CentraleSupélec

ML Engineer

Project 4

E-commerce customers clustering

Victor Benard

MISSION STATEMENT

olist

Olist is a Brazilian online e-commerce platform that wishes to carry out a segmentation of its customers. Objective is to create clusters, and subsequently propose a maintenance plan based on clustering temporal stability.

TABLE OF CONTENTS

01

OVERVIEW & TARGETS

OVERVIEW & TARGETS

Olist proposes an anonymized database that contains orders information and history, ordered products, satisfaction, and location of clients from 2016.

Objective:

Use unsupervised methods to regroup customers that have similar profiles.

Key information:

- Data available from October 2016 to October 2018
- 9 Dataframes for a total of 99k customers id
    - Customer, Seller, Geolocation
    - Order data, order payment, order review, order items
    - Product, product translation

OVERVIEW & TARGETS

Datasets overview

GEOLOCATION

geolocation_lat
geolocation_lng
geolocation_zip_code_prefix
geolocation_city
geolocation_state

SELLER

seller_id
seller_zip_code_prefix
seller_city
seller_state

CUSTOMER

customer_id
customer_unique_id
customer_zip_code_prefix
customer_city
customer_state

ORDER DATA

order_id
customer_id
order_status
order_purchase_timestamp
order_approved_at
order_delivered_carrier_date
order_delivered_customer_date
order_estimated_delivery_date

ORDER ITEMS

order_id
order_item_id
product_id
seller_id
shipping_limit_date
price
freight_value

ORDER REVIEW

review_id
order_id
review_score
review_comment_title
review_comment_message
review_creation_date
review_answer_timestamp

ORDER PAYMENT

order_id
payment_sequential
payment_type
payment_installments
payment_value

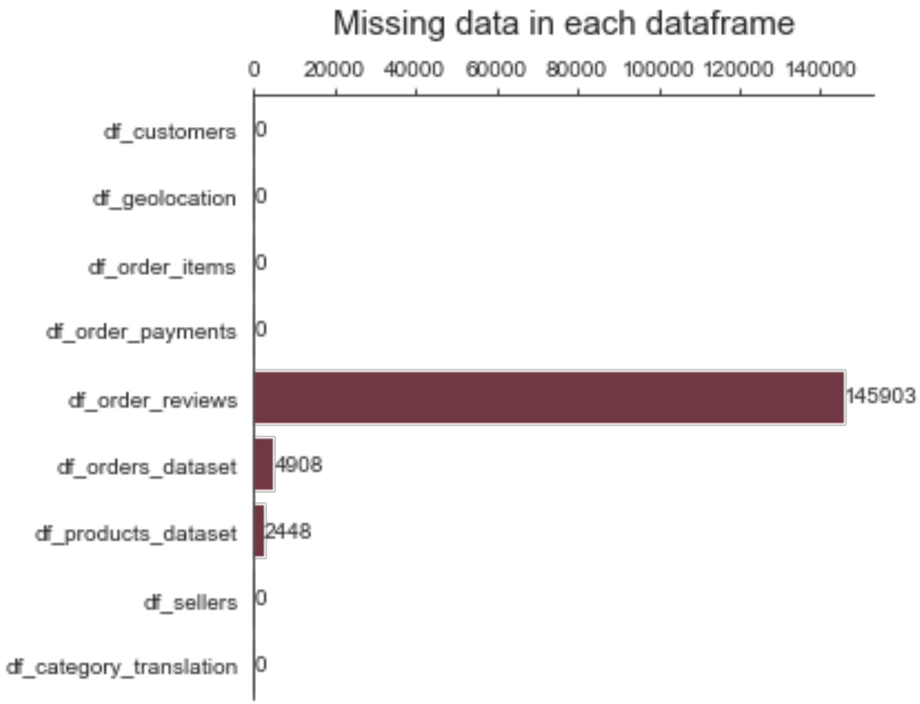
PRODUCT

product_id
product_category_name
product_name_lenght
product_description_lenght
product_photos_qty
product_weight_g
product_length_cm
product_height_cm
product_width_cm

PRODUCT TRANSLATION

product_category_name
product_category_name_english

OVERVIEW & TARGETS



02

DATA MERGING & ANALYSIS

DATA MERGING

Between products, and their translation

- Missing categories added under a translation dictionary
- Inner merge between products, and category translation, on the ‘product_category_name’ feature.
- Dropped portuguese version
- Regrouped categories into 9 large sets
    - hygiene, electronics, furniture, leisure, fashion, groceries, office, diy, misc
- Added missing dimensions and weight by category average
- Calculated volume from dimensions
- Removed:
    - name length, description, photos, length, height, width

PRODUCT
product_id
product_category_name
product_name_lenght
product_description_lenght
product_photos_qty
product_weight_g
product_length_cm
product_height_cm
product_width_cm
product_volume_cm3

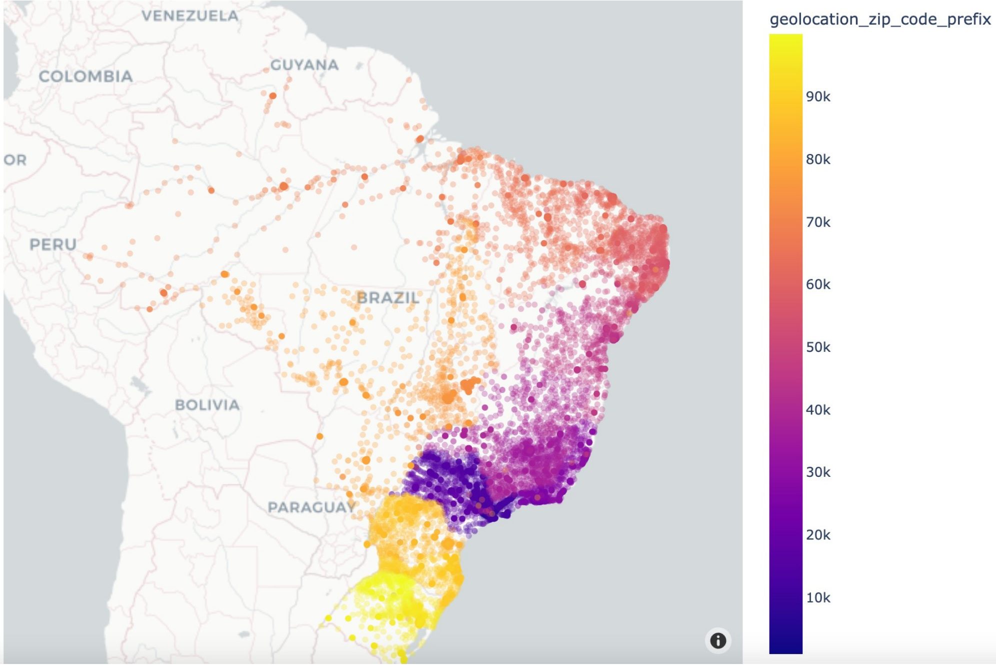
PRODUCT TRANSLATION
product_category_name
product_category_name_english

DATA MERGING

Between sellers, and their location

- Dropped duplicates of ZIP code
- Removed geographical outliers based on Brazil coordinates
  - -35 < latitude < 5
  - -75 < longitude < -35
- Added sellers location by Left join on seller dataset, on zip code
- State average coordinate to fill missing coordinates
- Drop: ‘state’, ‘city’, ‘zip codes’

GEOLOCATION	SELLER
seller_lat	seller_id
seller_long	seller_zip_code_prefix
geolocation_zip_code_prefix	seller_city
geolocation_city	seller_state
geolocation_state	



DATA MERGING

```
agg_dict = {'order_item_id': np.max,
            'seller_id': np.max,
            'price': np.sum,
            'freight_value': np.sum,
            'product_weight_g': np.mean,
            'product_volume': np.mean,
            'product_category_name_english': mode
            }
```

Order items with products and sellers

- Left join on order items from products, based on 'product_id' feature
- As there are multiple items per order, these items are aggregated for each order.
- Left join on order items from sellers, on 'seller_id'

SELLER
seller_id
seller_zip_code_prefix
seller_city
seller_state
seller_lat
seller_long

ORDER ITEMS
order_id
order_item_id
product_id
seller_id
shipping_limit_date
price
freight_value

PRODUCT
product_id
product_category_name_english
product_weight_g
product_volume_cm3

DATA MERGING

```python
agg_dict2 = {
             'perc_credit': np.mean,
             'payment_installments': np.mean,
             'payment_value': np.sum
             }
```

Regrouping all orders data

- Order payments rows with same order id are grouped
- Only orders that are already delivered are considered, the others are discarded
- Left join of order_dataset with order_item, on 'order_id'
- Left join of orders with order_review, on 'order_id'
- Left join of orders with oder_payments, on 'order_id'
- Missing review scores are replaced by the median grade 3

ORDER DATA
order_id
customer_id
order_status
order_purchase_timestamp
order_approved_at
order_delivered_carrier_date
order_delivered_customer_date
order_estimated_delivery_date

ORDER ITEMS
order_id
order_item_id
product_id
seller_id
shipping_limit_date
price
freight_value
seller_city
seller_state
seller_lat
seller_long
product_category_name_english
product_weight_g
product_volume_cm3

ORDER REVIEW
review_id
order_id
review_score
review_comment_title
review_comment_message
review_creation_date
review_answer_timestamp

ORDER PAYMENT
order_id
payment_sequential
perc_credit
payment_installments
payment_value

DATA MERGING

Regrouping all customer data

- Left join of customers with orders, on 'customer_id'
- 2972 customers have no order, these rows are dropped
- Left join of customer with geolocation, on zip code
- Filling missing customer coordinates by using states

GEOLOCATION

geolocation_lat
geolocation_lng
geolocation_zip_code_prefix
geolocation_city
geolocation_state

CUSTOMER

customer_id
customer_unique_id
customer_zip_code_prefix
customer_city
customer_state

ORDER DATA

customer_id
order_purchase_timestamp
order_delivered_customer_date
order_item_id
price
freight_value
seller_city
seller_state
seller_lat
seller_long
product_category_name_english
product_weight_g
product_volume_cm3
review_score
perc_credit
payment_installments
payment_value

DATA CLEANING

Duplicates

- Customer_id, and order_id show 529 duplicates. They are removed.
- ‘customer_id’ feature is removed, as it is unique to order_id, and its name is misleading

Location

- Customer_city contains hundreds of cities, which don’t include significant part of the population. This feature is removed.
- States are grouped in 5 regions. 2 regions have more than 95% orders.

```
state_dict = {
              'North': ['AC', 'AP', 'AM', 'PA', 'RO', 'RR', 'TO'],
              'Northeast': ['AL', 'BA', 'CE', 'MA', 'PB', 'PE', 'PI',
                            'RN', 'SE'],
              'Southeast': ['ES', 'MG', 'RJ', 'SP'],
              'South': ['PR', 'RS', 'SC'],
              'Centerwest': ['DF', 'GO', 'MT', 'MS']
              }
```

DATA CLEANING

Payment

- ‘Price’ feature can be removed as as payment value already gives the total.

