DataScientest.com

```
fig, ax = plt.subplots(1, figsize=(10,10))

# plot des data
plt.scatter(df_new['Nombre de vues'], df_new['Nombre

# Legende
legend_elements = [Line2D([0], [0], marker='o', colo
                  markerfacecolor=mcolor, markersize=5

# plot de la legende
plt.legend(handles=legend_elements, loc='upper left

# titre et labels
plt.title("Visualisation des utilisateurs", bbox={'
plt.xlabel('Nombre de vues')
plt.ylabel('Nombre de transactions')
```

Projet PyCommerce

Quentin Lubken • Jérémie Guedj • Cédric Bouré

Bootcamp Data Analyst • Mars 2021

Contexte

Site ecommerce Retailrocket

- Le projet consiste en l'analyse des données du site Ecommerce Retailrocket disponible sur Kaggle (https://www.kaggle.com/retailrocket/ecommerce-dataset), données partiellement anonymisées.

- De l'analyse de données, réaliser un clustering, définir une problématique et modéliser par l'utilisation du machine learning une réponse permettant de résoudre la problématique posée.

- Après une tentative d'utilisation de Google Colab pour travailler en commun nous avons finalement opté pour Jupyter en local qui était plus réactif et sans bugs.

- 5 fichiers composent notre travail : 1 fichier d'exploration des données excel, 1 notebook d'exploration, 1 de visualisation, 1 de clustering, 1 de machine learning

Fichier d’exploration

Premiers constats

Fichier template rapport exploration des données.xlsx

	Nombre lignes dans la table : *à remplir*											
N° Col	**nom de la colonne**	**Type de variable**	**Description**	**Disponibilité de la variable** *a priori*	**Type informatique**	**Taux de NA**	**Gestion des NA**	**Distribution des valeurs**	**Étendue des valeurs**	**Relation entre les valeurs**	**Équilibre**	**Regroupement de valeurs**
			Que représente ce champs en quelques mots ?	*Pouvez vous connaitre ce champ en amont d'une prédiction ? Aurez vous accès à cette variable en environnement de production ?*	*int64, float etc... Si "object", détaillez.*	*en %*	*Quelle mode de (non) - gestion des NA favorisez vous ?*		*Dans le cas des variables catégorielles donnez l'ensemble des catégories possibles. Dans le cas des variables quantitatives détaillez la distribution.*		*Quelle est la répartition des valeurs ? La répartition vous semble-t-elle équilibrée ou trop focalisée autour d'une valeur*	*Pensez vous modifier ou regrouper ou découper les valeurs ? si oui détaillez votre démarche*
1	**item_properties.csv (1 et 2)**											
2	timestamp	explicative	l'heure Unix d'enregistrement de l'évènement	oui	int64	0,00 %		Valeur unique		Ordinale	équilibrée	oui, remettre au format date, heure
3	itemid	explicative	Identifiant produit	oui	int64	0,00 %		Valeur unique		Ordinale	équilibrée	oui, regroupement par somme
4	property	explicative	Propriété produit	oui	object (strings OR numbers)	0,00 %		Catégorielle - sup. à 10 catégories		Ordinale		NSP
5	value	explicative	ID catégorie	oui	object (string & numbers)	0,00 %		Valeur unique		Cardinale		oui, regroupement par somme
6	**category_tree.csv**											
7	categoryid	explicative	catégorie du produit	oui	int64	0,00 %		Catégorielle - sup. à 10 catégories		Ordinale		oui, regroupement par somme
8	parentid	explicative	famille de catégorie produit si existe	oui	float64	1,50 %	valeur 0 (non définissable)	Catégorielle - sup. à 10 catégories		Ordinale		oui, regroupement par somme
9	**events.csv**											
10	timestamp	explicative	l'heure Unix d'enregistrement de l'évènement	oui	int64	0,00 %		Valeur unique		Ordinale		oui, remettre au format date, heure
11	visitorid	explicative	identifiant unique de visiteur	oui	int64	0,00 %		Valeur unique		Cardinale		oui, regroupement par nombre de visites par visiteur, nouveaux vs clients, bcp de croisements possibles
12	event	cible	Type d'événement enregistré	oui	object (strings OR numbers)	0,00 %		Catégorielle - sup. à 10 catégories		Ordinale		pareil que visitorid
13	itemid	explicative	Identifiant produit	oui	int64	0,00 %		Valeur unique		Ordinale		oui, regroupement par somme
14	transactionid	cible	transaction commerciale validée	oui	float64	99,00 %	valeur 0 (non aboutie)	Valeur unique		Ordinale		oui selon visitor id, churn, conversion
15												

- Dossier constitué de 4 fichiers dont 2 à réunir (continuité). Données équilibrées, disponibles et avec peu de valeurs nulles (sauf transactions mais logique).
- 20 275 902 lignes pour les 2 fichiers properties réunis, peu de colonnes et des données en partie anonymisées.
- Variable cible difficile à identifier à la première lecture car dépendra de ce que l’on cherche à prédire.
- Nous nous intéressons rapidement au fichier des événements qui regroupe les données timestamp, identité visiteurs, produits et usages (click, add to cart, transaction) et nous donne un aperçu de données à exploiter.

