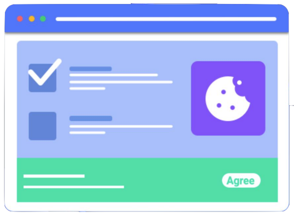
Trying to estimate web behaviors after GDPR regulation

Charles LIETAR - Omar EL HAJJAR - Clara JOUY - Xavier MELLEVILLE - Hojun JUNG

TABLE OF CONTENTS

01

Framing the project



Problem Statement

Tracking online data is more difficult

With the GDPR regulations evolving since May 2018, the usage of cookies to track online data has become more difficult.

Data loss in tracked traffic is growing

Since more and more users either refuse all or do not give preferences, the non tracked traffic on the website has become more important.

Understanding web visitors' behaviors is harder

Peugeot will lose a large volume of web visitors information, making it harder to understand their online behaviors

Our Goal

Extract historical patterns

Using historical user sessions with tracking consent, the goal is to capture session patterns on a daily level

Estimate overall user behaviors

By modelling daily captured patterns, the objective is to estimate overall user behaviors on a given day

4 target KPIs to quantify user behaviors

Specific metrics have been selected to mathematically quantify daily user behaviors on Peugeot's website :

Qualified sessions

% of sessions during which the user did not bounce

Sessions with vehicle

% of sessions during which at least one vehicle was seen by the user

Started configurations

% of sessions during which the user started a vehicle configuration but did not finish it

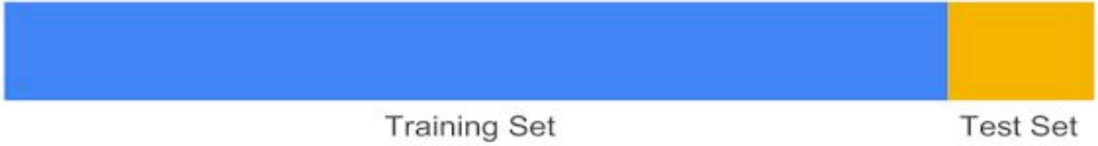
Engaged configurations

% of sessions during which the user started a vehicle configuration and finished it

Our data

User sessions with tracking consent until 08/08/2020, described by **more than 40 attributes** (number of hits on product pages, number of product added to the cart, which vehicle pages the user consulted…)

Historical sessions are **split into training and test sessions**



Training sessions: to learn session patterns and thus to model the overall user behaviors

Period: 558 days
from 01/01/2019
until 11/07/2020

Test sessions: to simulate the tracked data loss and evaluate the target KPIs predictions

Period: 28 days
from 12/07/2020
until 08/08/2020

Pipeline

How to group similar sessions?

Using clustering algorithms

How to segment days?

Using clustering algorithms as well

How to retrieve similar days ?

Using distance computation with the days from the same cluster

Clustering | Describing | Segmenting | Simulating | Comparing | Extrapolating

How to describe a day using its sessions ?

Using its session-cluster mix

How to simulate real data for test?

Sampling the sessions for a given test day

How to predict the KPIs ?

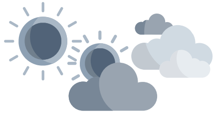
Using empirical and statistical extrapolation methods

