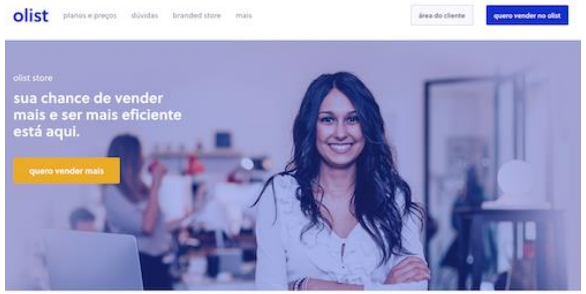
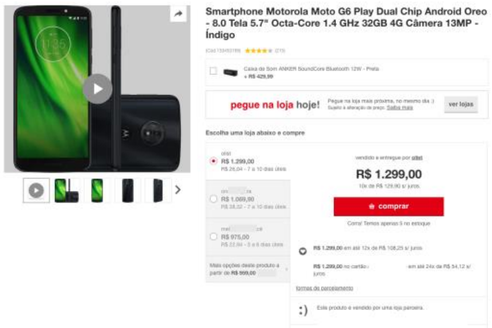
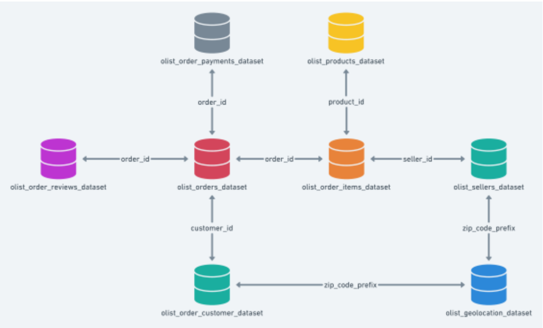
Olist Marketplace:
Customer Segmentation

✓ Olist datasets shared through Kaggle*,
✓ Improve marketing team's **customer understanding**,
✓ Through a usable **segmentation**,
✓ Identifying a right **update interval**







3 steps :

1. Perform EDA & Feature Engineering to **enhance** data
2. **Explore** a variety of **modell**ing approaches
3. Assess **actionability** and **stability**

* https://www.kaggle.com/olistbr/brazilian-ecommerce

Data Science : your best Support

1. **What can emerge out of data ?**
    a. **Exploratory Data Analysis**, toward **valuable Customer-Centric** data
    b. **Feature Engineering**: You Set the Limits, Pick Your Favourite !
    c. Refine your target : refine Use Cases

POlist_01_NotebookEDAandFE

jupyter

2. **What is a good segmentation ?**

    The most usefull Features, i.e. with adequate ***type*** and ***distribution***

    The most efficient Models, i.e. with adequate ***« sensitivity »***

    The most relevant Metrics, i.e. with ***meaningfull*** results

3. **How to achieve Olist business goals ?**
    a. Actionability
    b. Stability assessment
    c. Results & further proceedings