DataScientest.com

```
fig, ax = plt.subplots(1, figsize=(10,10))

# plot des data
plt.scatter(df_new['Nombre de vues'], df_new['Nombre

# Legende
legend_elements = [Line2D([0], [0], marker='o', colo
                  markerfacecolor=mcolor, markersize=5

# plot de la legende
plt.legend(handles=legend_elements, loc='upper left

# titre et labels
plt.title("Visualisation des utilisateurs", bbox={'
plt.xlabel('Nombre de vues')
plt.ylabel('Nombre de transactions')
```

Exploration

Bootcamp Data Analyst • Mars 2021

Notebook exploration

Fichier 1-Projet Ecommerce Exploration.ipynb

Constats

	timestamp	itemid	property	value
0	1435460400000	460429	categoryid	1338
1	1441508400000	206783	888	1116713 960601 n277.200
2	1439089200000	395014	400	n552.000 639502 n720.000 424566
3	1431226800000	59481	790	n15360.000
4	1431831600000	156781	917	828513

- On réunit les 2 fichiers item_properties_part1.csv et item_properties_part2.csv en un tableau properties aux colonnes timestamp (int), itemid (int), property (object) et value (object). On suppose property et value sont des données aléatoires, il sera donc difficile de les relier à des articles, nous décidons de ne pas exploiter ce fichier ultérieurement.

	categoryid	parentid
0	1016	213.0
1	809	169.0
2	570	9.0
3	1691	885.0
4	536	1691.0

- le fichier category_tree.csv présente 2 colonnes categoryid (int) et parentid (float) aux données anonymisées. On suppose qu'il s'agit de données aléatoires, il sera donc difficile de les relier à des articles, nous décidons également de ne pas exploiter ce fichier.

Notebook exploration

Fichier 1-Projet Ecommerce Exploration.ipynb

Constats

- Le fichier events.csv dispose des colonnes timestamp (int), visitorid (int), event (object), itemid (int), transactionid (int). Avec 2 756 101 lignes et 2 733 644 valeurs nulles dans transactionid, nous avons un nombre de transactions très bas (99% de valeurs nulles).

	timestamp	visitorid	event	itemid	transactionid
0	1433221332117	257597	view	355908	NaN
1	1433224214164	992329	view	248676	NaN
2	1433221999827	111016	view	318965	NaN
3	1433221955914	483717	view	253185	NaN
4	1433221337106	951259	view	367447	NaN

- 1 407 580 utilisateurs uniques, 235 061 produits, 17 672 transactions. 460 lignes sont dupliquées.

- 2 664 218 produits ont été consultés, 68 966 produits ont été ajoutés au panier, 22 457 produits ont été vendus.

Notebook exploration

Fichier 1-Projet Ecommerce Exploration.ipynb

Constats

- Après conversion du timestamp en dates lisibles, on s’aperçoit que le Timedelta est de 4,5 mois (02/05/2015 -> 17/09/2015).
- En utilisant .count(), .unique(), .value_counts() on constate un déséquilibre, le visitorid 1150086 est venu 7757 fois, le visitorid 530559 est venu 4328 fois, suspicion de bots.
- Pour identifier s'il y a des bots on définie des bornes arbitrairement et on classe selon ce qu'on estime être des fréquences d'utilisateurs faibles à excessives. Au-dessus de 3 000, on classe ces visitorid dans bots et on ne les exploitent pas dans les calculs suivants.
- Après plusieurs calculs sur les produits les plus populaires et les moins populaires, on souhaite réaliser l'étude statistique d'une variable cible (que l’on définit étant events_transaction) avec d'autres variables quantitatives afin d'analyser la corrélation inter-variables.