INPUT

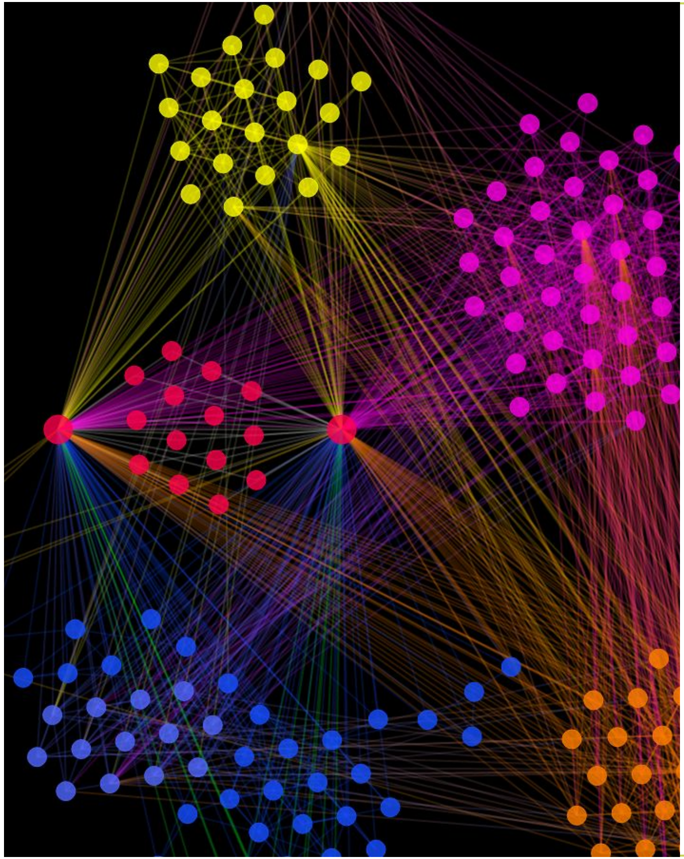
Web Sessions of a given day with data loss

MODEL

OUTPUT



KPIs estimations for that day



02

Segmenting our data

Clustering the sessions

From a session to a daily level

Activity	Key	Insight
Bounce	Bouncer	Lowest activity
Low	VEH focused	No time to waste
	Browser	Not sure what I'm looking for
	Curious	Check everything quickly
Medium	VEH focused	Know what I want
	Browser	Not sure what I'm looking for
High	Avid Users	Want to know everything

Initial database
About 25 KPI's describing each sessions (hits, date, medium….)

Data Preparation
(encrypting…)

KNN & Mini-batch clustering
(maximization of inter-cluster distance)

7 'types' of sessions
(see right)

Grouping by day

Daily level database

Date	Bouncer	Low/ VEH foc	Low/ Browser	Low/ Curious	Medium/ Browser	Medium/ Curious	High/ ActiveU
23/02/2019	69152	20342	2369	6624	20790	1626	5058
24/10/2019	80423	24144	3340	7340	52168	3624	5657
15/05/2019	75021	26089	83534	6416	27445	2042	5806
19/05/2020	87302	22474	1933	7954	13285	2428	6954
21/12/2019	69753	18876	2357	10292	30646	1877	4505

Final test and training databases qualify each day by the "types" of sessions that occured at this date

Segmenting the days - Approach

From sessions definitions to days definitions

Model building

1

Test of 3 clustering algorithms' performances for 2 to 10 clusters:

- K-Nearest Neighbors
-Birch algorithm
-Agglomerative clustering algorithm

Technical testing

2

Filtering the best algorithms with 2 measures of internal and external dispersion within/between clusters (one based on days, the other based on clusters)

Interpretability

3

Selecting the final algorithm on 3 interpretability criteria:

-The number of cluster
-The size of each cluster
-The interpretability of each cluster

Results

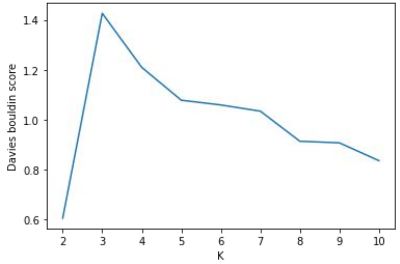
4

Model: Agglomerative clustering

# of clusters: 5

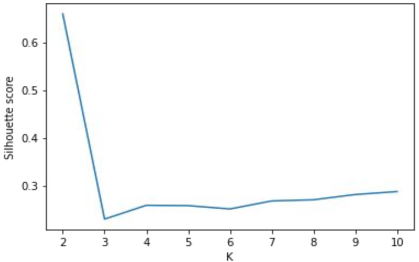
Result of a trade-off between performance and interpretability

Agglomerative clustering Davies Boulding score (left) and Silhouette score (right)



← To Minimize

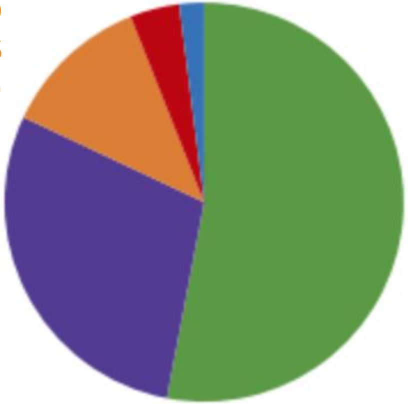
To Maximize →



Day-cluster presentation: overview

Our five clusters :