Dates

- Purchase and delivery dates are converted with pd.to_datetime
- ’delivery_time’ is created, it’s the difference between date of delivery and date of purchase.

DATA CLEANING

Customers aggregation

- Aggregation is carried out while grouping by customer_unique_id

```
agg_dict = {
            'nb_of_orders': np.max,
            'customer_state': mode,
            'freight_value': np.sum,
            'perc_credit': np.mean,
            'payment_installments': np.mean,
            'payment_value': np.sum,
            'category': mode,
            'product_weight_g': np.mean,
            'product_volume': np.mean,
            'nb_of_items': np.mean,
            'review_score': np.mean,
            'date_purchase': np.max,
            'delivery_time': np.mean,
            }
```

- Recency is calculated from date of purchase
- Frequency is directly given by nb of orders
- Monetary is directly taken from payment_value

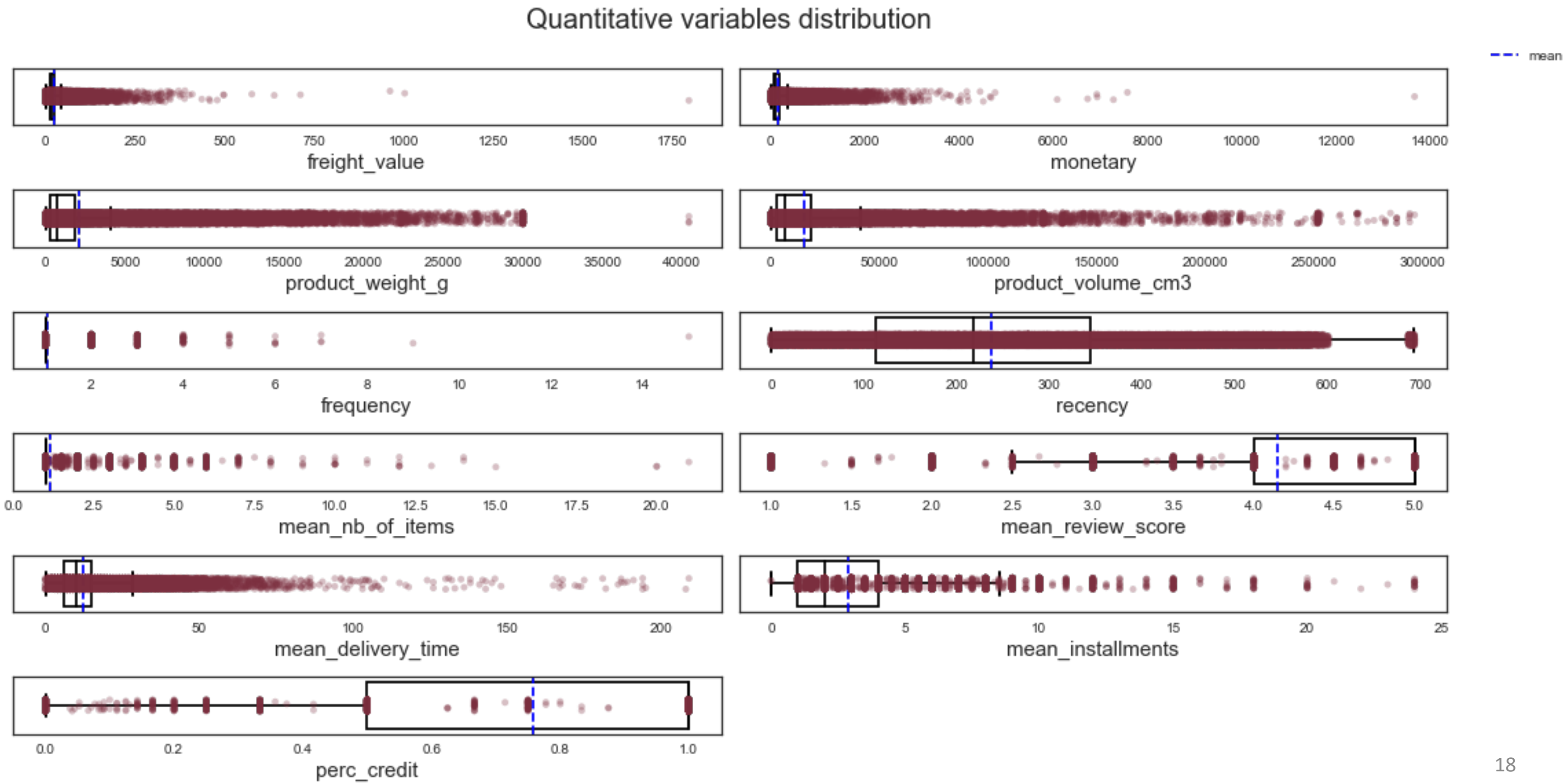
DATA CLEANING

Cleaned features overview

Customer information	ID customer_state
Order information	frequency product_weight_g product_volume_cm3 category mean_nb_of_items
Time	mean_delivery_time date_purchase recency
Payment	perc_credit mean_installments monetary freight_value
Review	mean_review_score