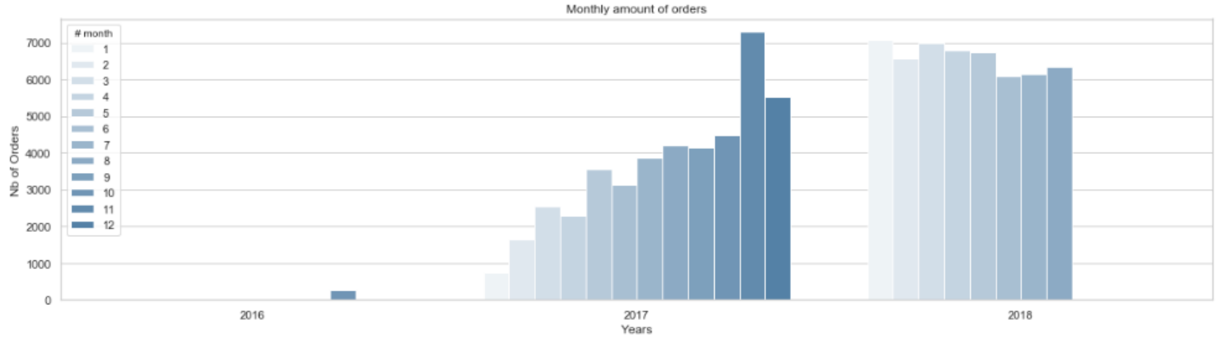
POlist_02_NotebookModels

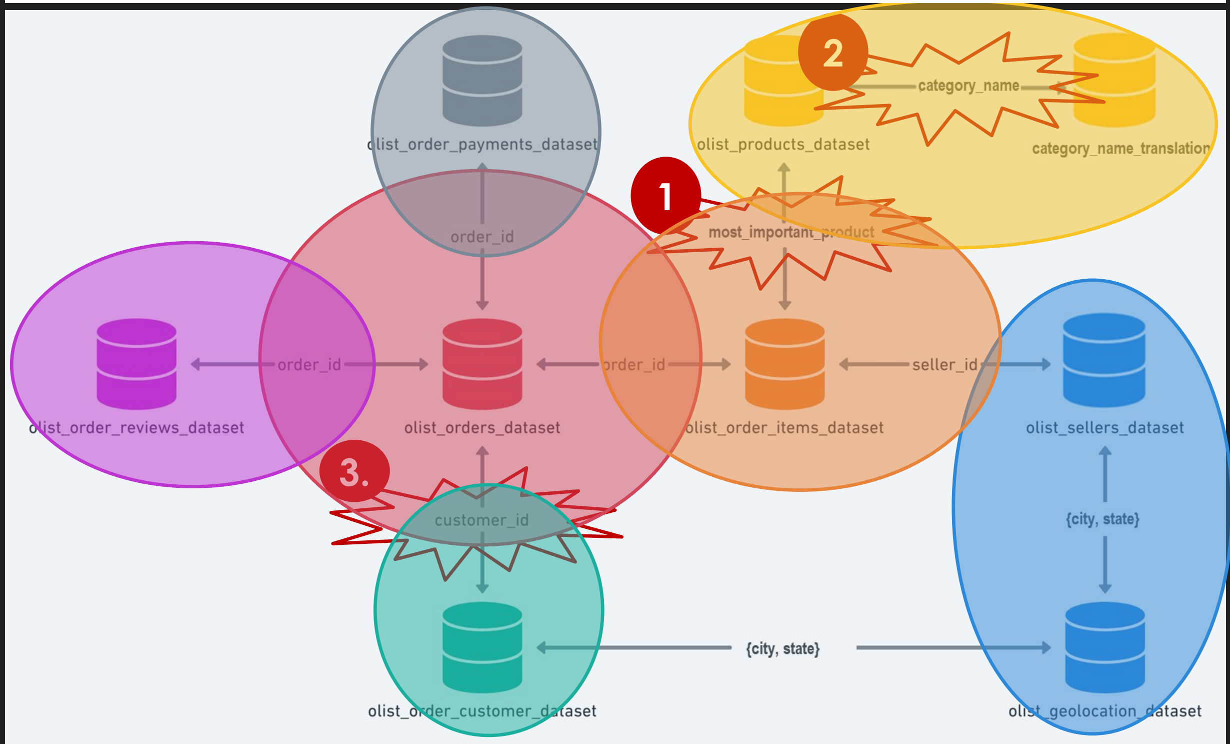
jupyter

1.a. EDA: Merge a Customer-Centric Dataset

- **Data Truncature**
  - **Orders number rise from end 2016 and is stable in 2018**
  - **Only 3% of Customers** (3k) made more than a **Single Order!**
    - **RFM*** techniques are **not valid**
  - **Only 3 % of Orders are not single_product** & 10% multi-item in shopping carts
    - Simplified **1:1** cardinality for {Customer_unique – Order – Product – Review} is **valid**

* ***R****ecency and* ***F****requency would require the knowledge of multiple timestamped orders and senseful anteriority.*





- ***From Order-Centric to Customer-Centric data:***
  1. Focused on the **most_important_product** (*of highest value*)
  2. Attached a **category** to the product
  3. Keeping for any customer_unique_id, the « ***single_product*** » & last delivered **Orders**

- **Let's browse some Customer-Centric features !**

1.b. FE: Engineer Customer-Centric Features

While Order-Centric datasets enable many calculation with groupby and consolidation by merge,

Client-Centric features **can be engineered** to get the **« Who » : Customers Groups**, e.g. *by studying :*

- **What :** the product, its value and price *(the « M » criteria of RFM)*
  - The « charm price » *(price with a « 9, 99 or x90 » termination)*
  - The product category, *and its caracteristics : size, weight*
- **When** : the purchase_time_zone *(as a clustering of purchase_dayofweek & purchase hour)*
- **Why** :
  - the Review Scores interest and behaviour, *as well as the « popularity » of the product or its seller*.
  - the quality of product's description.
- **Where** : the Customer-Seller distance, linked to the delivery time and freight cost.
  - each item has the freight calculated accordingly to its measures & freight value is splitted between items
- **How** :
  - The kind of payment, with payment_type, installements size, …
  - the review score given by the customer

