CUSTOMER SEGMENTATION BASED ON E-COMMERCE TRANSACTIONS

MARCO SORRENTI SIMONE BACCILE LORENZO SIMONE

DATA UNDERSTANDING

Dataset

customer_supermarket.csv, which contains information about the historical sales of a supermarket company which have been recorded for approximately one year (from 1-12-10 to 9-12-11). This dataset contains 471910 observations of 9 variables.

Column	Description	Category	Type
CartID ← ~~*BasketID*~~	ID of transaction with unique customer and date time	Categorical	Object
CartDate ← ~~*BasketDate*~~	Purchase date time in timestamp format	Categorical	Object
UnitPrice ← ~~*Sale*~~	Price of a single item, unique by *ProductID*	Numerical	Float64
CustomerID	ID representing each different customer	Categorical	Object
CustomerCountry	Birth country of the purchasing customer	Categorical	Object
ProductID ← ~~*ProdID*~~	ID representing each different product	Categorical	Object
ProductDescription ← ~~*ProdDescr*~~	Description of the associated *ProductID*	Categorical	Object
Quantity ← ~~*Qta*~~	Number of purchased items for each transaction	Numerical	Int64

DATA CLEANING

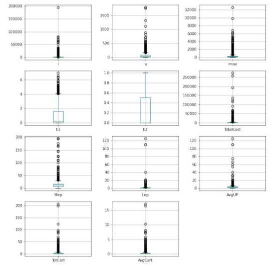
Qta
we drop 9758 rows with Qta less or equal than 0 and greater or equal than 3500, by using the inter-quartile distance and the standard deviation

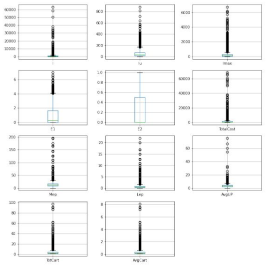
Sale
we drop 826 rows with Sale less or equal than 0 and greater or equal than 200, by using the inter-quartile distance and the standard deviation

CustomerID
we update 1268 rows with a null CustomerID, by creating a custom CustomerID by concatenating the letter “G” with relative BasketID

ProductID
we drop 1693 useless rows that did not represent real products. We drop the following Product description: s, B, M, m, C2, CRUCK, AMAZONFEE, POST, BANK CHARGES

Results
we reduce the size of the dataset from 471910 rows to 459612 rows without missing values.





FEATURE ENGINEERING

Features
useful features extracted from dataset:

- *TotalCost*: total cost of all basket purchased by a customer
- *Mep*: most expensive product bought by a customer
- *Lep*: less expensive product bought by a customer
- *AvgUP*: average UnitPrice of all products bought by a customer
- *TotCart*: total number of basket purchased by each customer
- *AvgCart*: average monthly basket bought by a customer
- *E1*: shannon entropy of the number of product of each basket related with the total number of product purchased by each customer
- *E2*: Shannon entropy of holiday baskets of each customer

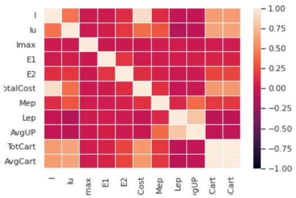
CLUSTERING ANALYSIS

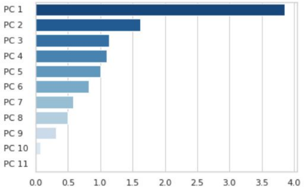
Standardization
prevent different features with
larger scales value from being dominant

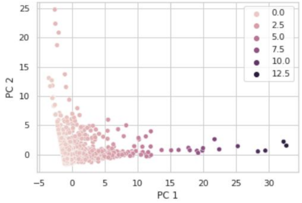
$$x_i \simeq \frac{(x_i - \mu)}{\sigma}$$

Dimensionality reduction
we performed PCA analysis by using
SVD of the matrix A

$$A = USV^T \qquad C = \frac{VSU^TUSV^T}{(n-1)} = V\frac{S^2}{(n-1)}V^T$$



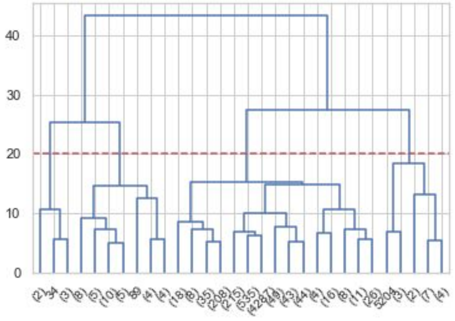


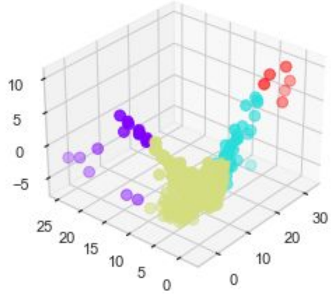


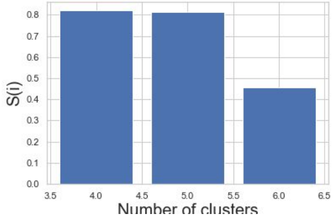
HIERARCHICAL CLUSTERING

Agglomerative Clustering
we used the euclidean distance as the measure of distance between points and complete linkage to calculate the proximity of clusters.