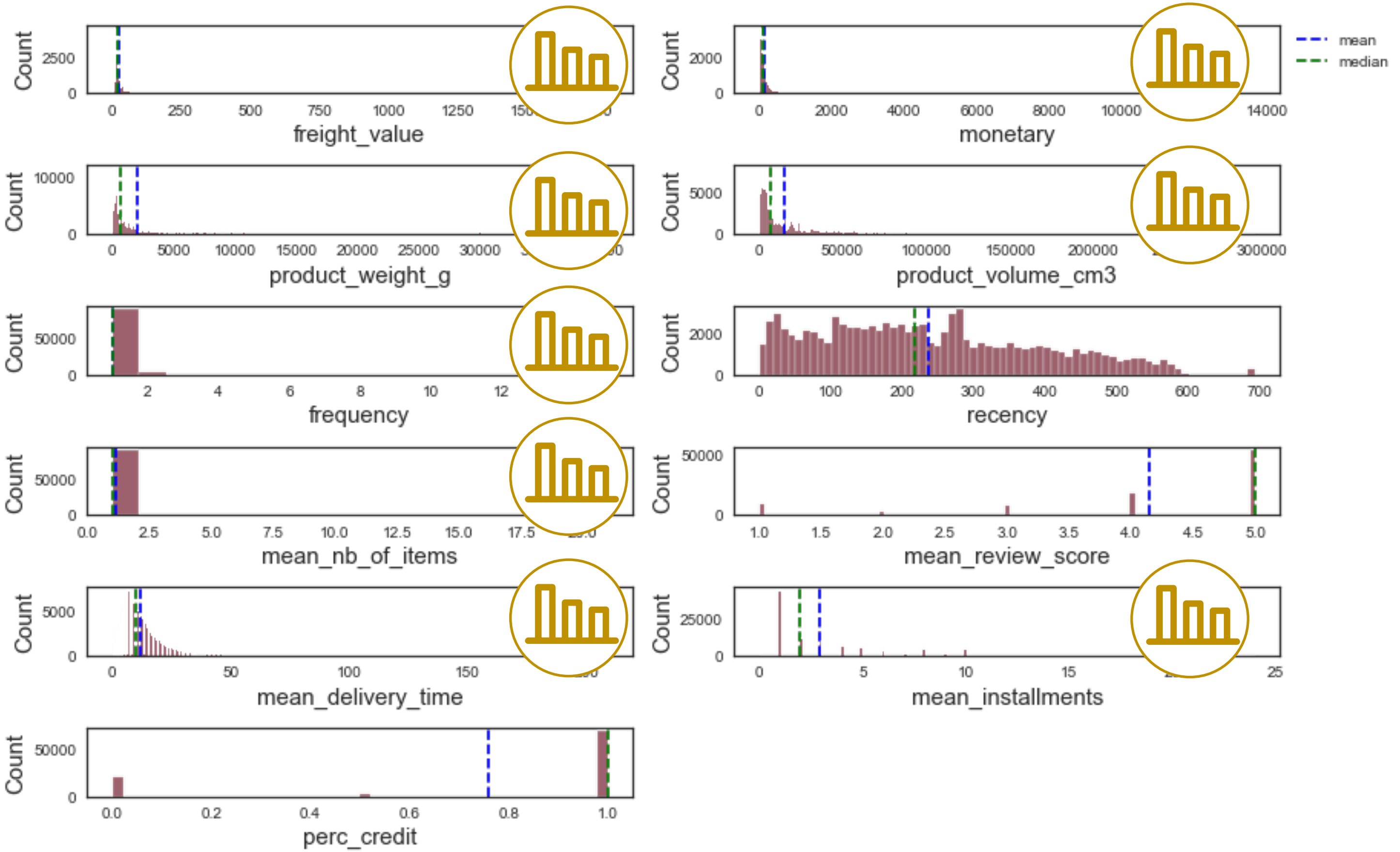
EXPLORATORY ANALYSIS

Numerical features



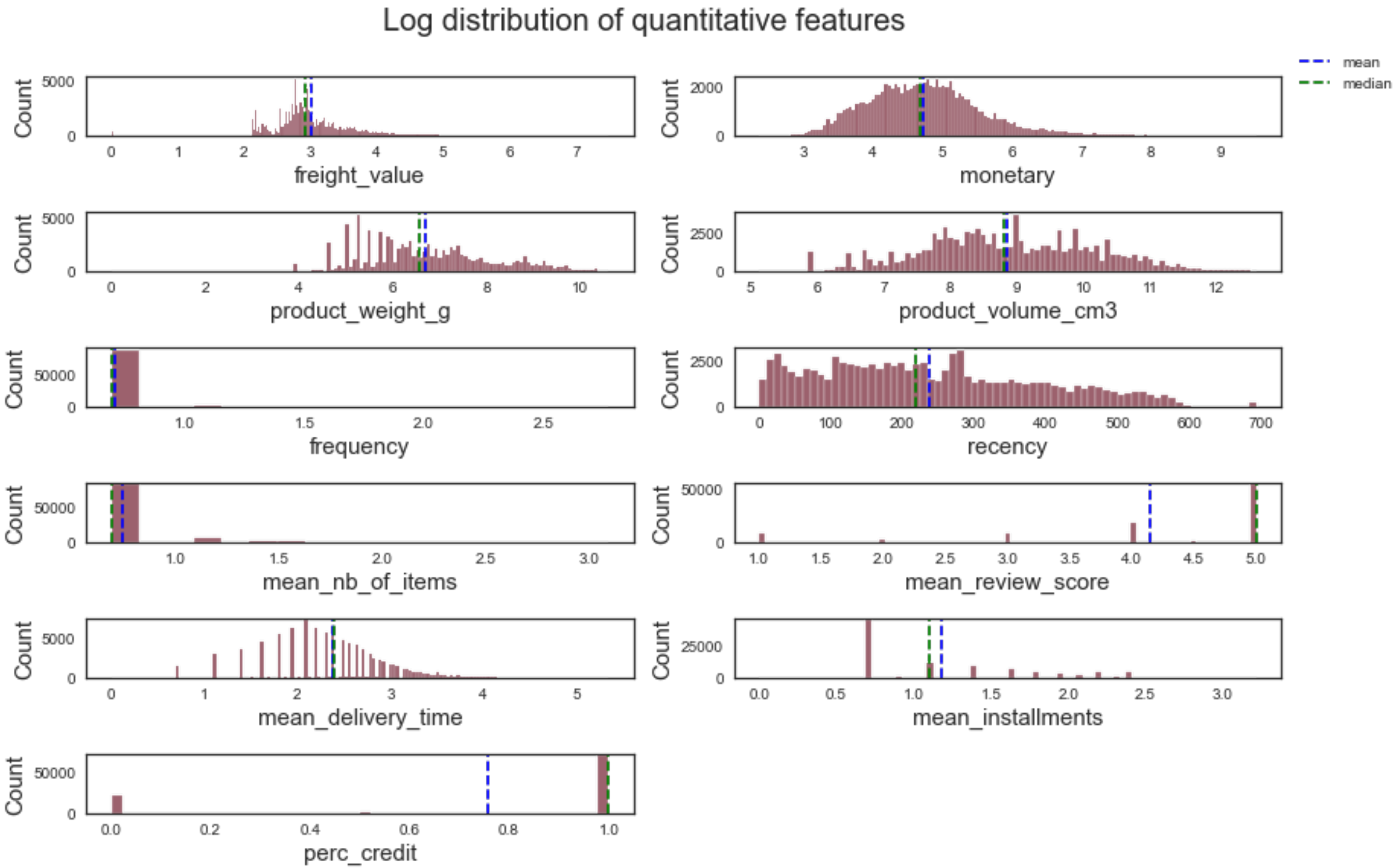
EXPLORATORY ANALYSIS

Numerical variables



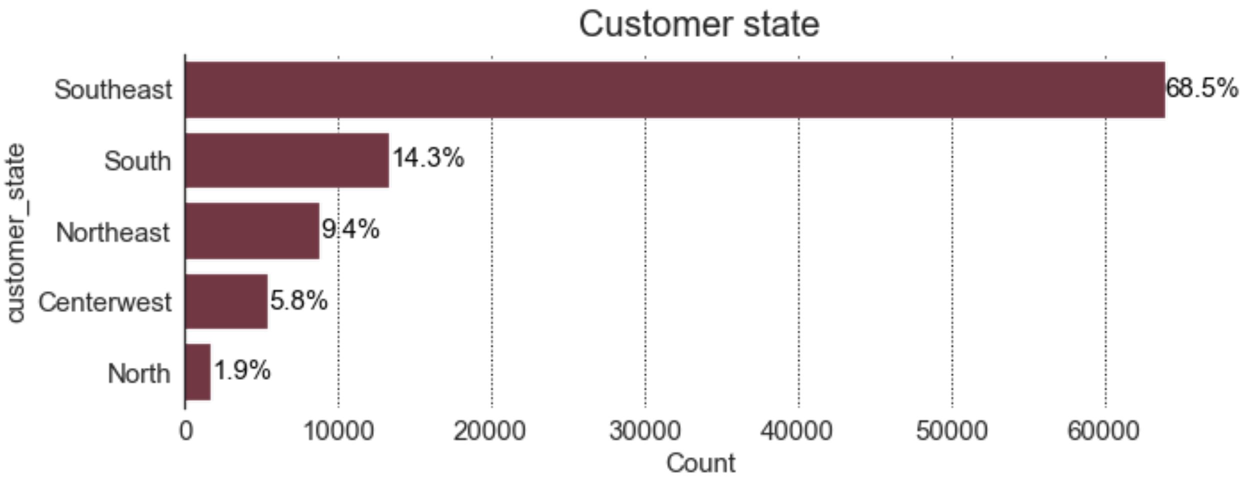
EXPLORATORY ANALYSIS

Numerical variables



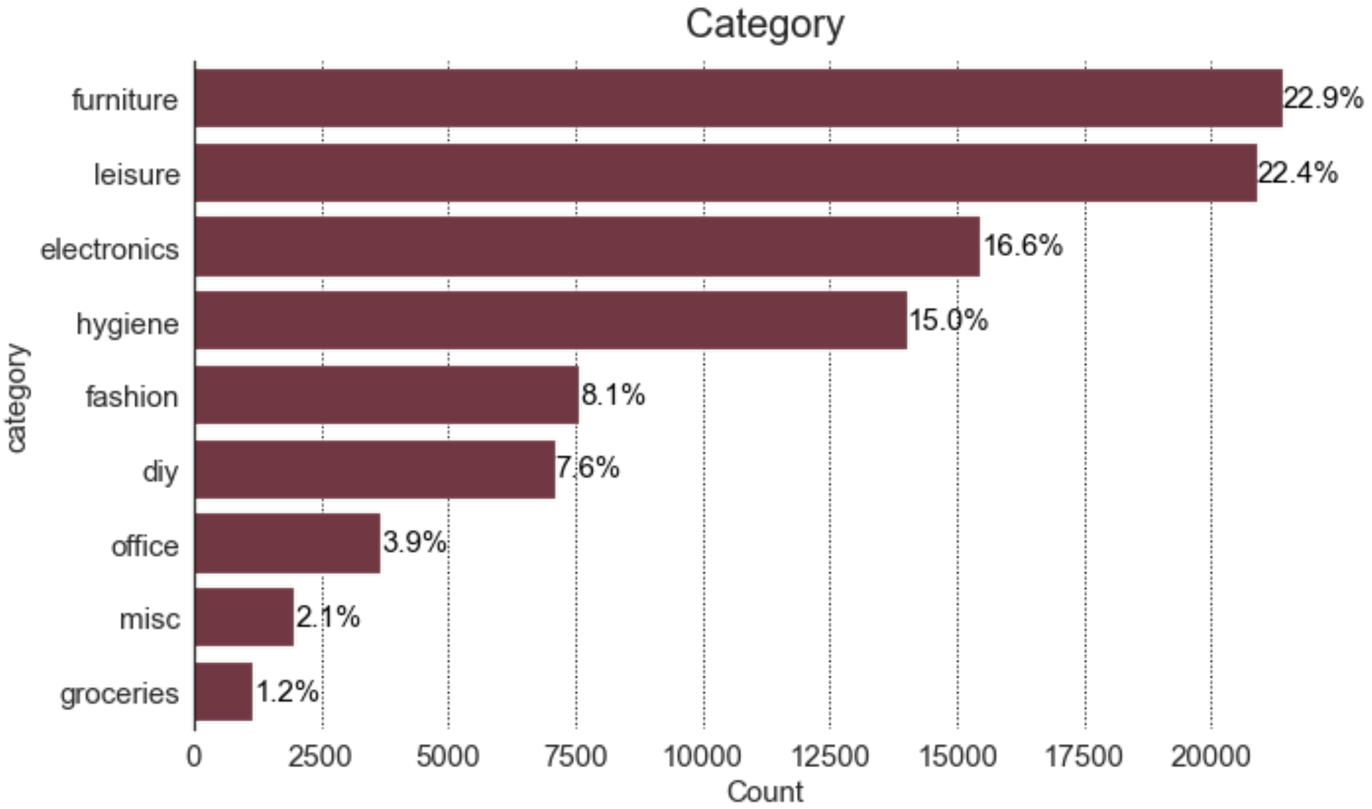
EXPLORATORY ANALYSIS

Categorical features

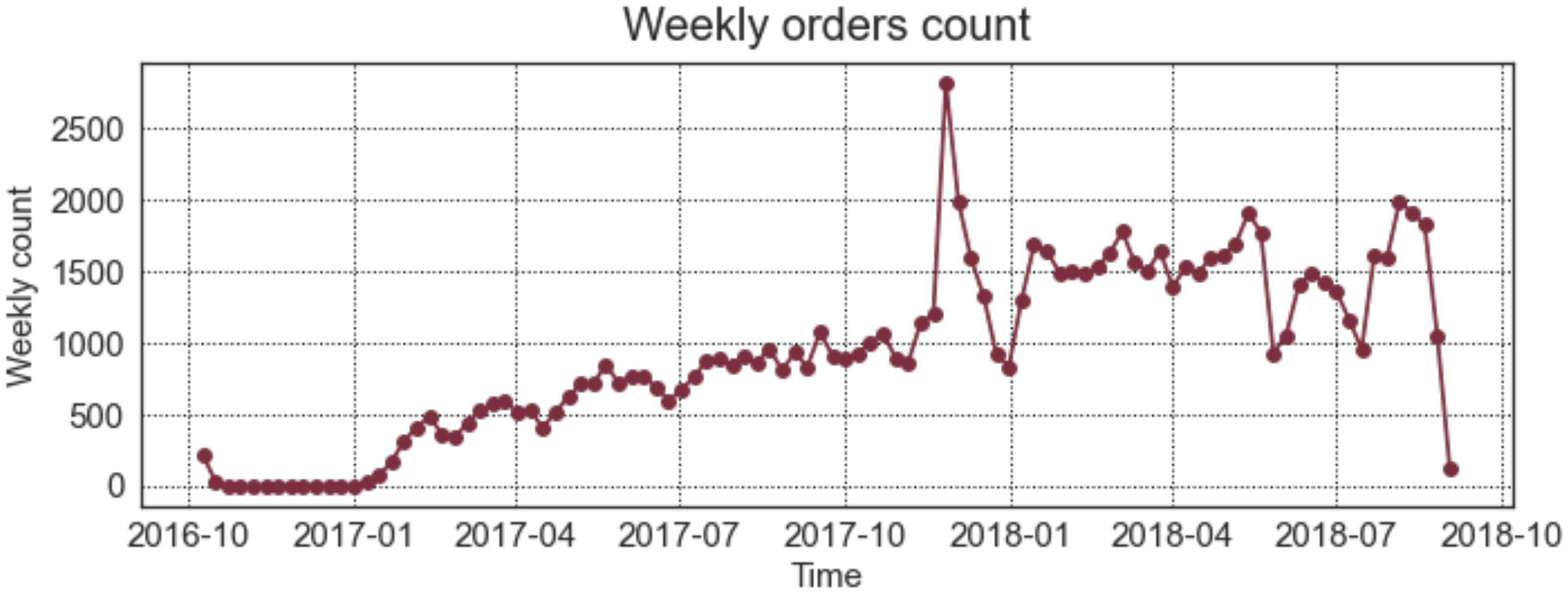


EXPLORATORY ANALYSIS

Categorical features

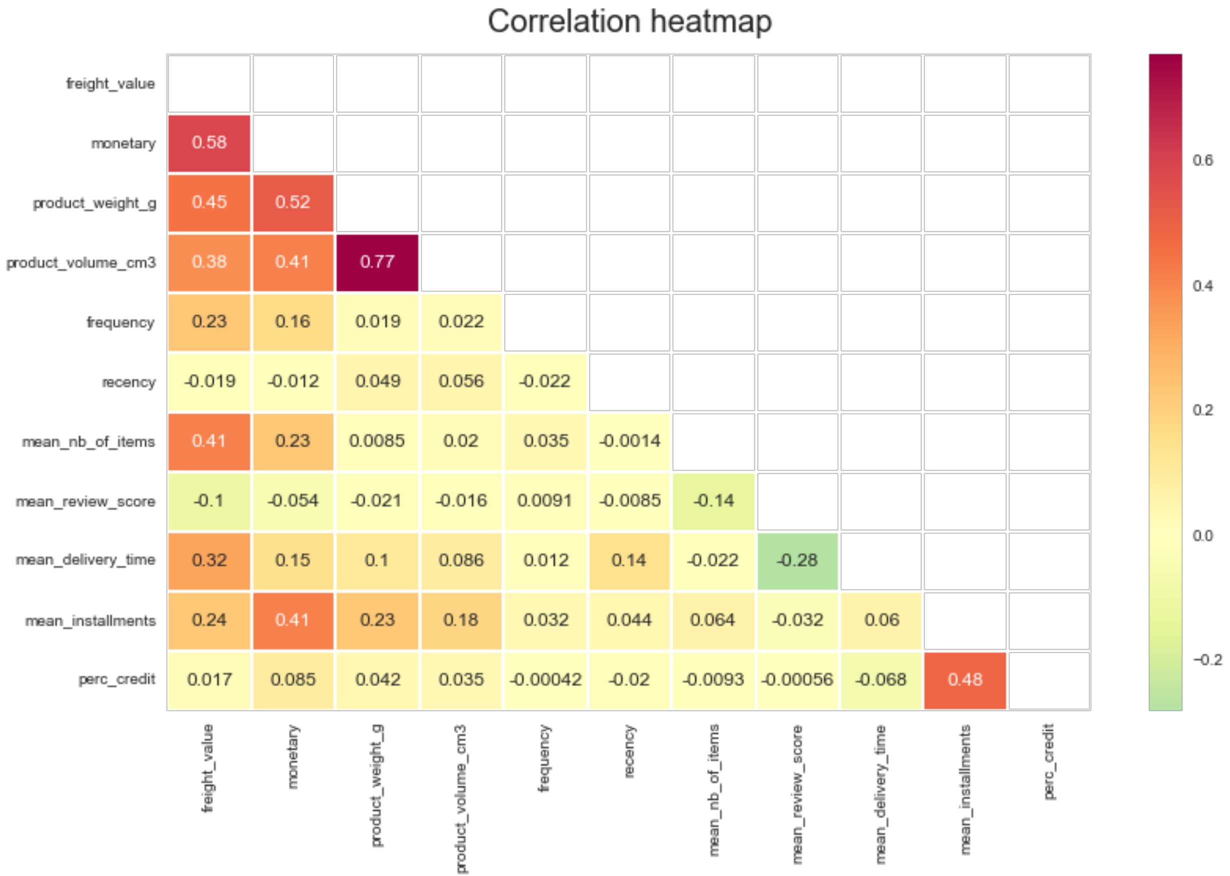


EXPLORATORY ANALYSIS



EXPLORATORY ANALYSIS

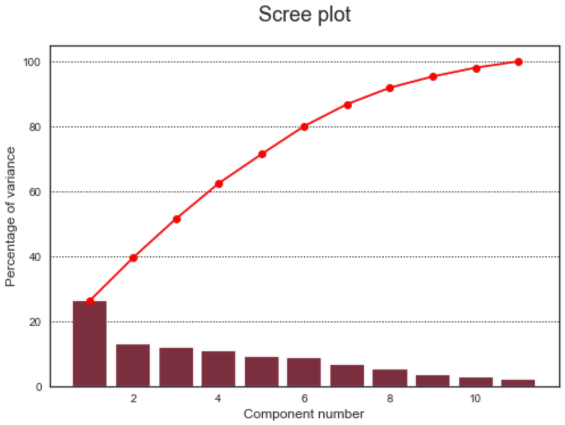
Multivariate analysis

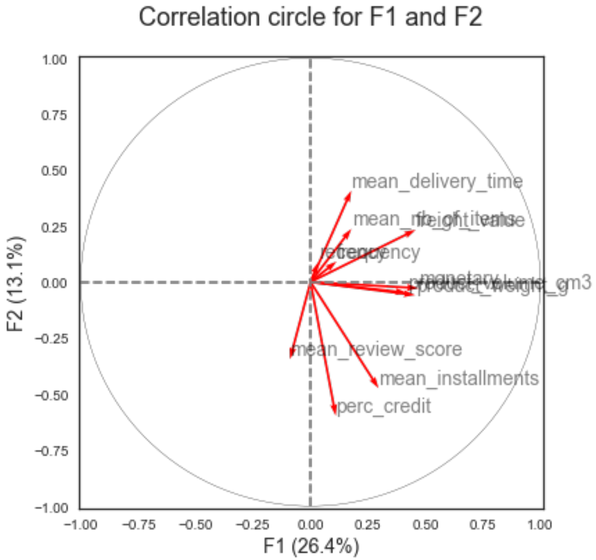


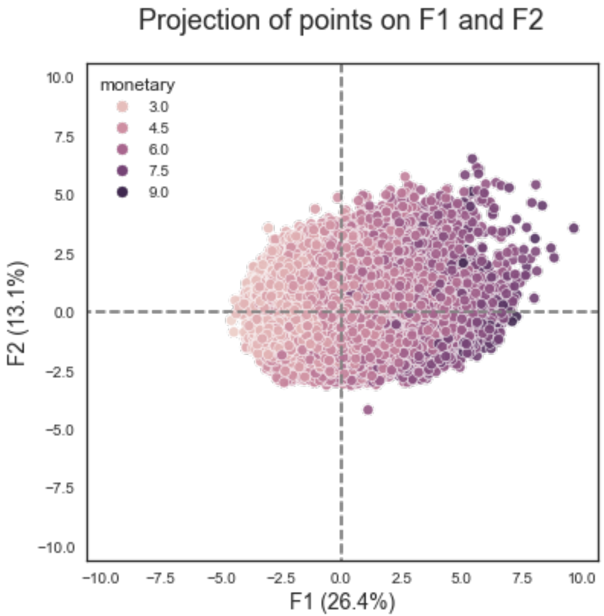