1. Active days
2. Pop-in days
3. Research days
4. Advanced interactions days
5. Configurations days



Key insights on their distribution :

Pop-in Days	The **highest variance** The **highest rate of bouncers** The **largest number of sessions with at least one vehicule** seen during the day: probably due to online advertising campaigns
Active & Research days	The **longest time** on the website Quite consistent (**low variance**)
Interactions & Configurations days	The **greatest interaction** with the website : The **largest number of configurations started and engaged** The **largest number of product details seen**

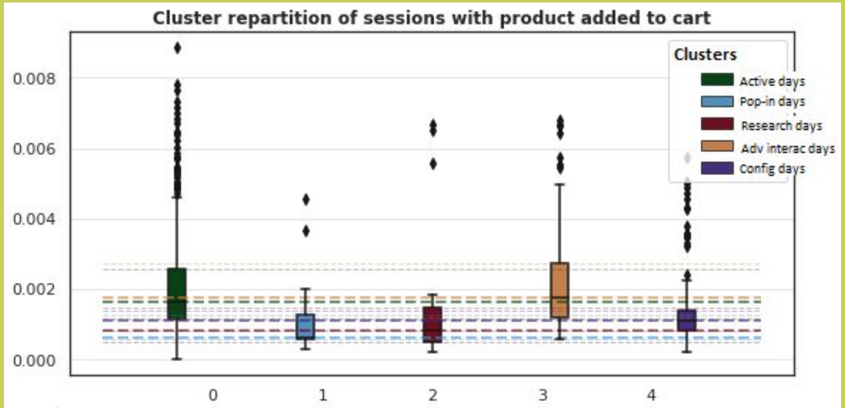
Day-cluster details

1. **Active Days:**

Majority of sessions of curious, browsers and avid users :
54.6% of sessions from ‘Curious’
4% of sessions from ‘Avid Users’

2. **Pop-In Days:**

Majority of sessions oriented toward vehicules :
39.5% of the sessions from ‘VEH focussed’



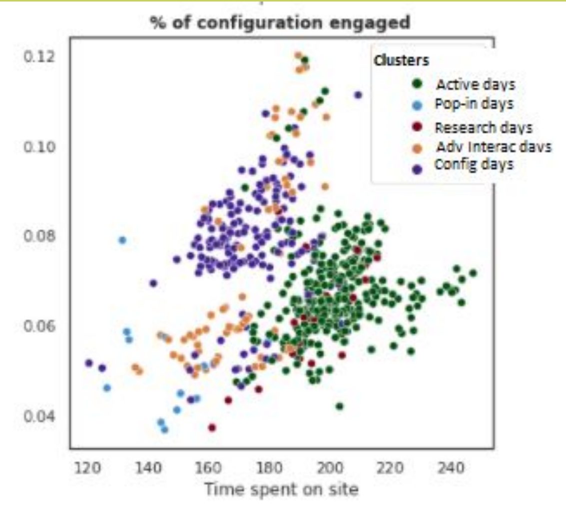


Figure 1, Scatter plot :
*The **advanced interactive days** and the **configuration days** have a **better ratio of engaged session/ time spent on site** than other clusters.*

Figure 2, boxplot:
*The **active days** cluster, the **advanced interactive days** and the **configuration days** are days during which **more products are added to cards.***