Dimensionality reduction
we performed PCA analysis extracting the top three principal components







K-MEANS CLUSTERING

SSE Estimation

For each of the runs we analyzed the Sum of Squared Error

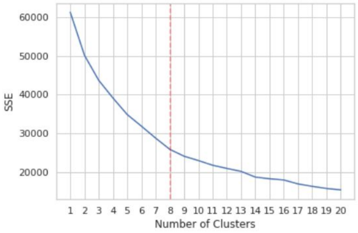
$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dist(m_i, x)^2$$

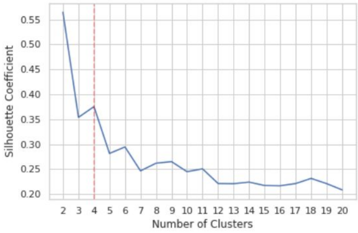
Silhouette Coefficient

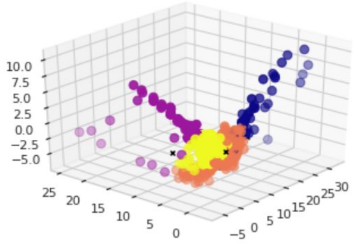
How similar a point is to its own cluster compared to other clusters

Hierarchical Clustering Analysis

We collected the results from the metrics and merged it with the informations from hierarchical clustering analysis



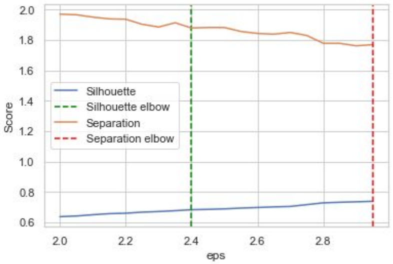


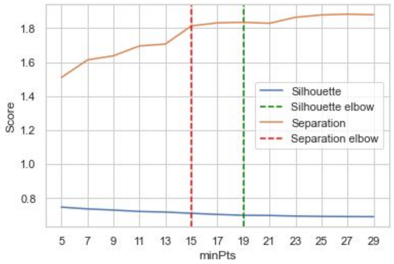


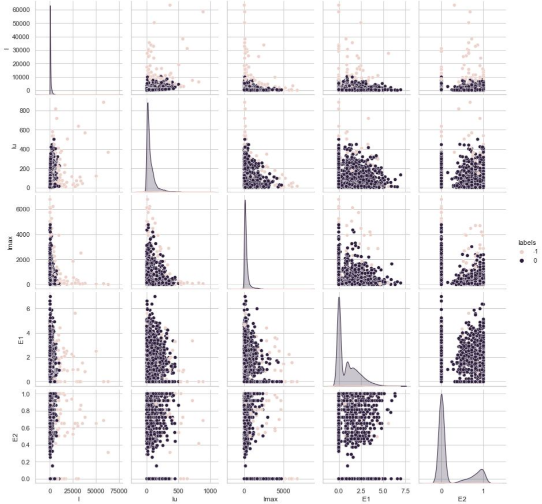
DBSCAN

Parameters Analysis
We analyzed the two parameters of the DBSCAN algorithm:
eps: the maximum distance between two customer to consider them similar.
minPts: the number of customers in a neighborhood for a point to be considered as a core point.

The evaluation of goodness of parameters was made considering two scores: the silhouette score and the separation score.







ALTERNATIVE CLUSTERING TECHNIQUES

Pyclustering library

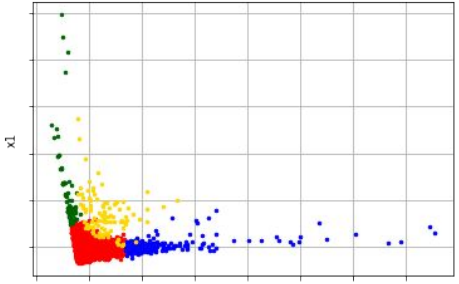
python, C++ data mining library that provides several algorithms for clustering analysis, oscillatory networks and neural networks.

G-Means

uses a statistical test to decide whether to split K-Means center into two centers in order to determine an appropriate amount of cluster.

X-Means

is an extension of the K-Means algorithm which tries to automatically determine the number of clusters based on BIC scores. It starts with only one cluster and makes local decisions, after each run of K-Means, about which subset of the current centroids should split themselves in order to better fit the data.