EXPLORATORY ANALYSIS

Multivariate analysis

Principal Component Analysis







EXPLORATORY ANALYSIS

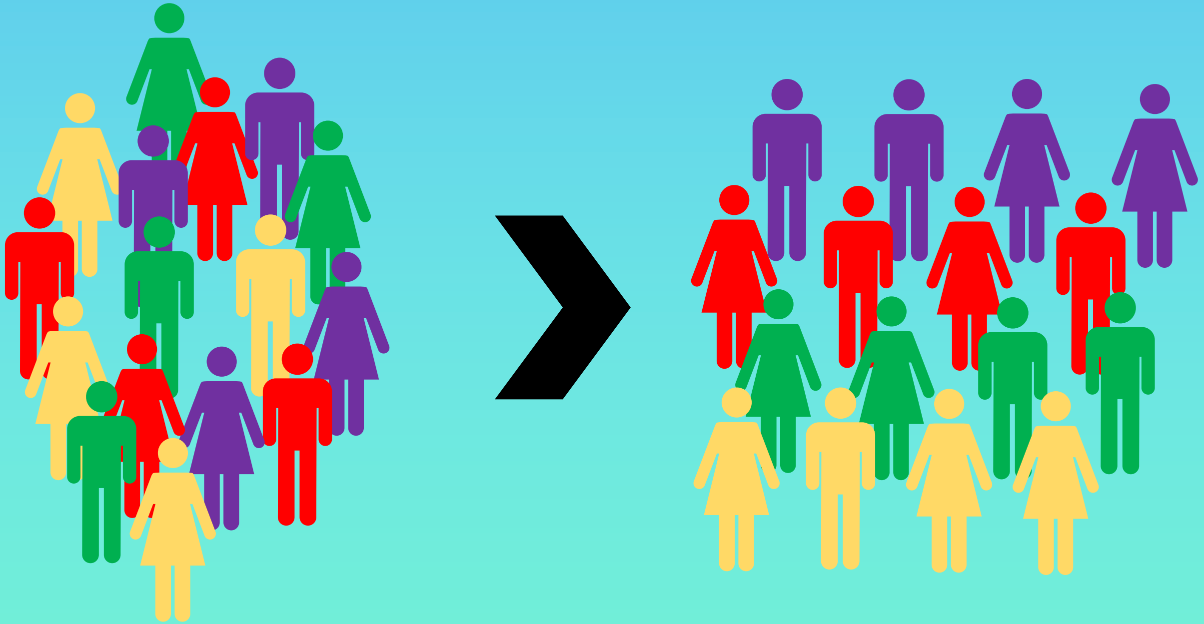
Multivariate analysis

ANOVA

$$\eta^2 = \frac{ESS}{TSS}$$

	Numerical_features	customer_state	category
0	freight_value	0.112305	0.026310
1	monetary	0.013882	0.022516
2	product_weight_g	0.000097	0.183267
3	product_volume_cm3	0.000592	0.235835
4	frequency	0.000124	0.004269
5	recency	0.001761	0.011464
6	mean_nb_of_items	0.000380	0.011825
7	mean_review_score	0.002563	0.002841
8	mean_delivery_time	0.142176	0.006026
9	mean_installments	0.005778	0.023064
10	perc_credit	0.001153	0.002137