1.b. Overview : the unlimited Feature Engineering field

How : main_payment_type(_cat)
payment_installments_size(_cat)
payment_sequence_size(_cat)
What : payment_total*
main_payment_value

What : product_cat
product_weight_g
product_size
product_density
Why : product_qlty_idx*
product_photos_qty
product_description_length
product_name_length
product_sales_count
product_revenue

When : purchase_time_zone_cat
order_purchase_dayofweek
order_purchase_hour
*delivery_vs_estimated**
estimated_delivery_time
effective_delivery_time

What : total_price*
charmed_price(_cat)
*freight_percentage**
total_freight
items_qty
product_price
product_freight

Why : product_review_mean*
product_review_count
customer_review_count
How : review_gap*
*review_answer_delay**
customer_review_mean

Where : cust_sell_dist*
seller_city &state
customer_city & state
Why : *seller_sales_count*
seller_revenue
seller_main_product_cat

nb *: Most features can also be derived as an ordinal « **level** », according to your **rules** or **targets**

Pick your favorite feature out of :
- **42 numerical**
- **10 categorical / ordinal**

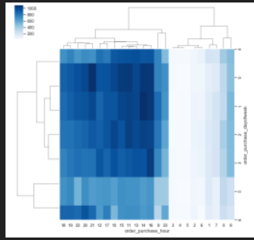
1.c. Refine your Goals, Find the Right Target

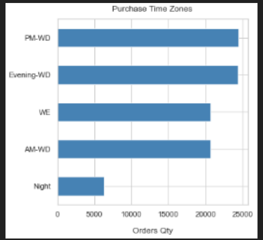
*RFM easy **actionability** has emerged during years of practices, is now enhanced by Machine Learning. You shall not « put » customers in a frozen matrix anymore but **drive** your ability to « learn » from data. Let's see practical examples :*

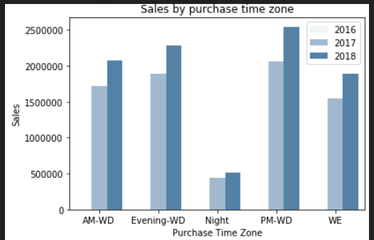
The « Right » features

- **The right time :**

purchase_time_zone_cat : *build purchase time zone (through hierarchical clustering technique out of day & hour timestamp)*







- **The right satisfaction level :**

review_gap : *value the gap between product and customer review, to define who's a worst, same or better scorer.*

- **The right product :**

Product_review_mean : *« stars » influence*

- **The right product description :**

product_qlty_idx : *e.g. build a product description index*

- **The right pricing and its attractivity :**

Total_price & charmed_price : *« charmed » by 0,99 termination*

- **The right location :**

cust_sell_distance : *so far so close thanks to a virtual marketplace*

*The most **powerfull & efficient** Customer's segmentation shall first **suit to Your Goals***

Use Case :
building a « Right » communication campaign

Why : product_qlty_idx(_lvl)

When : purchase_time_zone_cat

What : product_price(_lvl)
charmed_price_cat

Why : product_review_mean(_lvl)
How : review_gap(_lvl)

Where :
cust_sell_dist(_lvl)

We'll use this **shorlist** of **7 features**, incl. 2 of type Categories, and 5 Numerical (with their derivation as Ordinal)

Data Science : your best Support

1. **We're now able to define Use Cases and refine targets**
2. **Next, what is a good segmentation ?**

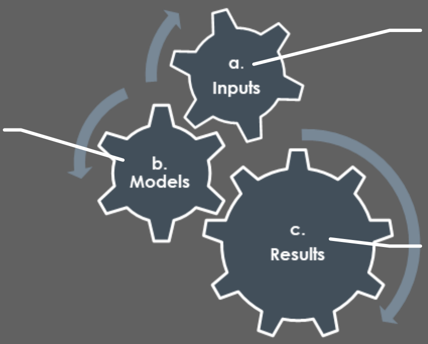
   The most usefull Features, i.e. with adequate ***type*** and ***shape***

   The most efficient Models, i.e. with adequate ***sensitivity***

   The most relevant Metrics, i.e. with ***meaningfull*** results

3 interdependent steps, with a resulting « sensitivity »

Initialize & « feed » clusters

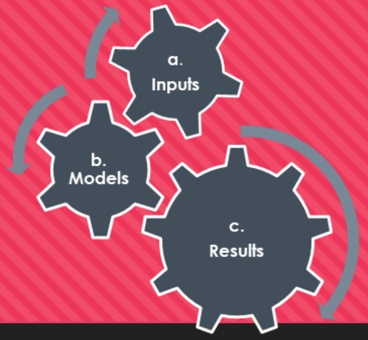


Transform inputs
Reduce dimension

Find the optimal solution

Consistent calculation of « distance » and its generalization

Right combination of transformers, reducers, models, metrics *must be* ***chosen*** *and* ***tuned*** *to* ***enhance results***