Notebook exploration

Fichier 1-Projet Ecommerce Exploration.ipynb

Constats

- Avec une p-value de 0 et un coefficient > 5%, on observe une forte corrélation entre les variables "events_transaction" et "User frequency", ce qui semble logique compte tenu des variables considérées :

	result
pearson_coeff	0.080954
p-value	0.000000

- * Si on réalise la même étude variable cible (events_transaction) et "Product frequency", on observe une corrélation beaucoup moins marquée car le coefficient est supérieur à 1 et la p-value s'approche de 5%. On peut tout de même considérer qu'il y a corrélation :

	result
pearson_coeff	1.623410e-02
p-value	5.617452e-160

Notebook exploration

Fichier 1-Projet Ecommerce Exploration.ipynb

Constats

- La fonction crosstab nous aide à identifier la corrélation entre deux variables quantitatives. On créer ici la table de contingence entre la variable cible events_transaction et User category :

User category	Infrequent user	Recurrent user	Big user	Bots
events_transaction				
0	2200007	326965	192297	13915
1	9268	5740	6255	1194

- Puis on réalise un test de proportions entre l'effectif de la cellule et l'effectif total de la colonne, le $\chi 2$ sur la table de contingence déterminée précédemment (variable cible events_transaction et day of week) renvoi la valeur 30471.79

- On réalise un dernier test V Cramer renvoyant la valeur 0.105

Notebook exploration

Fichier 1-Projet Ecommerce Exploration.ipynb

Constats

- Le V de Cramer donne un résultat relativement proche de 0, donc il existe une corrélation entre les variables events_transaction et day_of_week. Si on regarde la table de contingence on remarque que les jours de semaine il y a plus de ventes que le Week-end. Le jour avec le nombre de transactions le plus élevé est le mercredi.

- On réalise les mêmes manipulations statistiques avec d'autres variables et la variable cible où le V de Cramer donne également un résultat relativement proche de 0 (0.0196). donc il existe une corrélation entre les variables events_transaction et Product category.

- La table de contingence nous indique que les produits qui génèrent le plus d'événements ne sont pas ceux qui se vendent le plus. Ce sont les produits de la catégorie 'Very popular products' qui ont le pourcentage d'événements de vente le plus important.

- On créer un nouveau dataframe qui va être utilisé pour le clustering et qui marque pour chaque visitorid le nombre de fois où il a un événement de type nombre de vies, nombre d'ajouts au panier et de transactions.

DataScientest.com

```
fig, ax = plt.subplots(1, figsize=(10,10))

# plot des data
plt.scatter(df_new['Nombre de vues'], df_new['Nombr

# Legende
legend_elements = [Line2D([0], [0], marker='o', col
                  markerfacecolor=mcolor, markersize=5

# plot de la legende
plt.legend(handles=legend_elements, loc='upper left

# titre et labels
plt.title("Visualisation des utilisateurs", bbox={'
plt.xlabel('Nombre de vues')
plt.ylabel('Nombre de transactions')
```

Visualisation

Bootcamp Data Analyst • Mars 2021

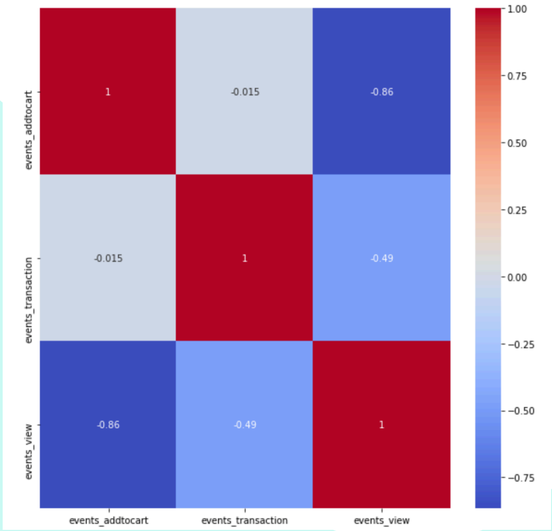
Notebook visualisation

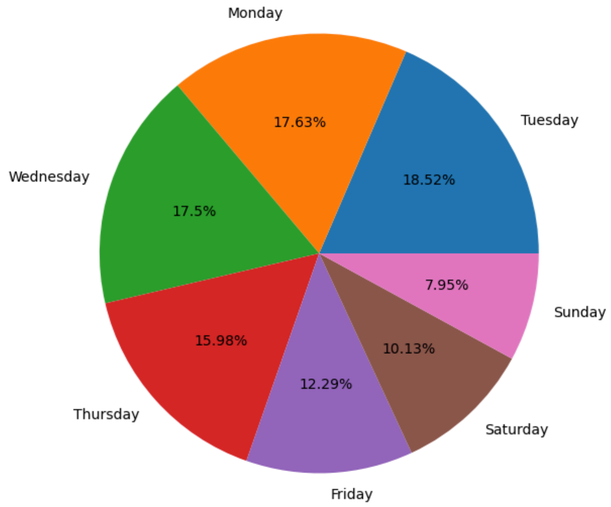
Fichier 2-Projet Ecommerce Visualisation.ipynb