CUSTOMER TYPE

CustomerType definition

We defined the CustomerType attribute as the average cost per cart for each customer. Then we divided the customers into the three different categories using the inter-quartile distance between the previous average cost.

New Features

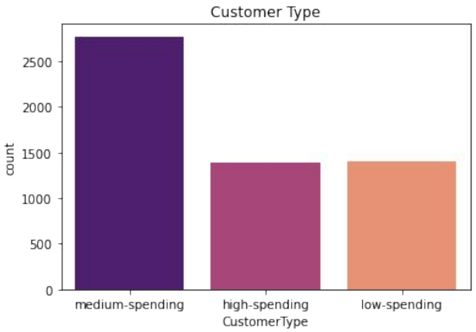
We added three new features for the predictive analysis task that contain the number of products bought by a customer based on the type of product (cheap, average, expensive)

Evaluation metrics

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$F1score = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall}$$



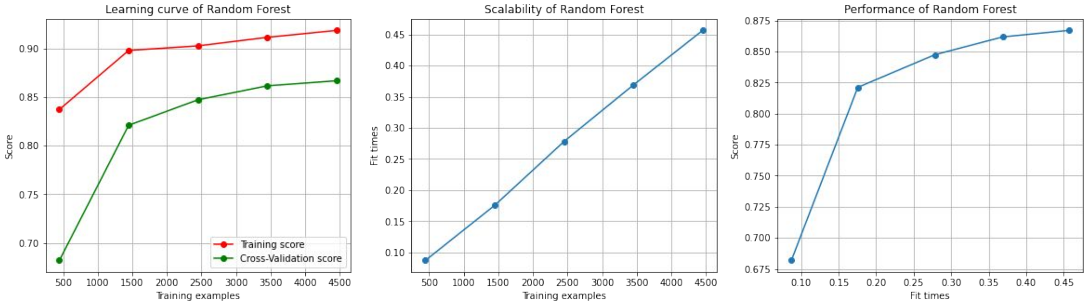
RANDOM FOREST

Parameters tuning

We evaluated the goodness of each parameter using Training score and Cross-validation score. For some parameters we used an exhaustive search, for others a randomized search of the best value.

Model evaluation

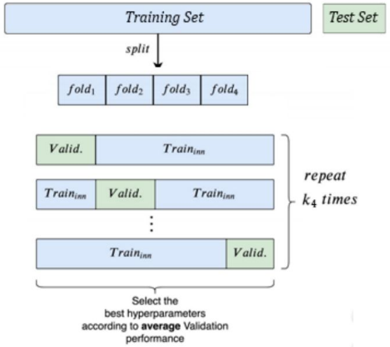
We evaluated not only the accuracy of the model, but also its scalability and its performance, related to an increasing number of data.



MULTI LAYER PERCEPTRON

K-Fold CV

We left out 20% of labeled data (1114 samples).
The remaining 80% (4453 samples) has been furtherly divided in TR and VL undergoing a k-fold cross validation procedure with k = 4



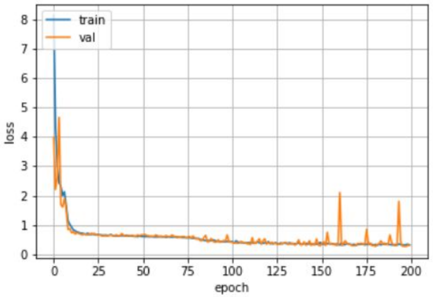
Model selection and evaluation

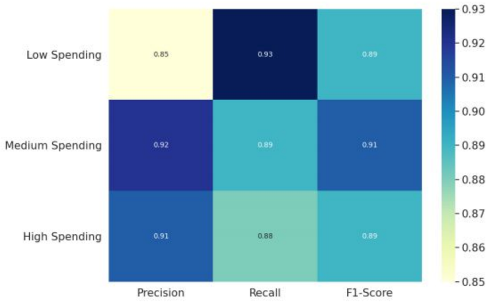
We performed a grid search model selection phase exploring:

- Batch sizes [12, 24, 32]
- Learning rate [10^{-1} , 10^{-2} , 10^{-3}]
- Hidden neurons [50, 100, 150]
- Epochs [100, 200, 300]

MULTI LAYER PERCEPTRON

Models comparison





Model	Test score	Train score	Precision	Recall	F1 score
Multi Layer Perceptron	90%	93%	89%	90%	90%
AdaBoost	87%	98%	89%	85%	87%
Random Forest	87%	93%	87%	85%	86%
Decision Tree	84%	97%	86%	85%	85%
KNeighbors	69%	75%	71%	65%	67%
Gaussian Naive Bayes	46%	47%	53%	50%	43%

GENERALIZING PRODUCTS [POS TAGGING]