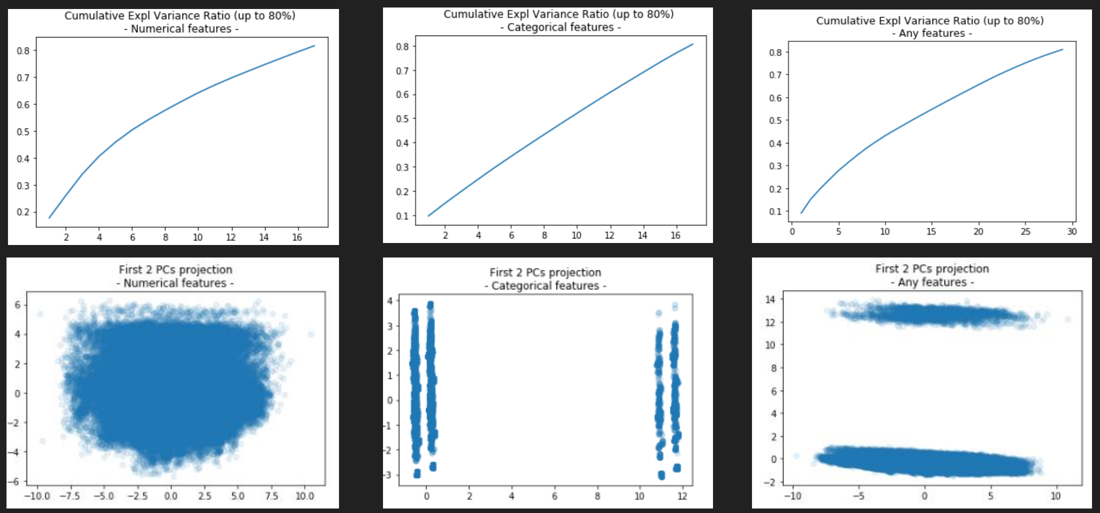


Toward the best segmentation

2.a. Issue : mixed type of features

- **Aim** is to deal *equally* with features of any type
- **Transformers** : onehot encoding / quantile transformer, scaled
- **Reducer** : hereby, resulting PCA's cumulative explained variance **increase linearly**

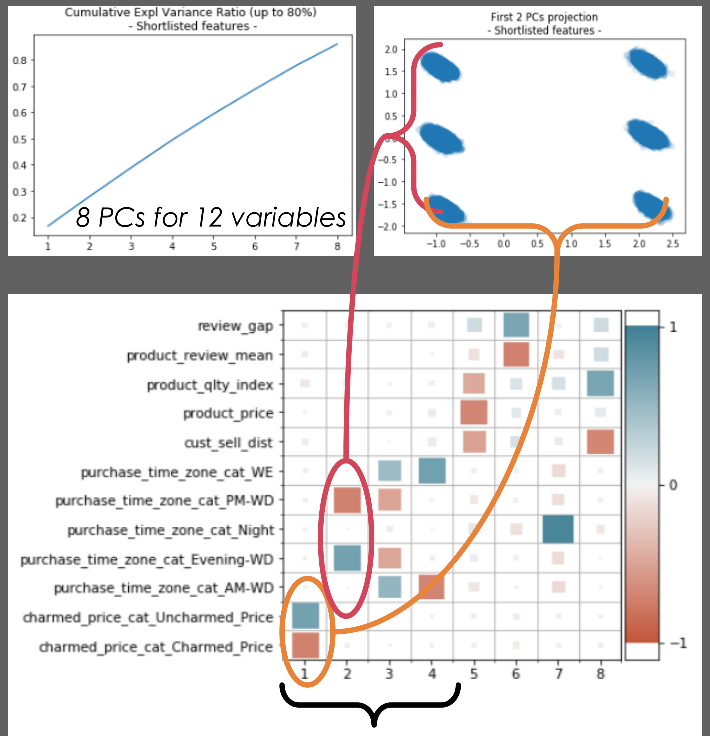
Only numerical (dim 42) | *Only categorical (dim 43)* | *Any features (dim 85)*



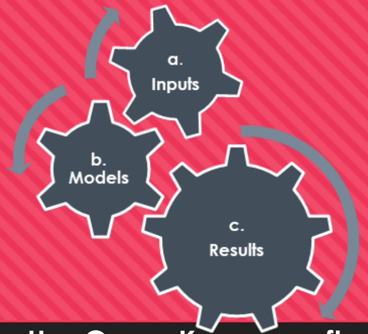
***Discrete** features highly **affect results** for such reducer*

