5 days compared to the 11 days initally estimated by the company. In this way, there is an increment in the delivery time prediction of a little more than 50%

LUISS

NTT DATA

Link to our animated presentation:

https://www.canva.com/design/DAFAbbyc9OY/LFaojfP_WHBMoP_O2PLYrw/view?utm_content=DAFAbbyc9OY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

LUISS

Thank you for your attention!

NTT DATA