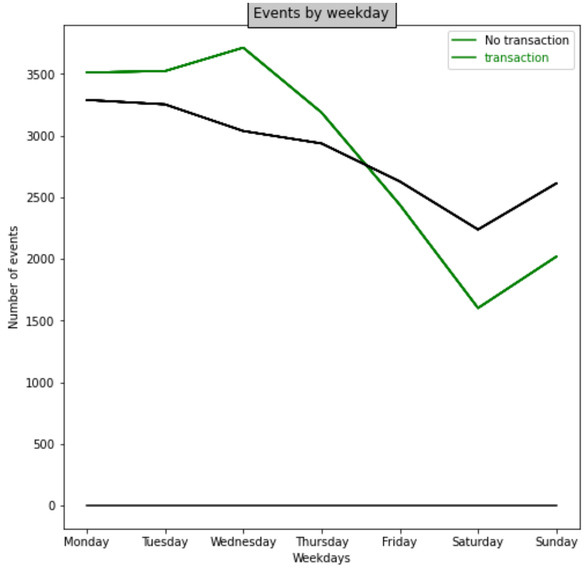
Constats

- On observe une forte corrélation négative entre les différentes valeurs des variables sur df_events et une répartition relativement homogène des transactions sur les jours de semaine avec une baisse significative le week-end.

- On réalise un sampling aléatoire de 20.000 lignes avec et sans achat pour visualiser la tendance et représentation graphique (fichier trop volumineux sinon) où on observe une confirmation de la tendance baissière pour la fin de semaine avec une légère hausse le dimanche. Transactions plus importantes en début de semaine (pic le mercredi).