3. **Research Days:**

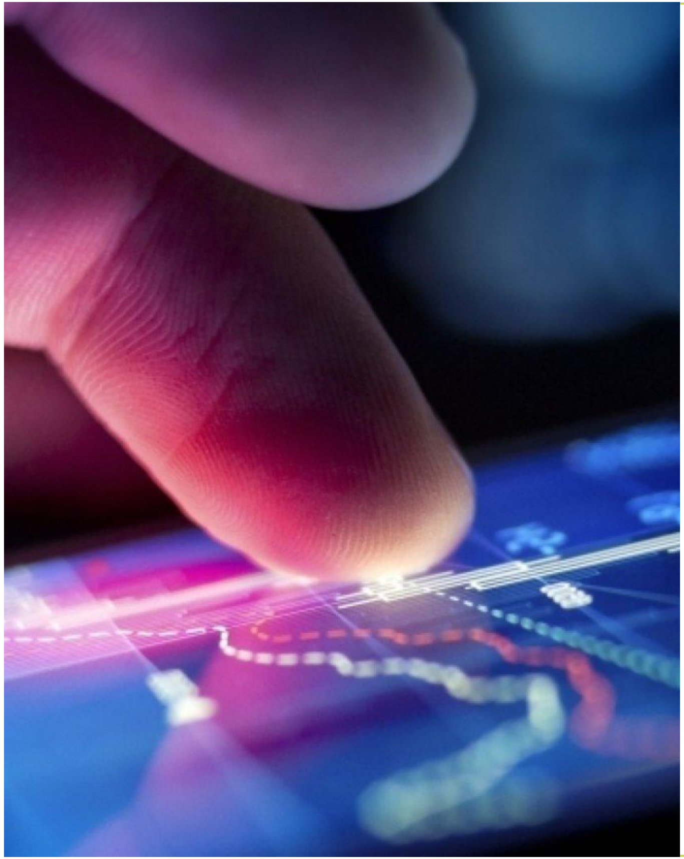
27% of sessions from ‘Browsers’
Lots of ‘VEH focussed’

4. **Advanced Interaction Days**:

Few time spent on the website
Lots of ‘VEH focussed’ and ‘Curious’ with few but targeted interactions

5. **Configuration Days:**

Mainly VEH focussed engaging with the brand
Lowest number ‘Curious’ with low interactions



03

Preparing test data

Preparing test data - Approach

1. SAMPLING TEST DAYS

Shifting from session-level to day level on test data : we simulate the tracked data loss due to new cookie consent banner, by keeping a sample of the test session in a given day

2.DESCRIBING THEM WITH SESSION-CLUSTER MIX

Calculating session-cluster repartition: we describe them by their session-cluster mix as we did for the training days

3. BUILDING DAY-CLUSTER CLASSIFIERS

Constructing day-cluster predictive models using the training days : we evaluate different multi-class classifiers to best predict on which day-cluster a given day can fall in

4. SELECTING BEST MODEL & DEPLOYING IT

Keeping the most accurate day-cluster classifier : we keep the best predictive model and deploy it on the sampled test days to predict their day-cluster

Sampling test days and describing them

	Days	Cluster Number
1.	Day 1	3
2.	Day 1	3
3	Day 1	2
...	...	...
10.	Day 1	1
11.	Day 2	2
12.	Day 2	4
13.	...	

For each test day, Randomly choose 70% of sessions

Get Repartition

Days	Session Cluster1	Session Cluster2	...	Session Cluster5
Day 1	30%	15%	...	25%
Day 2	23%	40%	...	12%
...				
Day 28	13%	50%	...	10%

Day Cluster
???
???
???

1 Dataset: Test sessions

2 Result: 70% Samples Repartition

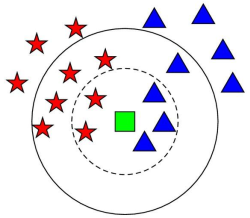
3 Goal: Predict Day Cluster using training days with their day-cluster

Predicting day-cluster on sampled test days

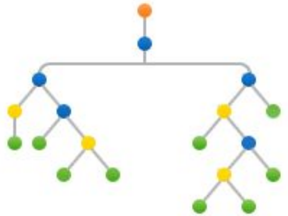
1 BUILDING

Constructing day-cluster classifiers using 2 methods by learning the classification rule from the training days, described by their session-cluster repartition

2 different methods :