Use Case Focus *(dim 12)*

Cumul. expl. Variance | PC1-PC2 projection



4 firsts PCs taken by « _cat »

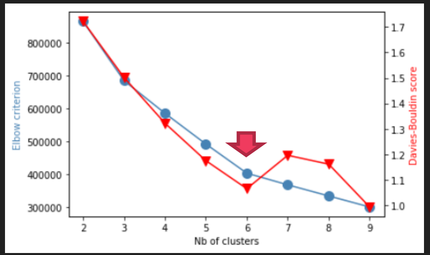


Toward the best segmentation

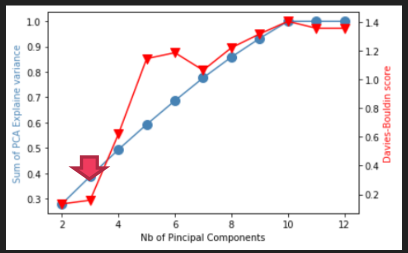
2.b. consistent reducer & model & meaningfull results

Use Case : K-means after PCA

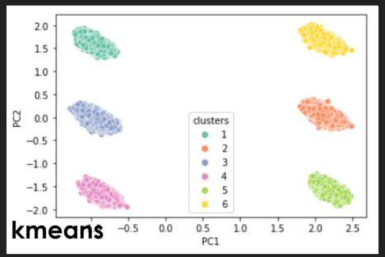
Optimal nClusters : 6



Optimal nComponents : 2

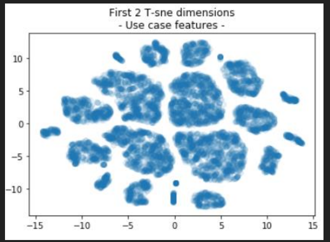


6 clusters projection PC1-PC2

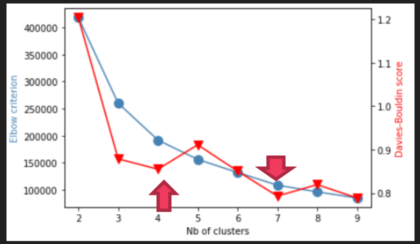


Use Case : alternate dimension « reducers » & K-Means

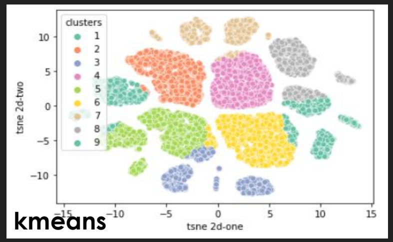
t-SNE 2d



t-SNE 2d Optimal k : 4 or 9

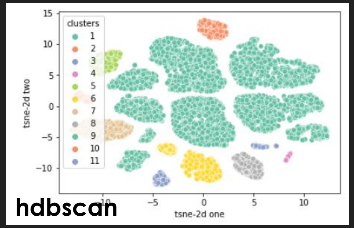


t-SNE with 9 Clusters

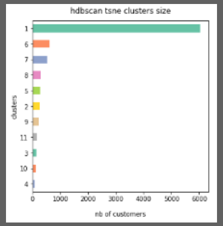


Use Case : alternate clusterer HDBSCAN - agglomerative

After t-sne



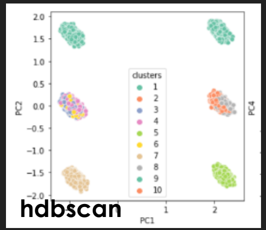
Cluster's balance



Discrimiating features

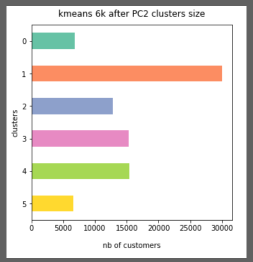
None !