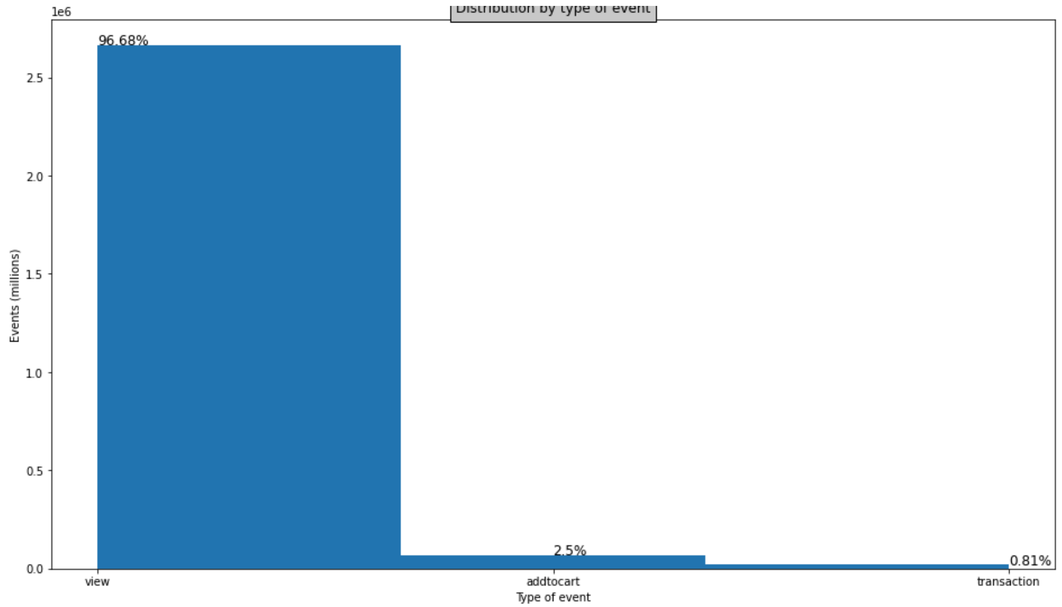


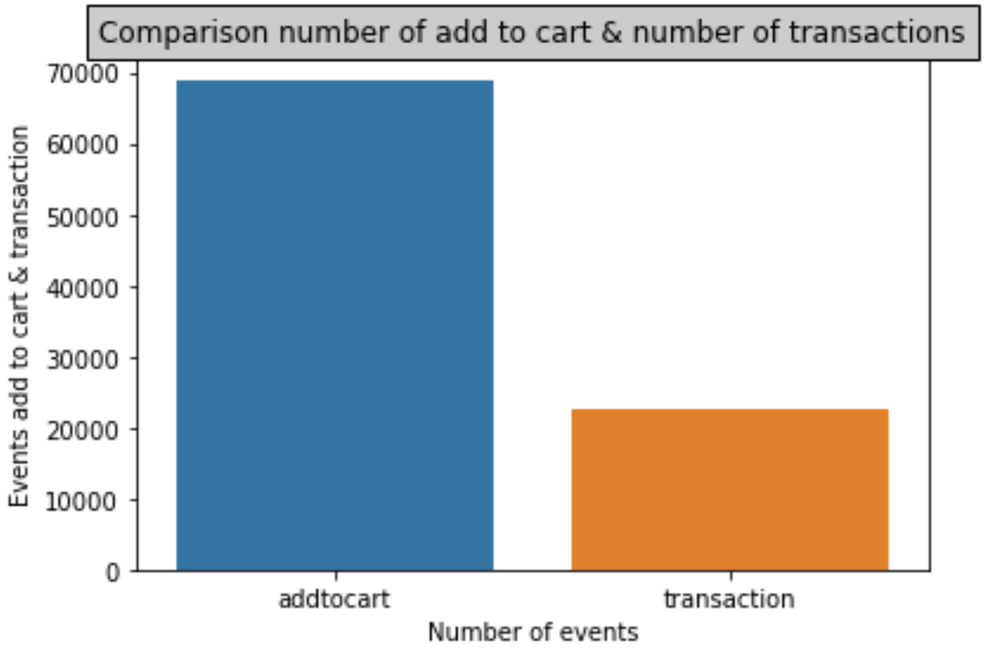
Notebook visualisation

Fichier 2-Projet Ecommerce Visualisation.ipynb

Constats

- On sépare le nombre d'événements pour visualiser le pourcentage de vues, de mises au panier et d’achats et on resserre pour ne comparer que le nombre d'ajouts au panier et de transactions dans df_events. On constate un taux d'abandon du panier 67,6 % , c'est un taux très élevé.