PoS Tagging

For each product description we remove color informations, numbers and english stopwords.
We perform PoS tagging removing adjectives and stopwords.

```
P = [
      'LUNCH BAG I LOVE LONDON' ,
      'LUNCH BAG RED RETROSPOT',
      'LUNCH BAG WITH CUTLERY RETROSPOT'
    ]
```

```
PN = [
      'LUNCH BAG LONDON' ,
      'LUNCH BAG RETROSPOT',
      'LUNCH BAG CUTLERY RETROSPOT'
     ]
```

	PRP	Personal Pronoun
	ADJ	Adjective
	IN	Conjunction

GENERALIZING PRODUCTS [CLUSTERING]

Distance Matrix

We built a distance matrix amongst each of the unique product descriptions for running agglomerative clustering

$$C = n + n - 1 + n - 2 + ...1 = n^2 - 1 - 2 ... - n = \frac{2n^2 - n^2 - n}{2} = \frac{n^2 - n}{2}$$

Jaro-Winkler Distance

$$sim_j = \left\{ \begin{array}{ll} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + t\right), & \text{otherwise} \end{array} \right\} \qquad sim_w = sim_j + \frac{l(1 - sim_j)}{10}$$

Cluster [Lunch bag]

FREQUENT ITEMSET

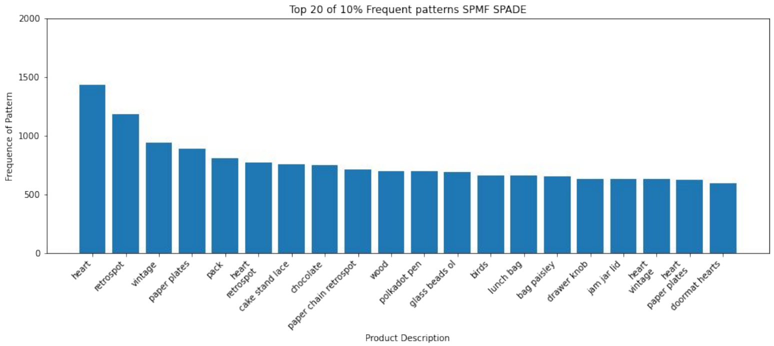
Frequent Itemset mining

To mine the frequent itemset in the dataset, we tested three different packages.

Package	MinSupport	Speed	Filter
GSPPy	Percentage	Slow	✗
PrefixSpan	Number	Fast	✓
SPMF	Percentage	Fast	✓

Frequent Itemset result

One of the results of frequent itemset mining given by the SPMF algorithm with minimum support of 10%.



ASSOCIATION RULES

Association rules mining

We used the Apyori library to find association rules inside the dataset.

Rule metrics

To find the rule we have to set up the minimum support and the minimum confidence for each rule that we want mine.

$$support(X \Rightarrow Y) = \frac{|t \in T; X \Rightarrow Y \subseteq t|}{|T|}$$

$$confidence(X \Rightarrow Y) = \frac{support(X \cup Y)}{support(X)}$$

Rule	Support	Confidence
retrospot ⇒ *heart*	21%	58%
vintage ⇒ *heart*	17%	64%
paperplates ⇒ *heart*	17%	64%
paperplates ⇒ *retrospot*	15%	58%
chocolate ⇒ *heart*	15%	72%
vintage ⇒ *retrospot*	15%	56%
pack ⇒ *heart*	14%	64%
glassbeadsol ⇒ *heart*	14%	70%

TGSP

Minimum gap
Number of days intervening amongst patterns
We are not interested into breaks between shopping sessions
[1 day]

Maximum gap
Pruning association rules too far from each other in
[3 to 7 days]

Minimum interval
Length of a time frame
Usually provides interesting results on the customer shopping behaviour
[3 to 6 months]

TGSP

TGSP Parameters [Min gap = 1 day Max Gap = 1 week - Min Interval = 3 months]

The cluster heart refers to all heart-shaped products, while vintage and retro contain old-style products. There is a common habit exhibited by customers in buying vintage or retro items consequently to heart-shaped ones. Another strong habit is the one of persistently buying vintage items.

Low spending customers patterns

Association Rule	GSP Support	TGSP Support
heart ⇒ *vintage*	22%	3%
heart ⇒ *retro*	21%	3%
vintage ⇒ *heart*	21%	3%
vintage ⇒ *vintage*	20%	2%

High spending customers patterns

Association Rule	GSP Support	TGSP Support
vintage ⇒ *vintage*	58%	5%
retro ⇒ *retro*	55%	5 %
retro ⇒ *vintage*	55%	5%
heart ⇒ *vintage*	53%	4 %
retro ⇒ *heart*	52%	5 %
vintage ⇒ *heart*	51%	3%
vintage ⇒ *retro*	49%	3%
heart ⇒ *retro*	47%	4%
chocolate ⇒ *heart*	42%	2%

THANKS FOR WATCHING

WE ARE OPEN FOR QUESTIONS