03 CLUSTERING



CLUSTERING

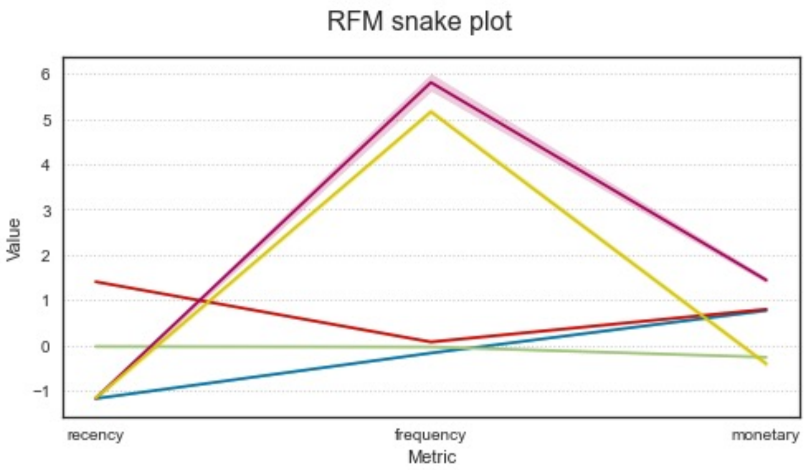
RFM Segmentation

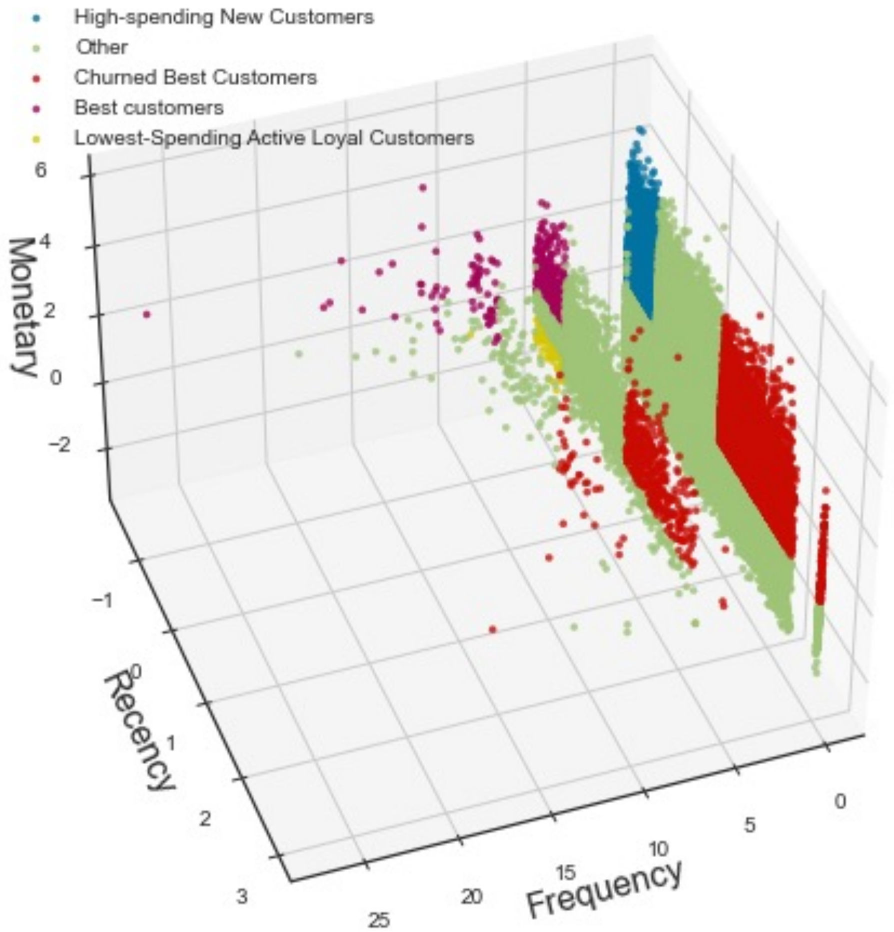
- Recency, Frequency, Monetary features are selected
- There are normalized through StandardScaler
- Segmentation is carried out based on tiers:
  - 4 tiers for recency and monetary
  - Only 2 tiers for frequency due to limited nb of customers that come more than once
- Names are attributed based on scores:
  - 1-1-1 Best customers
  - 1-2-1 and 1-2-2: High-spending new customers
  - 1-1-3 and 1-1-4: Lowest-spending active loyal customers
  - 4-1-1, 4-1-2, 4-2-1 and 4-2-2: Churned best customers

CLUSTERING

RFM Segmentation

- Low number of customers coming more than once limit the relevance of this type of study.





CLUSTERING

Selection of more features as an alternative to RFM

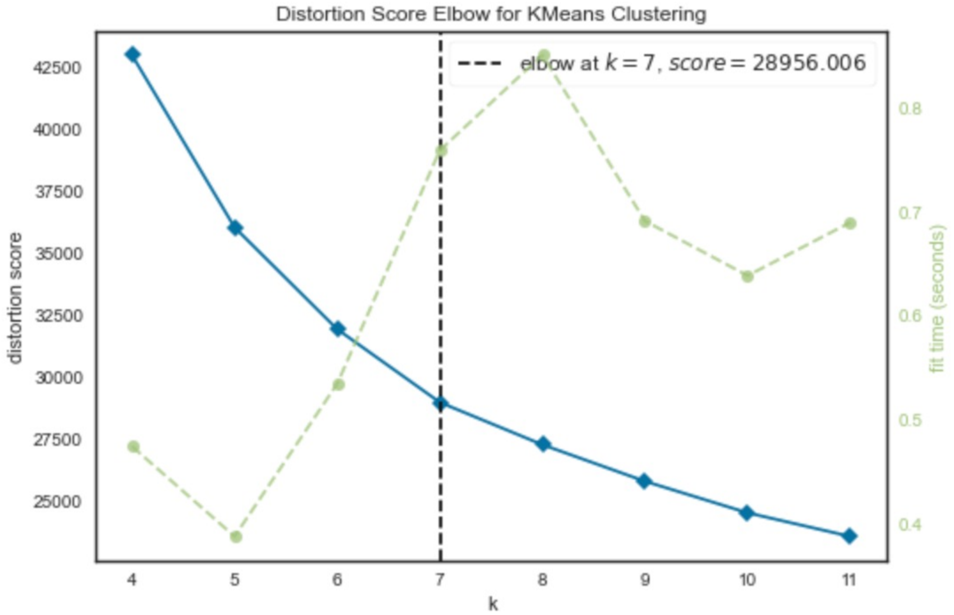
- A random sample of 10000 is considered, to reduce convergency time.
- Follow features are selected:
  - Monetary
  - Frequency
  - Recency
  - Percentage of credit
  - Mean review score
  - Mean number of items
  - Mean product weight
- Features are normalized with StandardScaler