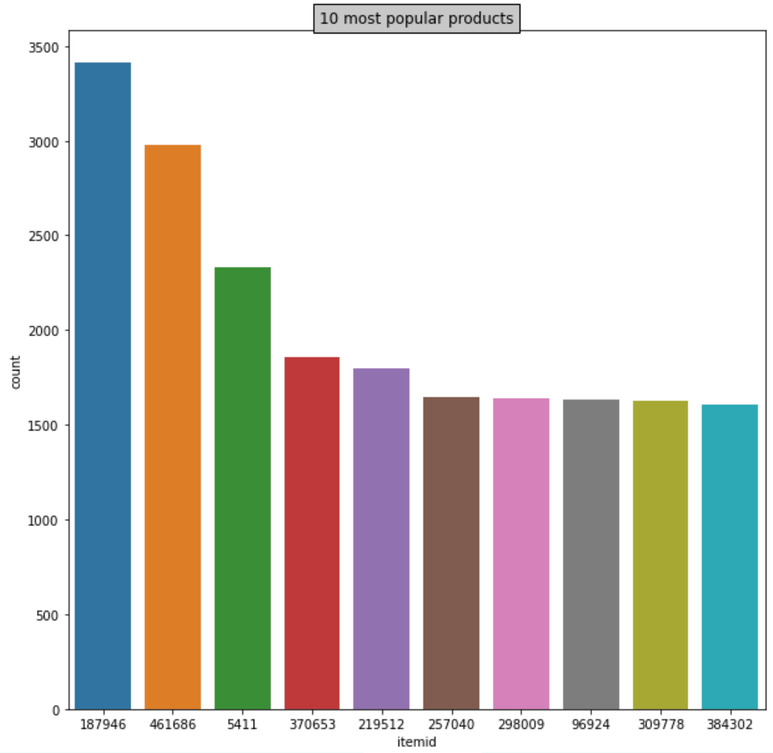


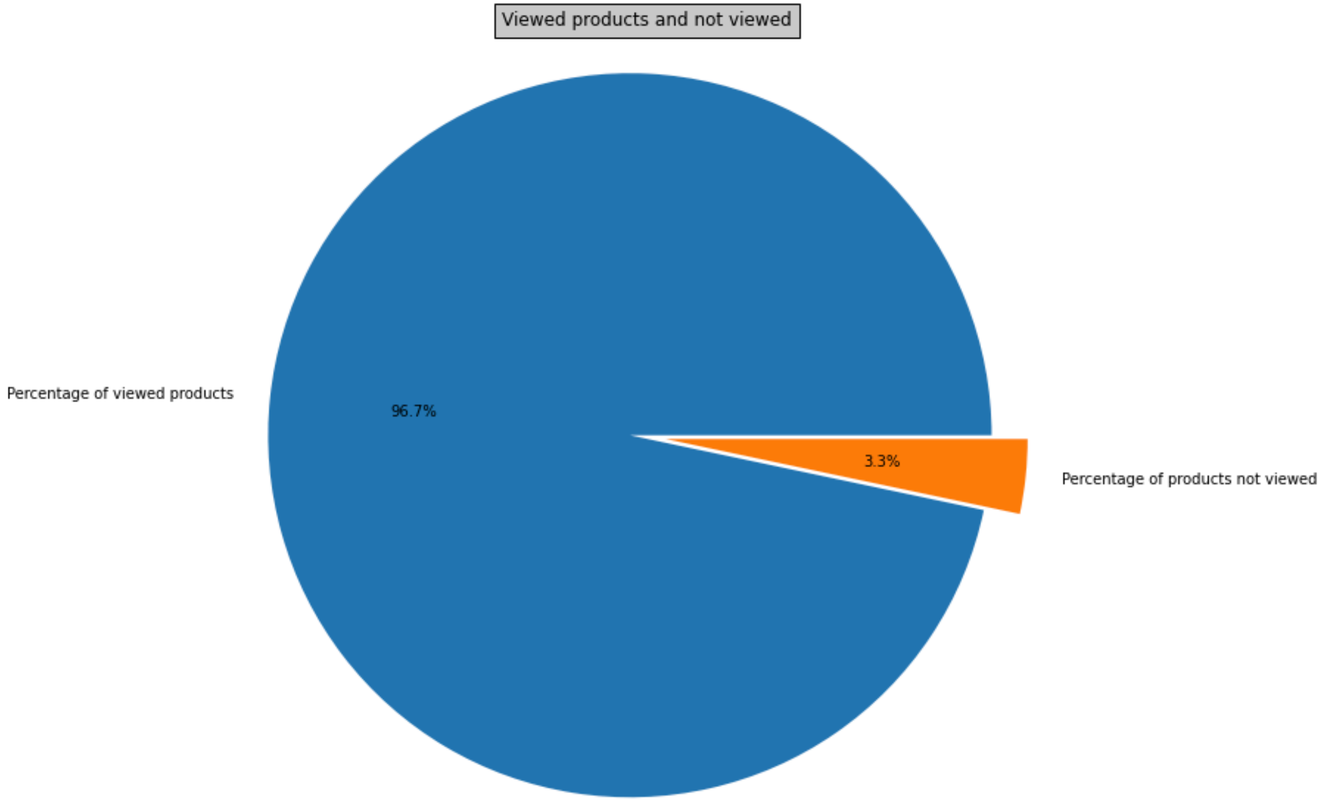
Notebook visualisation

Fichier 2-Projet Ecommerce Visualisation.ipynb

Constats

- On visualise ensuite les 10 produits les plus populaires puis le pourcentage de produits vus et non-vus où 96,7 % des produits ont été vus et seulement 3,3 % ne le sont pas.