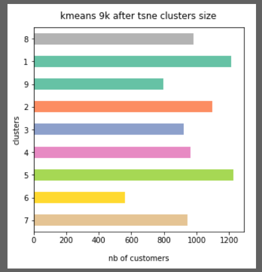
after PCA 80%

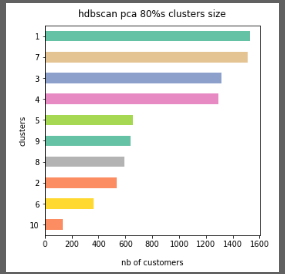


- Qualitative results -

Cluster's balance

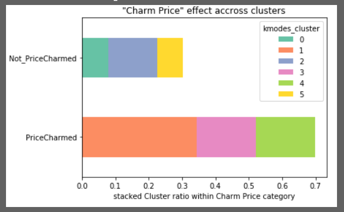




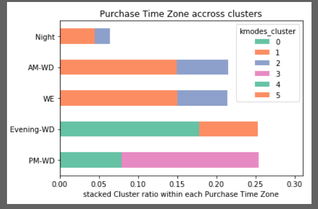


Discriminating features

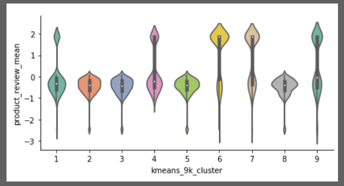
Charm_price_cat



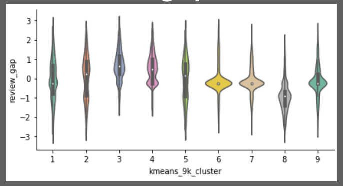
Purchase_time_zone_cat



Product_review_mean

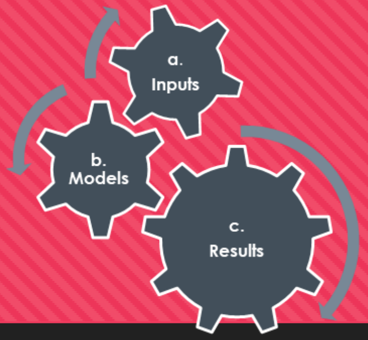


Review_gap



None !

Machine learning offers a large palette of alternate dimension reducers or clustering techniques that could be explored
Depending on the goal we want to achieve



Toward the best segmentation
2.c. Meaningfull results, what about actionability?

- **Quantitative** evaluation:
  - DB-Index, same ground
  - Silhouette Score, various ground (normalized)
  - Duration (time spent to compute)
- **Qualitative** evaluation:
  - Cluster balance
  - Discriminatory features
  - Actionability

Intermediate results:

Clusterers	**K-means**				**HDBSCAN**			
Reducers	none	pca (80%)	pca 2 PC	tsne	none	pca (80%)	pca 2 PC	tsne
Duration	fast	fast	fast	slow	slow	slow	slow	slow²
optimal cluster number	**6**	**6**	**6**	9	20	10	2	11
silhouette score	0.257372	0.326331	**0.904481**	0.404643	0.130021	0.399960	**0.634935**	0.006387
cluster balance	average	**good**	**bad**	average	-	average	average	**very bad**
***discriminatory* features**	2 / 7	2 / 7	2 / 7	2 / 7	-	**none**	**none**	**none**
actionability	poor	biased (categories)		poor	**bad**			

- to emphasize actionability : explore alternate approaches to reach most valuable **qualitative** results
  - **Option 1** : Introduce **weighted** techniques to ensure desired feature to be discriminant
    - ***Pre-requisite is a refined and stable target***
  - **Option 2 (selected)** : K-Modes/K-Prototype extension, to assert actions considering categorical features

Data Science : your best Support

1. **We're now able to define Use Cases and refine targets**
2. **We know how to select and tune an approach**
3. **Next, how to achieve Olist business goals ?**
   a. Actionability
   b. Stability assessment
   c. Results & further proceedings

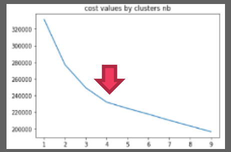
3.a. Actionability Explore K-Modes

- **What :** k-Modes extends k-Means algorithm for categorical features, minimizing a cost function measuring **matching dissimilarity**
  - **Pros :**
    - « raw » data (turned to categorical)
    - **clear cluster description**
    - *deterministic (with « Cao » init)*
  - **Cons :**
    - **introduce a sensitivity to feature engineering : reward same k optimal clusters than feature discretization**
    - would consider ordinal as categorical (losing the « real » distance between levels)
    - stability is compromized because of its sensitivity to discretization

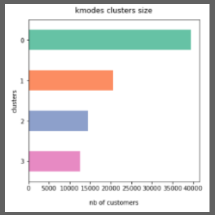
- **K-Modes gets a direct « cluster Zero » description : feature's most frequent values (n_init=1)**