CLUSTERING

Selection of more features as an alternative to RFM

K-means clustering

- Elbow method based on distortion gives a elbow at k = 7 clusters

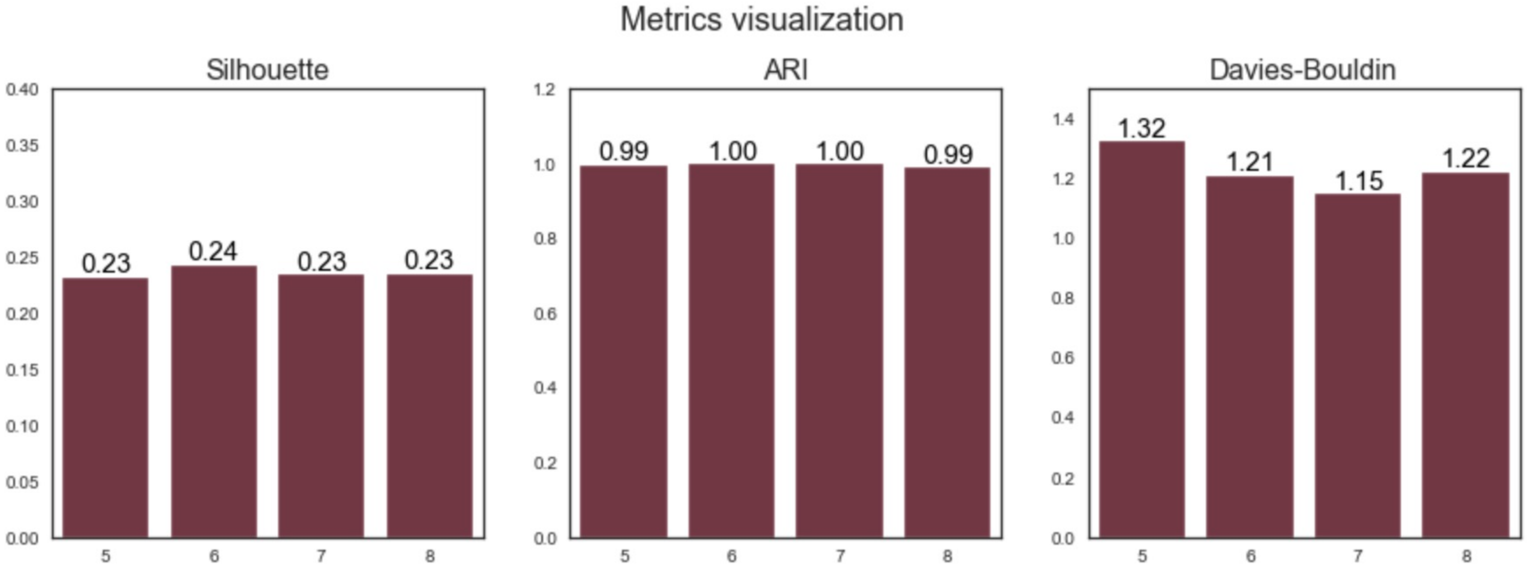


CLUSTERING

Selection of more features as an alternative to RFM

K-means clustering

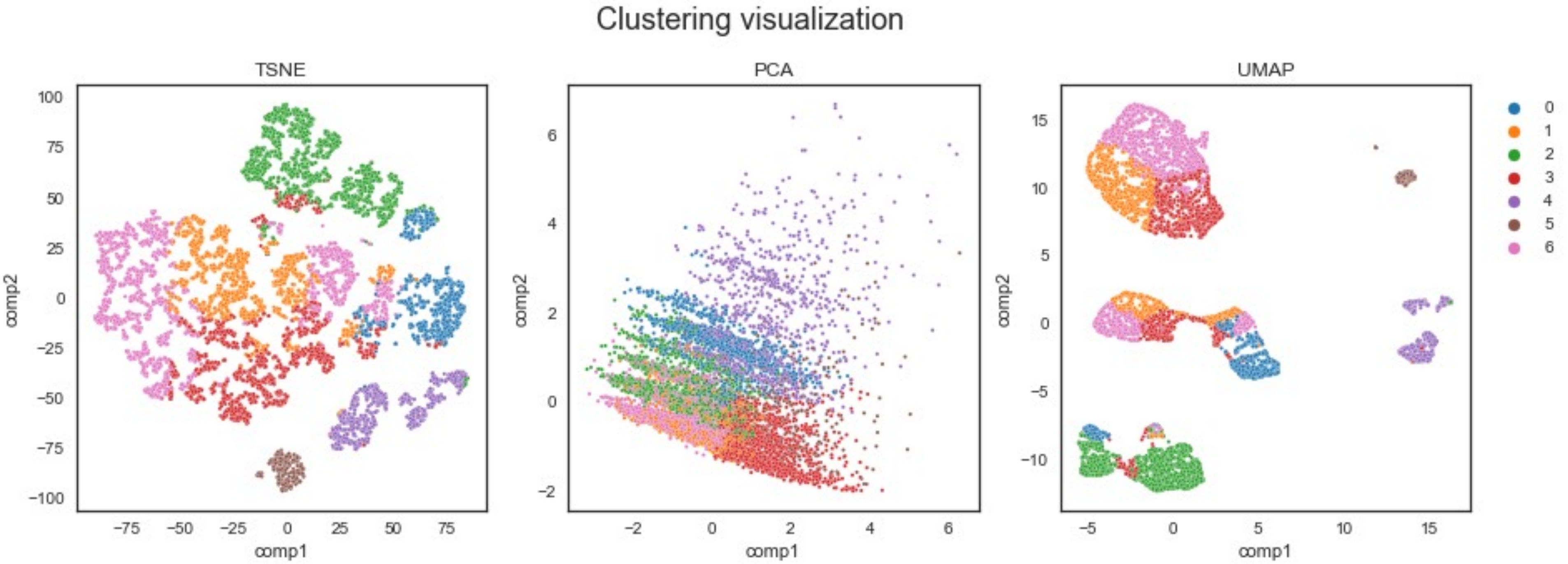
- Metrics also confirm 7 clusters is an interesting choice



CLUSTERING

Selection of more features as an alternative to RFM

K-means clustering

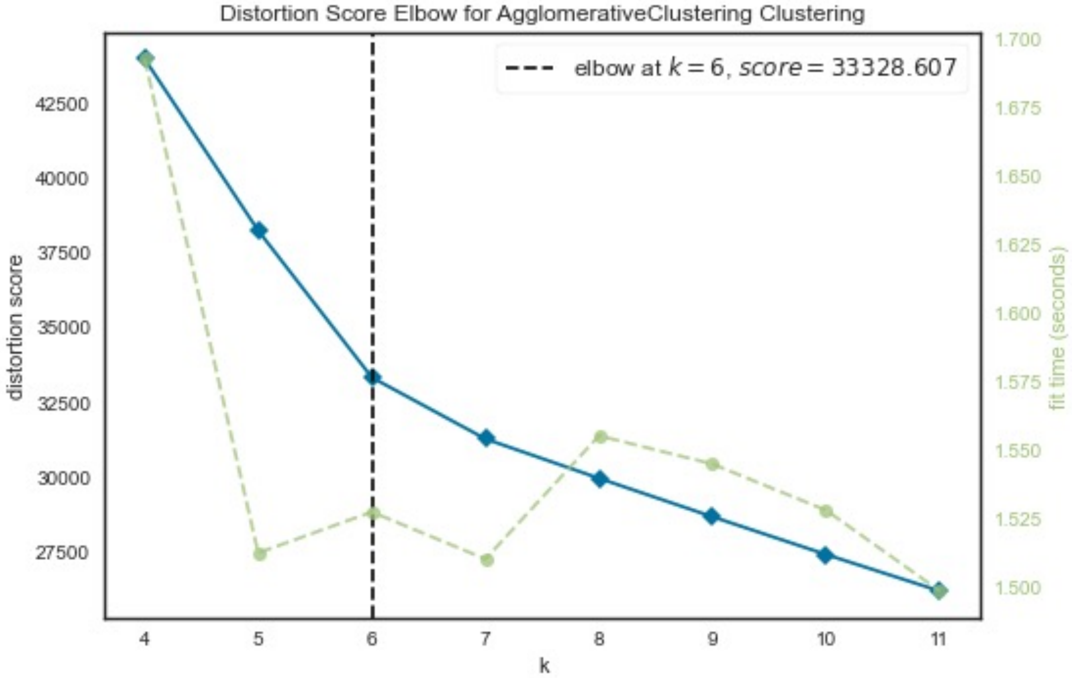


CLUSTERING

Selection of more features as an alternative to RFM

Hierarchical clustering

- Elbow method based on distortion gives a elbow at k = 6 clusters

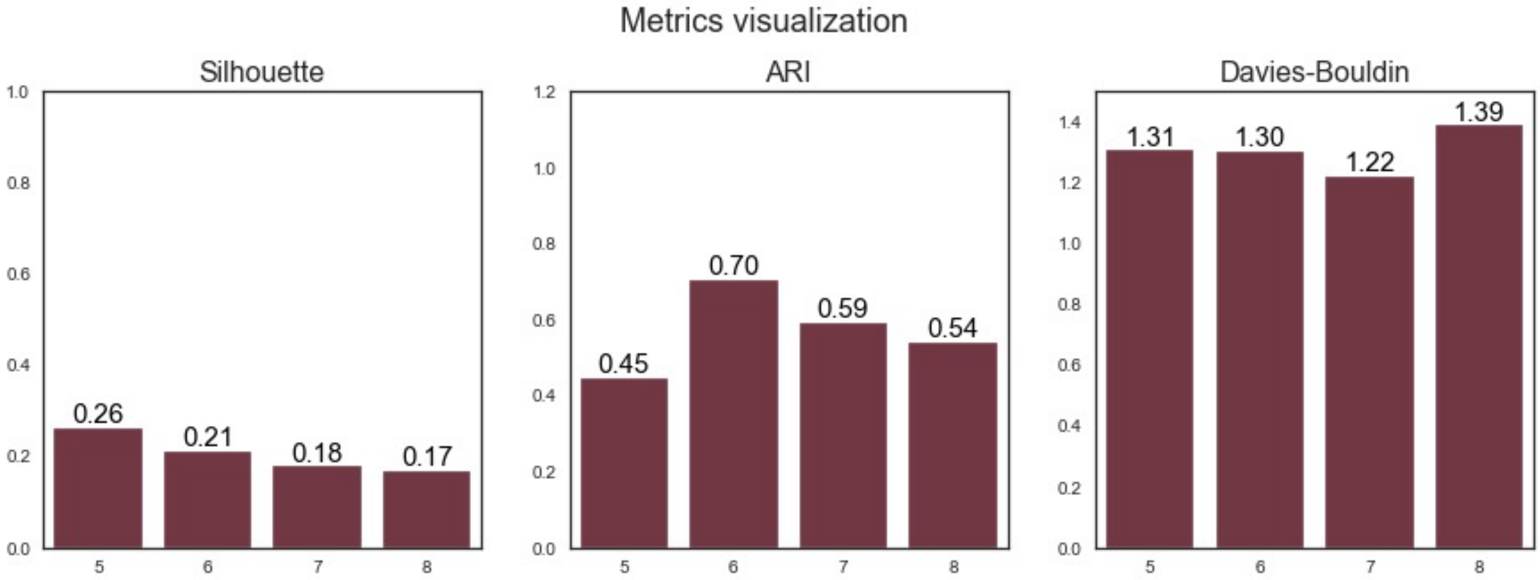


CLUSTERING

Selection of more features as an alternative to RFM

Hierarchical clustering

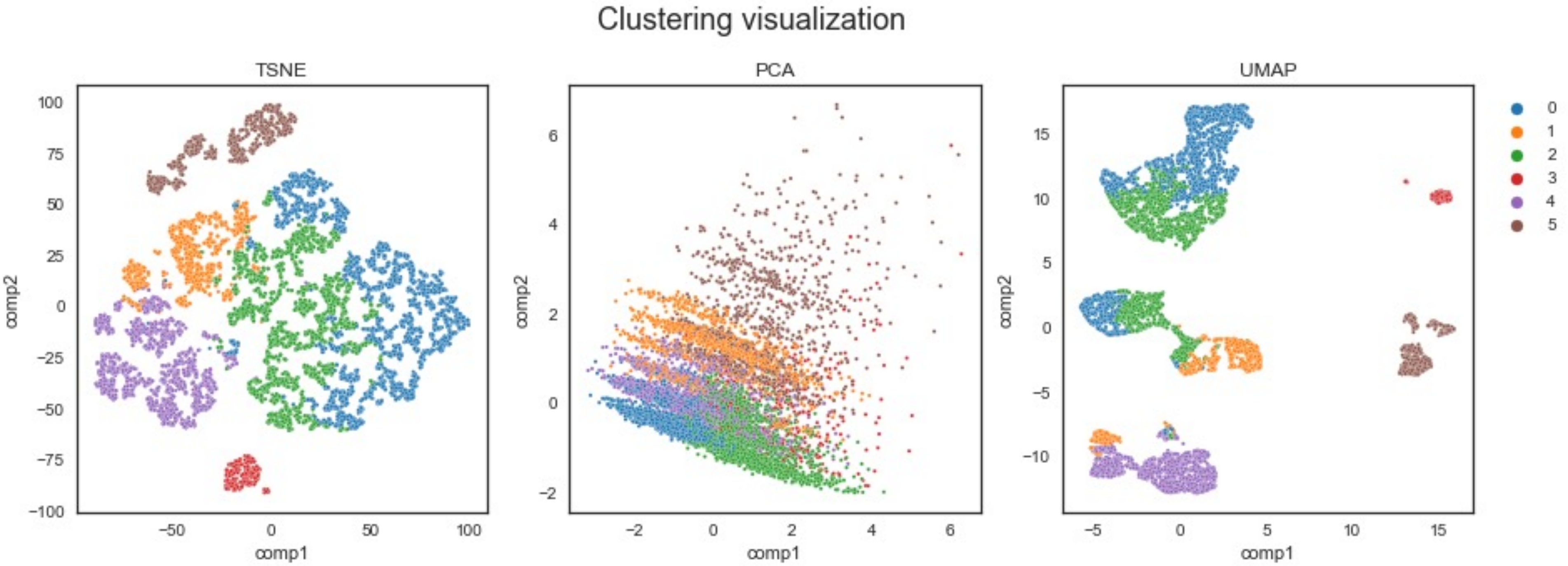
- Metrics also confirm 6 clusters is an interesting choice, mostly for stability