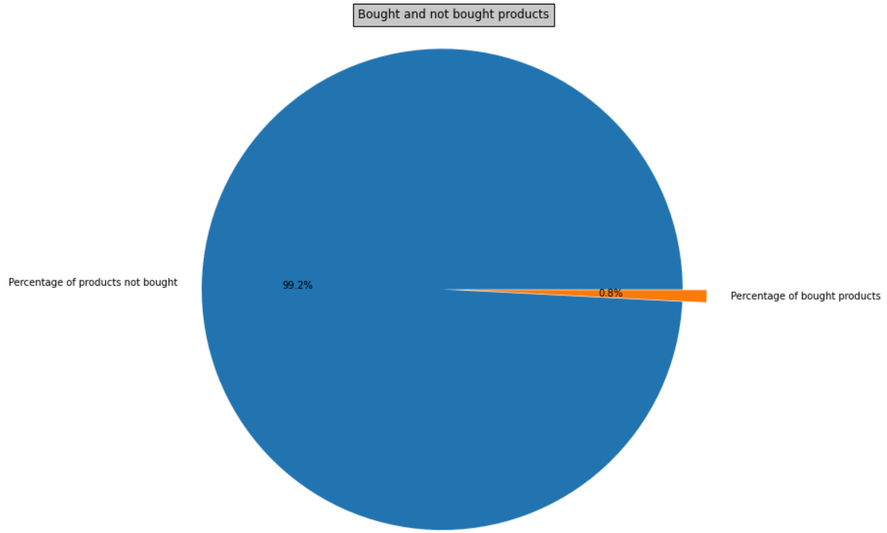


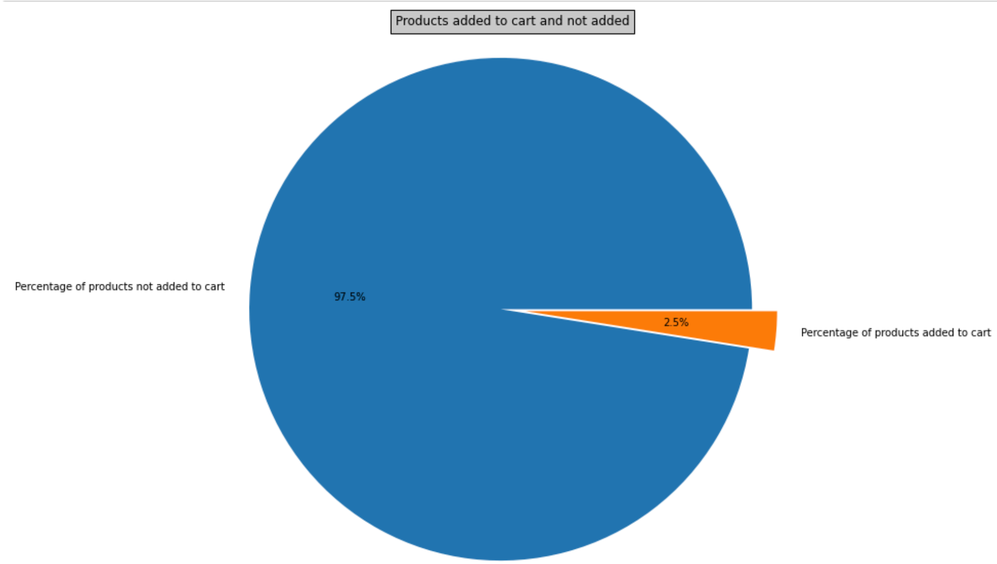
Notebook visualisation

Fichier 2-Projet Ecommerce Visualisation.ipynb

Constats

- On visualise enfin le nombre de produits achetés et non-achetés (99,2 % des produits ne sont pas achetés et seulement 0,8 % le sont, un taux de transformation très faible) et le nombre de produits ajoutés au panier et non-ajoutés (97,5 % des produits ne sont pas ajoutés au panier et seulement 2,5 % le sont, score très faible).





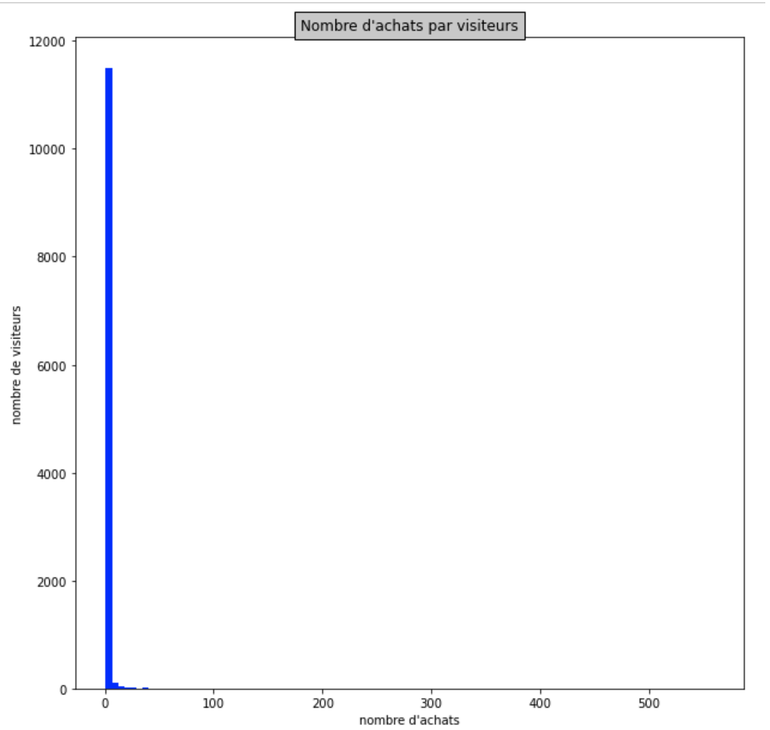
Notebook visualisation

Fichier 2-Projet Ecommerce Visualisation.ipynb

Constats

- On souhaite connaître la répartition du nombre d'achats par visiteur ainsi que les valeurs correspondantes.

- Le graphique nous indique bien que le nombre d'achats par visiteur est anormalement élevé et un rapide calcul nous le confirme, où par exemple le visiteur 1150086 aurait donc fait 4 achats en moyenne par jour ce qui semble très douteux. 99,8% des personnes font moins de 100 achats. On peut considérer que tout le reste sont des outliers.