  Once optimal settings found : Optimal n_init=3 and max_iter=30

1. **K-Modes build clusters iteratively until cost slows its decrease**

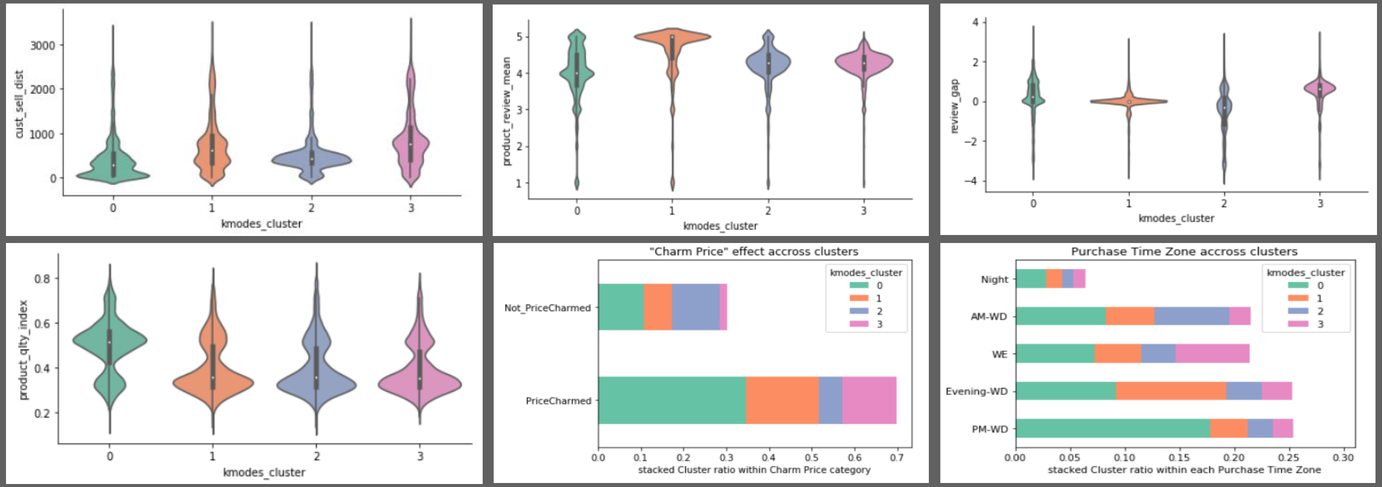


2. **Resulting cluster's balance**



3. **Goal: Mean Price**

4. **Find your target**



5. **Refine your use case : what matters most?**
6. **Re-engineer your features accordingly**

3a. Actionability K-Modes results

- Cluster 0 : Afternoon Buyers Majority
  - Customer of the largest segment, they claim a good product description of below mean priced products. They seem satisfied by low review score while giving a better score. They live the closest to commercial areas and seems not sensitive to charm pricing. They spread across any purchase time zone.
    - *['Near_Dist', 'Light_Price', 'High_QltyIdx', 'Low_Score', 'Better_Review', 'Charmed_Price', 'PM-WD'],*
- Cluster 1 : Evening Best Buyers
  - Customers of the interesting second largest segment, buy more expensive products, no matter their description's quality and are not sensitive to charm pricing. They live far from the sellers, meaning they could not get to stores. Top review score seems mandatory to them while they score the same. Their favorite purchase time zone is the evening of a working day.
    - *['Far_Dist', 'Medium_Price', 'Low_QltyIdx', 'Top_Score', 'Same_Review', 'Charmed_Price', 'Evening-WD']*
- Cluster 2 : Morning worst Reviewers
  - Customers of the second smallest segment are the worst reviewers while purchasing medium scored products, not matter their description's quality. They are located around the median distance to sellers but live already two far to get those shops other than virtually. These customer seems to reject charm pricing. Their favorite purchase time zone is the morning of a working day. They buy more often products of Electronics, Computers & Accessories.
  - *['AroundMed_Dist', 'Light_Price', 'Low_QltyIdx', 'Medium_Score', 'Worst_Review', 'Uncharmed_Price', 'AM-WD']*
- Cluster 3 : Week-end Best Reviewers
  - Customers of the smallest segment are the best reviewers while purchasing medium scored products, not matter their description's quality. They are the farthest customers. They have the highest sensitivity to charm pricing. Their favorite purchase time zone is the week-end. They buy more often products of Telephony, Supplies and Health Beauty Baby Categories.
    - *['Far_Dist', 'Light_Price', 'Low_QltyIdx', 'Medium_Score', 'Better_Review', 'Charmed_Price', 'WE']*

With basic goal of sales increase :

- **Action 1: improve scoring,** targeting **cluster 3** customers, i.e. mainly during the week-end, catching them on the charm price sensitivity, arguing that they can afford any products thanks to the marketplace, no matter the live far from the original commercial areas (action about freight fares to study). Additional action targeting **cluster 2** could be, mainly during the morning, to fasten regular cart.
- **Action 2: improve sales,** targeting **cluster 1** customers, i.e. mainly during the evening of a working-day, catching them on the top review scores and arguing that those selected products are now available thanks to the marketplace (new sellers joined, top ratings).