Instance-based with **K-Nearest neighbors**



Tree-based with a single **decision tree**

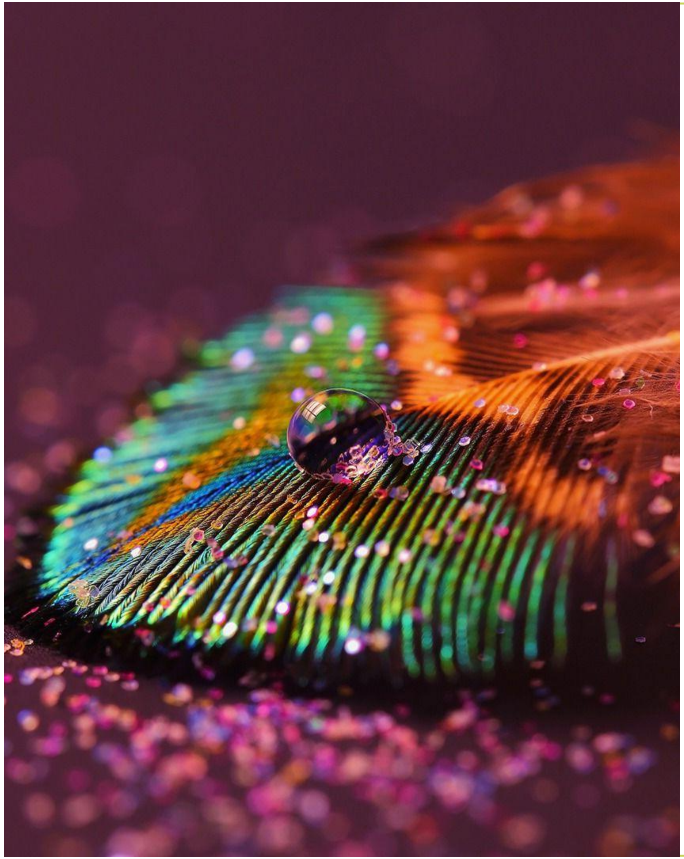
2 SELECTING

Checking accuracy metrics, and selecting the best method with optimal hyperparameters

Selected classifier : **K-Nearest neighbors** with **K = 2**

3 DEPLOYING

Deploying selected day-cluster classifier on the sampled test day to predict their cluster



04

Modelling and Selecting

Introduction and extrapolation techniques

<u>We have now:</u>

- **The sampled test days,** where we have tracked only 70% of the sessions, representing the kind of tracking per days we might have after new GDPR implementation.
- **The cluster these sampled test days belong to** and **the k most similar days from them** we can find in our training set (ie. the days for which we could track all the sessions)
- The **4 target KPIs** we want to predict **for each of these similar days**

<u>What we have and what is left to do</u>:

INPUT

Sampled days from the test dataset

25 attributes (sampled) characterizing the traffic on the website

K most similar days (in train set, with distance function)

4 target KPIs (for each similar days)

25 attributes characterizing the traffic on the website

OUTPUT

4 target KPI's for the sample days

Extrapolation techniques

- Simple average
- Average ponderated by the distance between test and the k similar training days
- Linear regression trained on the similar days

Model performance evaluation & selection

1

Defining **how to measure the distance** from one given day to its k-nearest days within its cluster

4 distances tested (different mathematical implications):
- Braycurtis
- Euclidean
- Correlation
- Canberra

2

Defining **the k number of nearest days we use for the extrapolation.** Between 5 and 100.

3

Computing predictions on KPI's for the three pre-selected extrapolation methods

3 preselected models:
-Average
-Ponderated average
-Linear regression

4

Comparing the prediction based on each **sampled days** VS **the true value** of the same entire days

5

Comparing the performance of each method for each KPI by computing **2 types of error rates.**

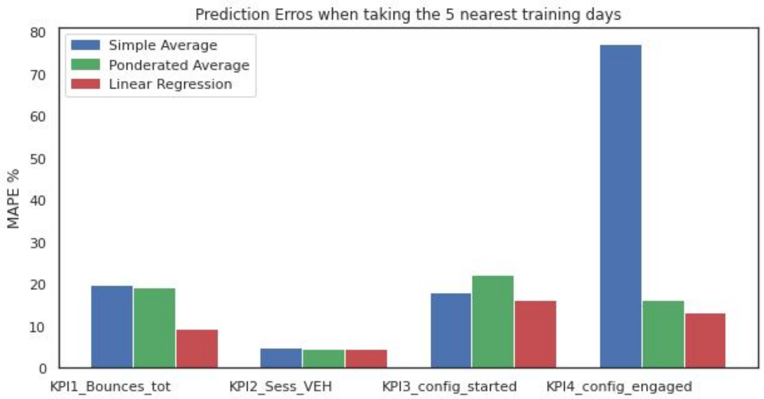
1. **Mean Squared Error** (MSE): average variance between prediction and true value → **More consistent**