CLUSTERING

Selection of more features as an alternative to RFM

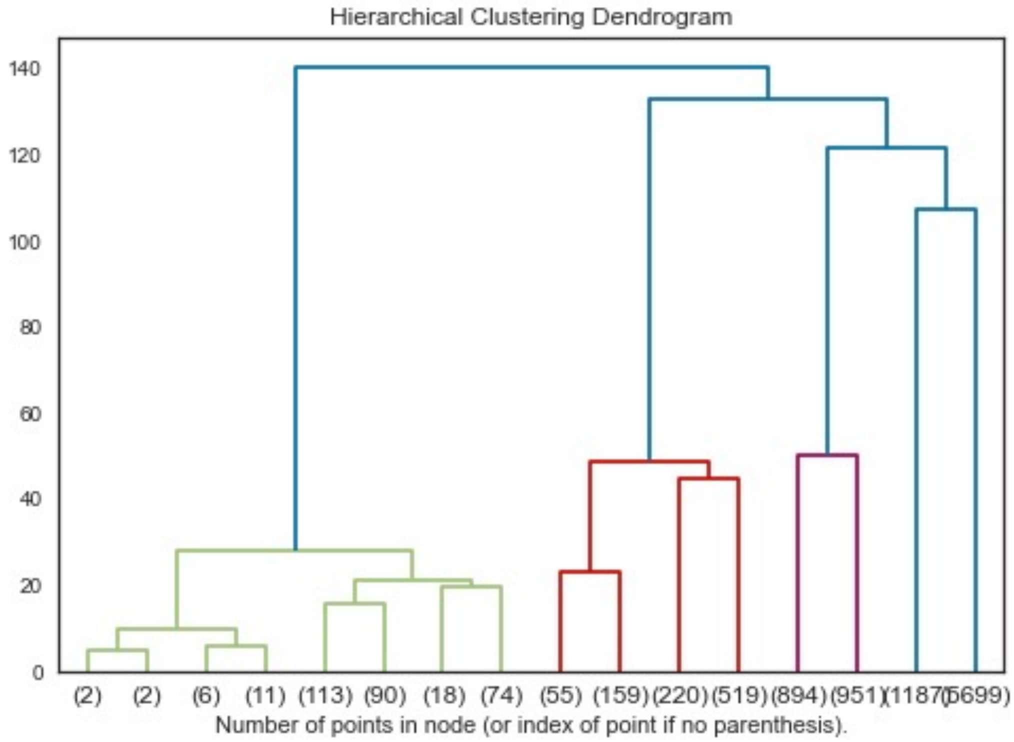
Hierarchical clustering



CLUSTERING

Selection of more features as an alternative to RFM

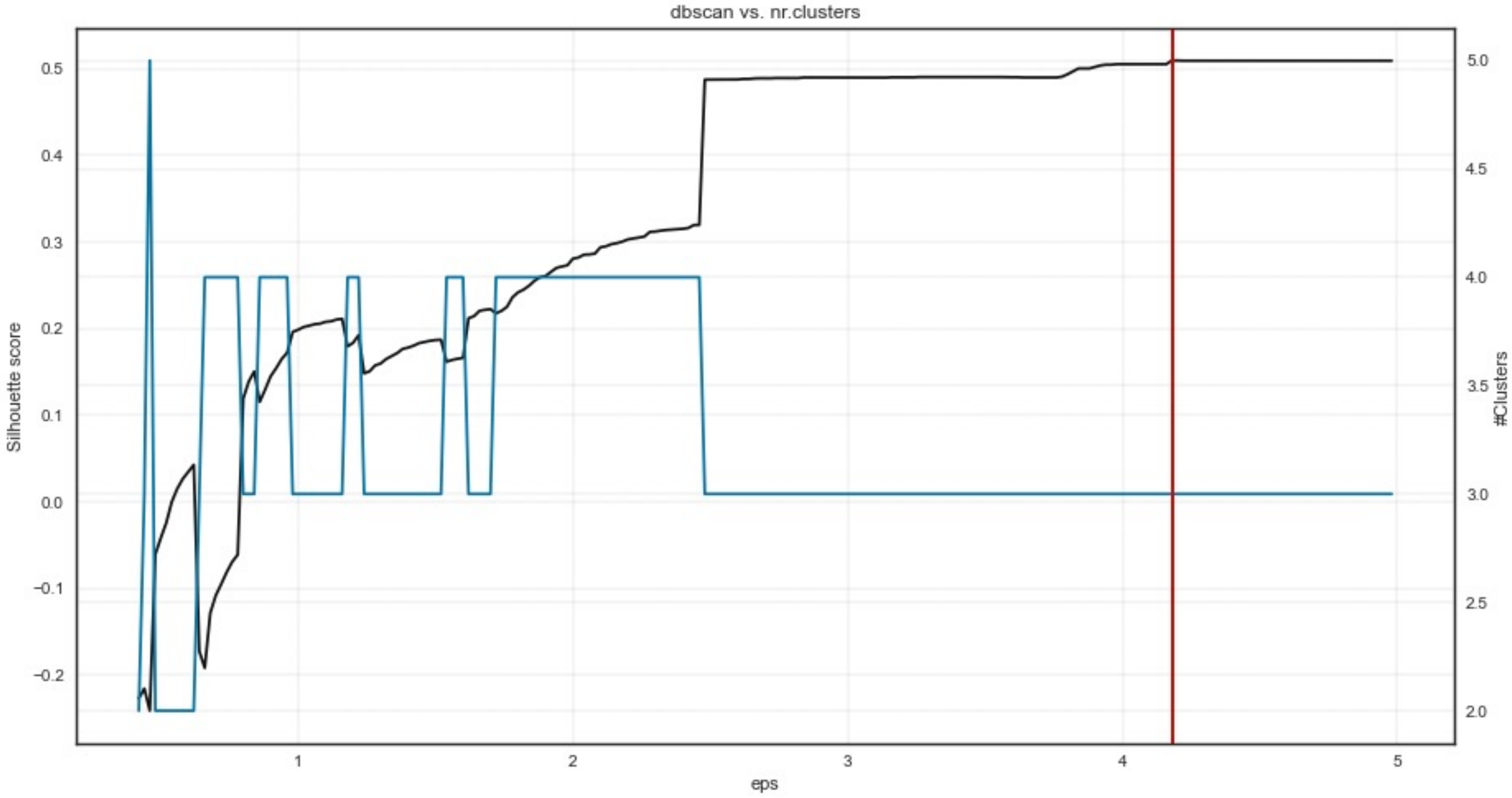
Hierarchical clustering



CLUSTERING

Selection of more features as an alternative to RFM

DBScan clustering

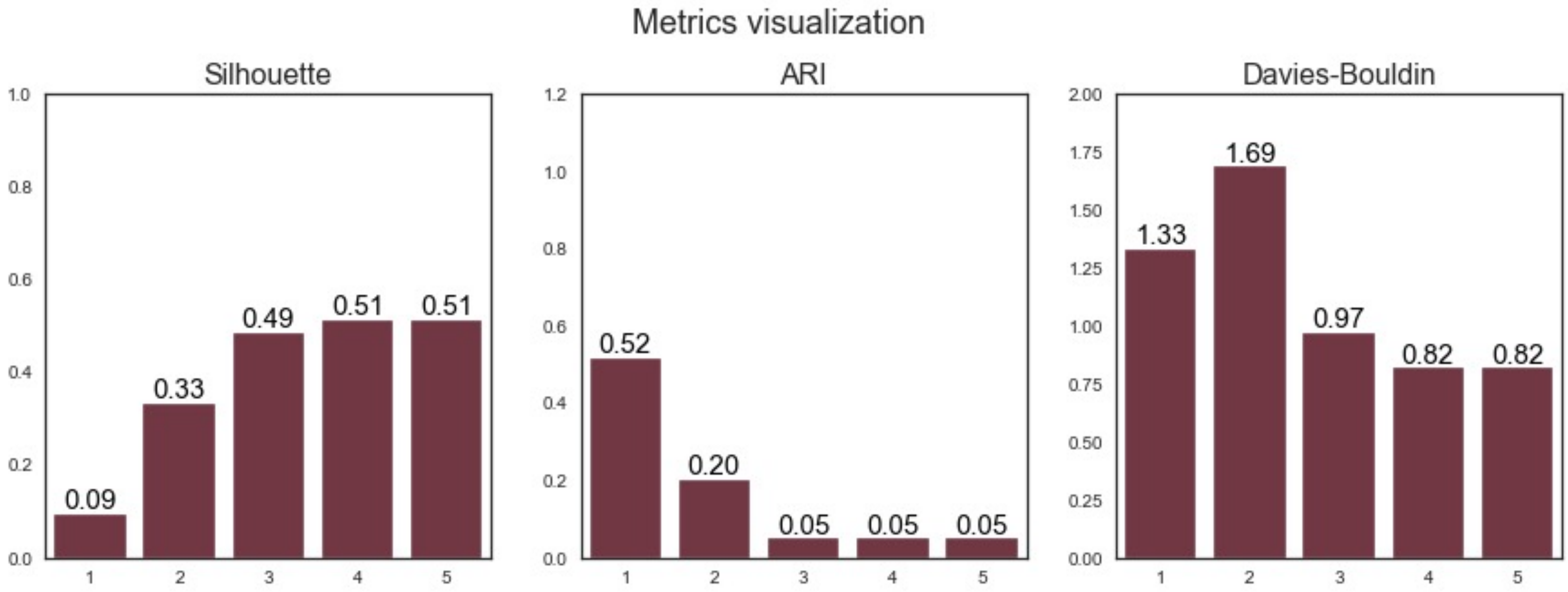


CLUSTERING

Selection of more features as an alternative to RFM

DBScan clustering

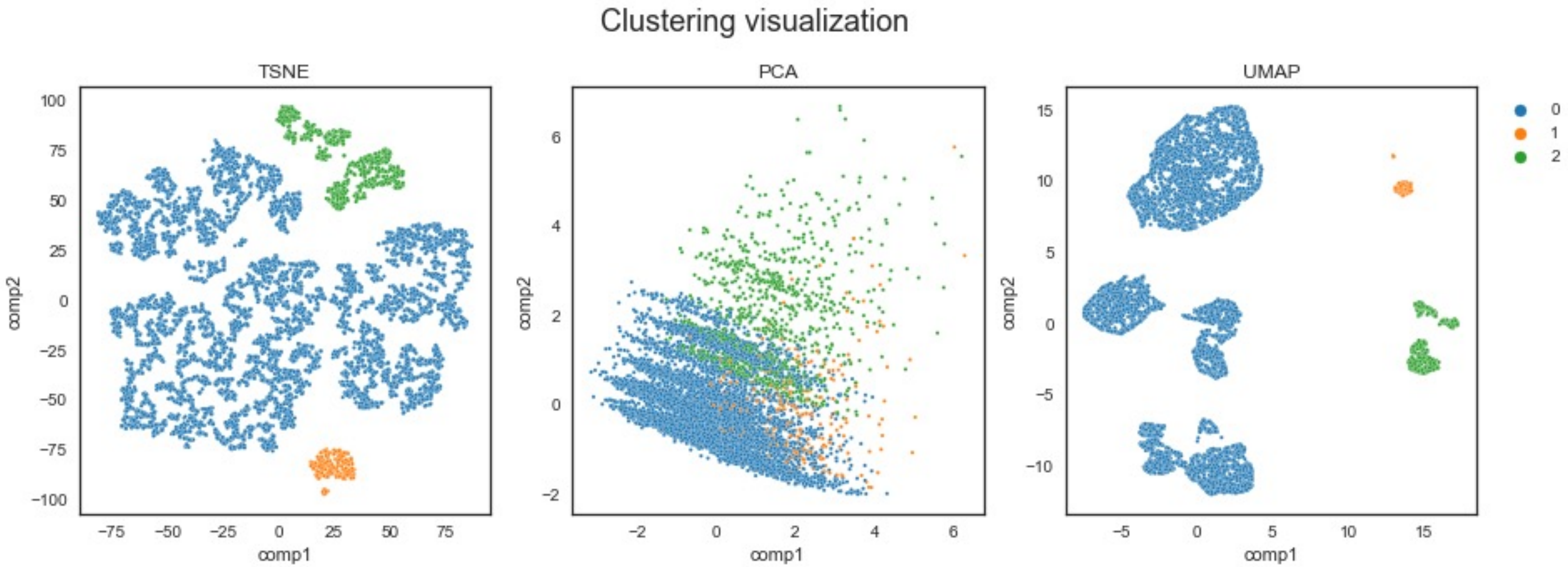
- Metrics confirm eps = 4 gives best results for silhouette



CLUSTERING

Selection of more features as an alternative to RFM

DBScan clustering

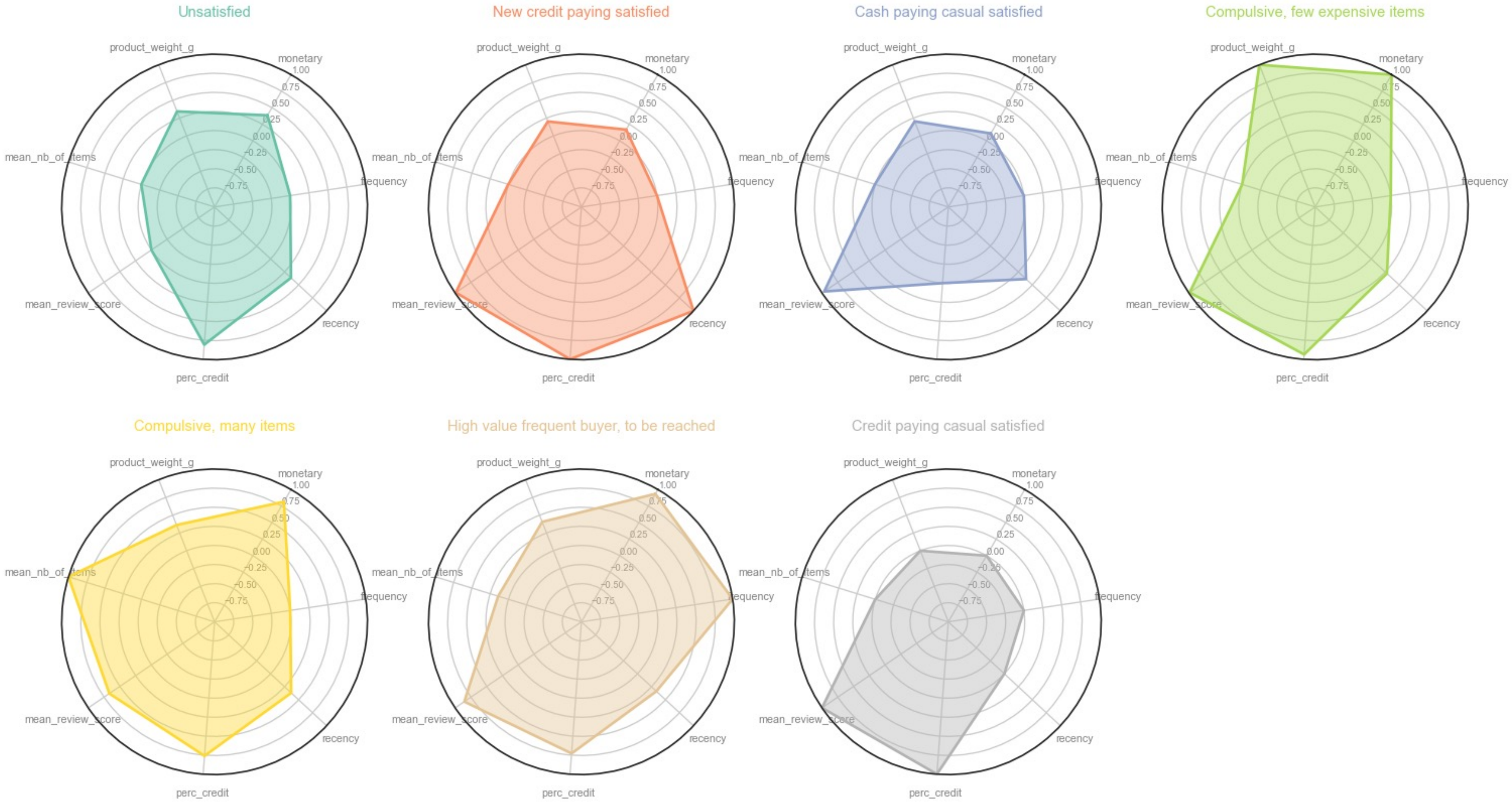


CLUSTERING

Selection of more features as an alternative to RFM

Best model and radar plot

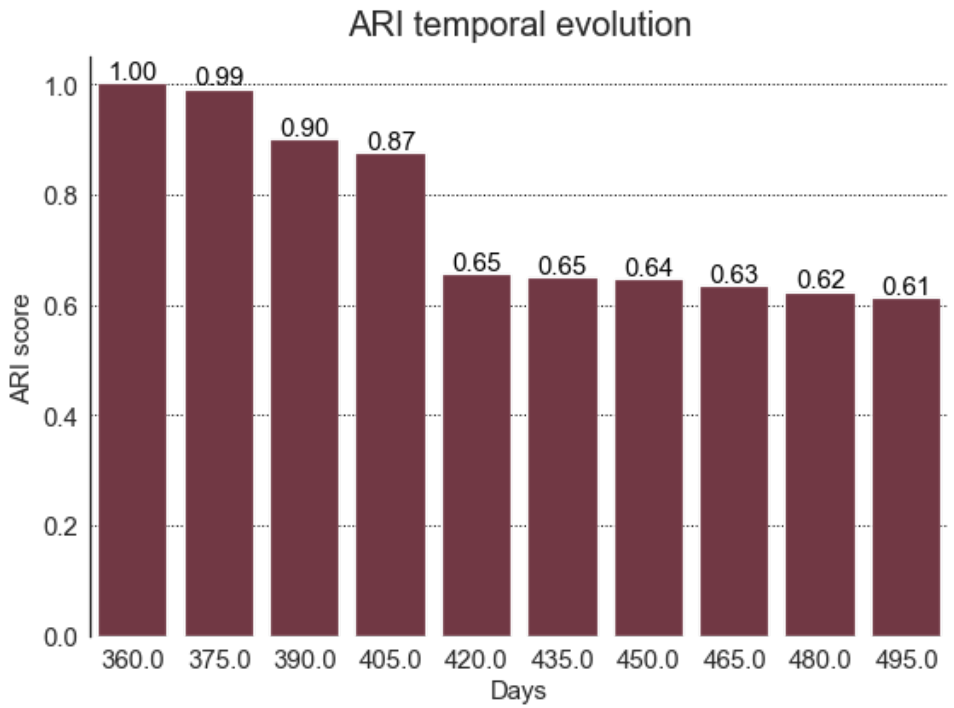
- K-means has the best stability according to ARI score, it is kept with 7 clusters


Unsatisfied
New credit paying satisfied
Cash paying casual satisfied
Compulsive, few expensive items
Compulsive, many items
High value frequent buyer, to be reached
Credit paying casual satisfied
product_weight_g
monetary
frequency
recency
perc_credit
mean_review_score
mean_nb_of_items
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75

MAINTENANCE

K-means clustering with 7 clusters

- ARI score as a function of time
- There seems to be a drop after 45 days



04

CONCLUSION

CONCLUSION

- Data available from October 2016 to October 2018
- 9 Dataframes for a total of 99k customers id
  - Customer, Seller, Geolocation
  - Order data, order payment, order review, order items
  - Product, product translation
- Objective: cluster customers with unsupervised learning
- Only 3% of customers buy more than once, RFM is therefore limited
- Best clustering model: K-means, with 7 clusters
- Maintenance proposed: every 45 days, based on stability

THANK YOU

Victor Benard