3.b. Stability assessment Choice of baseline & further action

Stability is asked to define the timeframe for maintenance actions.

Remember stability is biased due to data troncature, and marketing should decide either to keep « rising » period (2017) or focus only on « stable » period (2018)

***By comparison*:**
To keep your customers in your target

- Stability assessment by **comparison**
  - Compute segmentation on 2 similar periods
  - **Re-map clusters according to centroïds**
  - **Measure deviation of centroïds coordinates**
  - If needed : refine categories or levels
  - Clear understanding of **target** is a pre-requisite

Action*: review features *periodically
to ensure customers matching your target

***By aggregation – deviation*:**
To adapt your target to customers

- Stability assessment measuring **dissimilarity**
  - Compute segmentation on a baseline, whatever its size :
    - allow larger baseline and smaller additions
  - **Aggregation : put new data (e.g. monthly)**
  - **Assess deviation computing adjusted Rand index**
  - **Target** may change according to new clustering

Action*: review clusters *periodically
to match your new customers

3.b. K-Prototype to remedy K-Modes high unstability

K-Modes: <u>By comparison</u> 2018 / 2017

- Due to **threshold definition**, risk is high to loose the essence of a Segment !
- Solution to restrain volatility is to work on feature's discretization :
- Most « unstable » features are :
  - product_review_mean_lvl, review_gap_lvl
  - Product_qlty_idx
- Here only 2 clusters remain « stable » through such comparison.

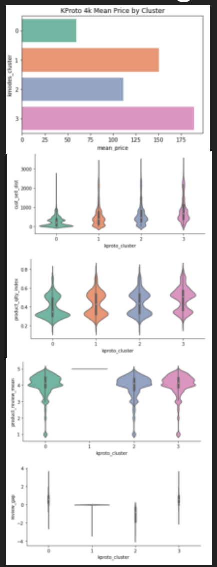
	Stable 2018			
Rising 2017	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Cluster 0	**2 / 7**			
Cluster 1		**4 / 7**		
Cluster 2			**3 / 7**	
Cluster 3				**2 / 7**

Bad stability uneasy to follow-up

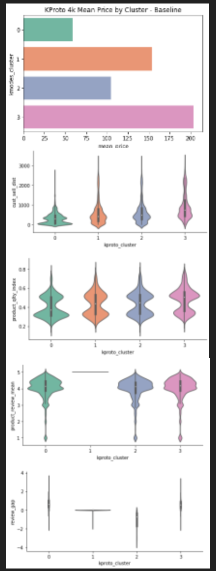
K-Prototype: <u>By comparison</u> 2018 / 2017

- **k-Prototypes** aim is to combine K-Means and K-Modes, again with cost **matching dissimilarity, enabling « really raw »** data and **remedy the bias** of considering ordinal as categorical (losing the « real » distance between levels)

2017 - rising



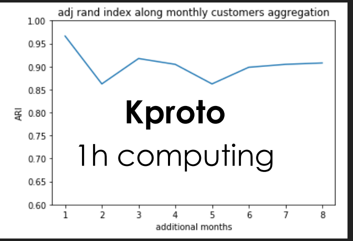
2018 - stable



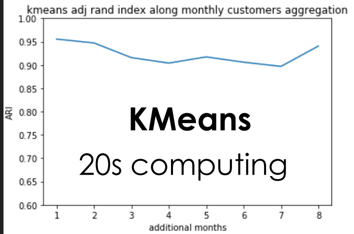
With clusters & cat balance unchanged

K-Prototype: <u>By aggregation - deviation</u>

ARI does not fall while increasing customers number taken into account
Its stability is even better on higher timeframe



Same trend for a basic K-Means



"Good" stability, easy to follow-up

3.c. Results & further proceedings

Final results:

Clusterers	K-means	HDBSCAN	KModes	KPrototype
Best Reducers	pca	pca	*not used*	*not used*
Duration	fast	slow	fast	**very slow**
Get optimal cluster number	good	bad	good	good
Best Silhouette or Cost	good	average	good	average
Best cluster balance	average	average	good	good
Best discriminatory features	2 / 7	**none**	7/7	5/7
Best actionability	biased	**bad**	easy	good
stability	good	-	**bad**	good

Further proceedings, recommendations:

1. Refine your target (i.e. use case) through a Kmode / Kprototype rough segmentation for an **easy actionability**

2. Optimize both **Quantitative + Qualitative** results through relevant technique (Kmeans or Kprototype if mixed feature types, or even Kmodes through the right FE)

Alternate option: explore Weighted Kmeans (or K-Prototype) … *as long as we know the target to achieve*

- Thank you for your time
- Any questions?