2. **Mean Absolute Percentage Error** (MAPE): average percentage of difference from the true value in the sample → **More interpretable**

Model comparison : example

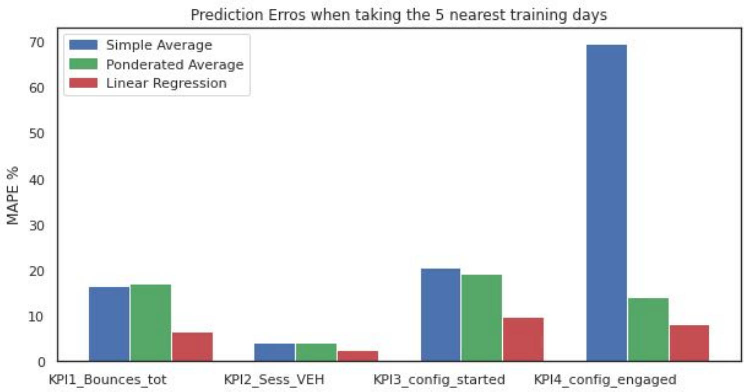
Model A

Distance methodology: **Euclidean distance**
K-Nearest days : **5 days**
Extrapolation methodology : **Test of the 3 methods**



Model B

Distance methodology: **Canberra distance**
K Nearest days : **5 days**
Extrapolation methodology : **Test of the 3 methods**



→ The **Canberra distance** seems to **perform better** in average

→ We can see that it **improves** a lot **the performance of the linear regression model**

Final step and results

Best hyperparameters:

Distance methodology: **Canberra distance**

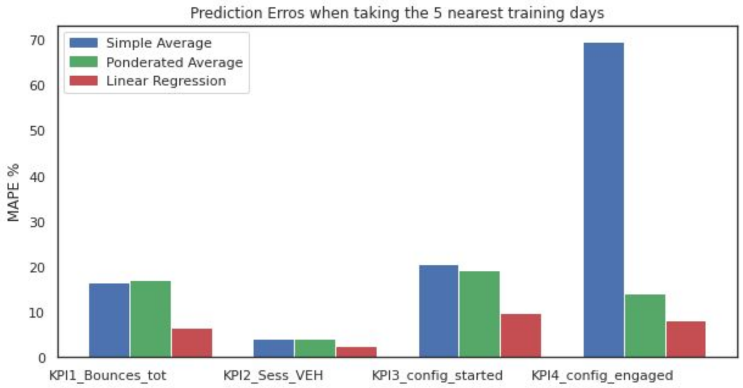
K-Nearest days : **100 days**

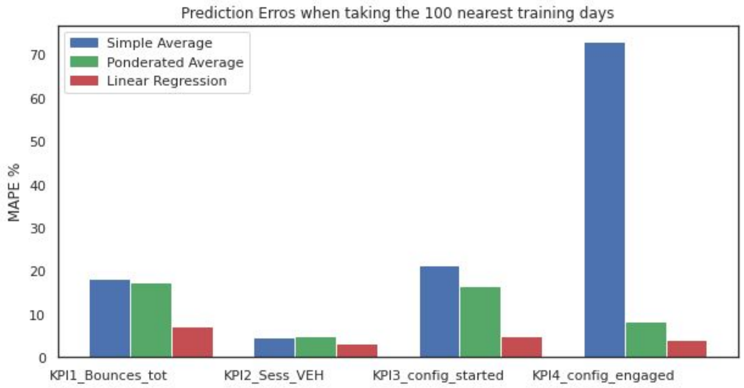
Extrapolation methodology : **linear regression for the 4 KPI's**

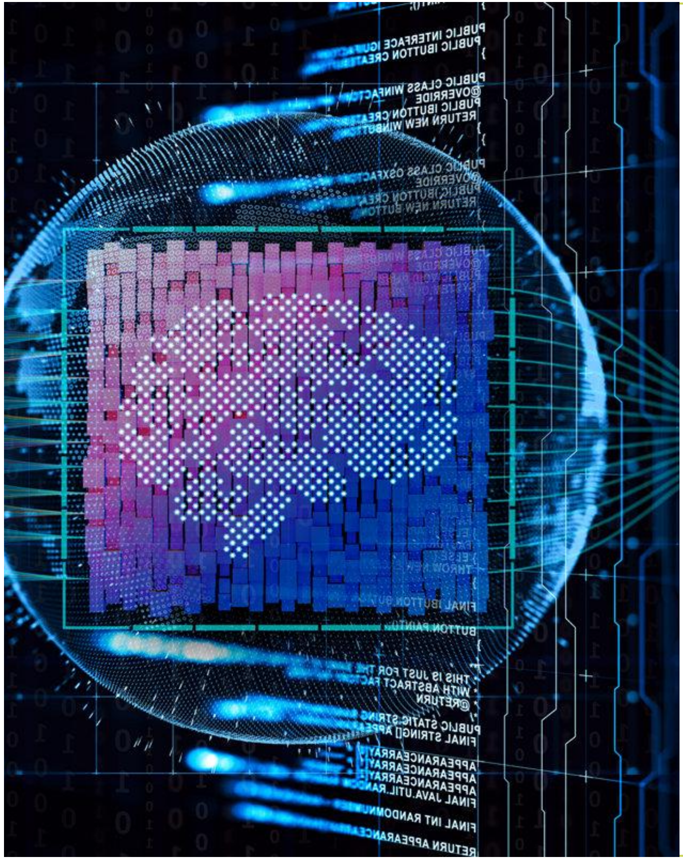
Final MAPE error rates for each KPI's:

- Bounce rate: 7.4%
- Session with vehicule: 2.9%
- Configuration started: 3.7%
- Configuration engaged: 3.4%

Impact of increasing K on performance: a consequent error reduction







05

Recommending final solution

Adopting a 6-step approach

Predicting some specific **insights on the users' activity** of Peugeot's website after **the new version of GDPR implementation** in 2021, while we were no longer able to track the activity on the website from users who refused cookies tracking.

4 KPI's to predict

- The **bounce rate** for the day
- The rate of sessions with **at least one vehicle seen** for the day
- The rate of vehicle **configuration started** during the day
- The rate of vehicle **configuration engaged** during the day.

1. **Cluster the days of** the training sessions
2. **Sampled days** from test sessions
3. **Predict** their **day-cluster**
4. Find their **nearest training days** using the clustering
5. **Extrapolate their KPI's** with those nearest days
6. **Evaluate the predictions** with the true value

Opting for a robust model with linear regression

Model selected :

Distance : **Canberra**
K-nearest days chosen : **100 nearest days**
Extrapolation method : **linear regression**

Error rate for each KPI's:

Bounce Rate: **7%**
>1 vehicule consulted: **2.9%**
Configuration started: **3.7%**
Configuration engaged: **3.4%**

How can we use it ?

Example with August 08th, 2020

KPI	**True Value**	**Predicted Value**	**Error rate**
Bounce Rate	35.4%	37.9%	2.5%
>1 vehicule consulted	49.8%	45.8%	3.0%
Configuration started	7.9%	7.5%	0.4%
Configuration engaged	5.3%	5.4%	0.1%

The predictions are quite accurate.

We can see that the error rate on the configuration started is quite higher than expected for the percentage of session with at least one vehicule consulted, but it is normal as long as the percentage error indicated above are averages.

A methodology to be tuned

Tuning hyperparameters

2 hyperparameters in our modelling approach:

- **K**: number of nearest days to select
- **Distance function**: method to compute how far a day is from another

Possible to find **optimal K** with **iterative approach**, and try other distance computation methods using probabilities

Trying other extrapolation methods

To extrapolate the KPIs, only empirical and simple methods were tried

Other regression techniques with more complexity could be tried (tree-based methods, support vector machines..)

Getting the features' importance

More than 40 variables were used to describe a session.
Some must have more importance across all algorithms used in the methodology, than other features.

Different techniques could be used to **extract the modelling power of each feature**, and **get more transparency**.


Thank you for listening!