Notebook visualisation

Fichier 2-Projet Ecommerce Visualisation.ipynb

Constats

- En conclusion du premier travail exploratoire, nous avons un grand nombre de lignes à analyser et peu de colonnes sur ce dataset.
- Le taux d'abandon du panier est trop élevé (67,6%) et le taux de transformation global très faible (0,8%).
- On détecte la présence de bots par l'anormalité de la popularité d'items ayant déclenché un événement, il faut explorer le fichier différement pour comprendre ou exclure les hypothèses autres.
- La répartition des ventes dans la semaine est homogène avec une forte baisse le week-end.
- Les valeurs sont peu corrélées entre-elles, celà est dû à des ID attribués aléatoirement et aux données modifiées sur le dataset.

DataScientest • com

```
from matplotlib.lines import Line2D
fig, ax = plt.subplots(1, figsize=(10,10))

# plot des data
plt.scatter(df_new['Nombre de vues'], df_new['Nombre

# Legende
legend_elements = [Line2D([0], [0], marker='o', colo
                          markerfacecolor=mcolor, markersize=5

# plot de la legende
plt.legend(handles=legend_elements, loc='upper left

# titre et labels
plt.title("Visualisation des utilisateurs", bbox={'
plt.xlabel('Nombre de vues')
plt.ylabel('Nombre de transactions')
```

Clustering

Bootcamp Data Analyst • Mars 2021

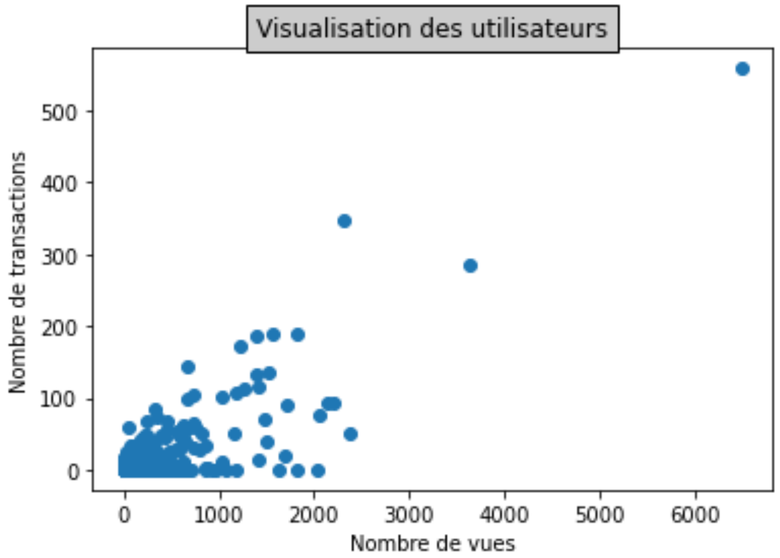
Notebook clustering

Fichier 3-Projet Ecommerce Clustering.ipynb

Constats

- Toujours en se basant sur le dataset events, et en préalable au travail de modélisation, il est nécessaire de séparer en clusters nos utilisateurs selon le nombre et le type d'actions et ainsi repérer et isoler les utilisateurs anormaux.

- Après un essai de clustering hiérarchique non convaincant, on réalise un clustering avec K means où on cherche à visualiser les données où on observe que la majorité des utilisateurs ont consulté et acheté des produits de manière raisonable, cependant certains d'entre eux ont réalisé plus d'une centaine d'achats en 4 mois et ont consulté le site de manière trop fréquente pour un utilisateur lambda.



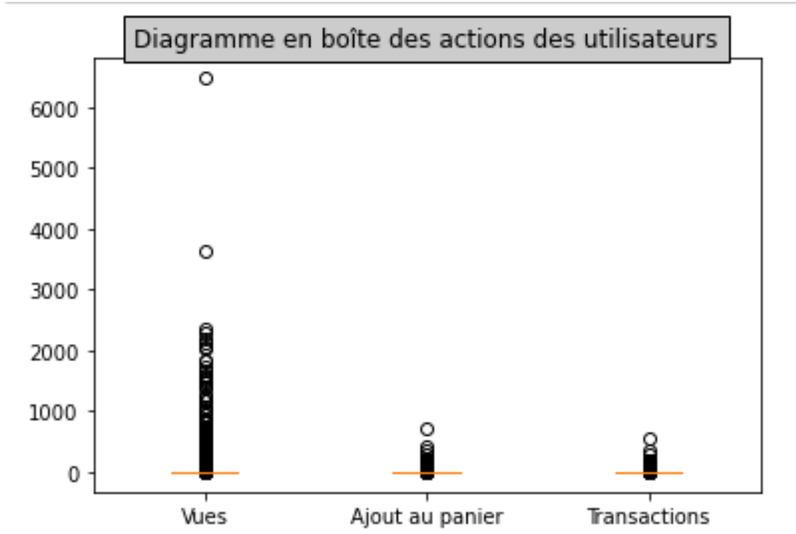
Notebook clustering

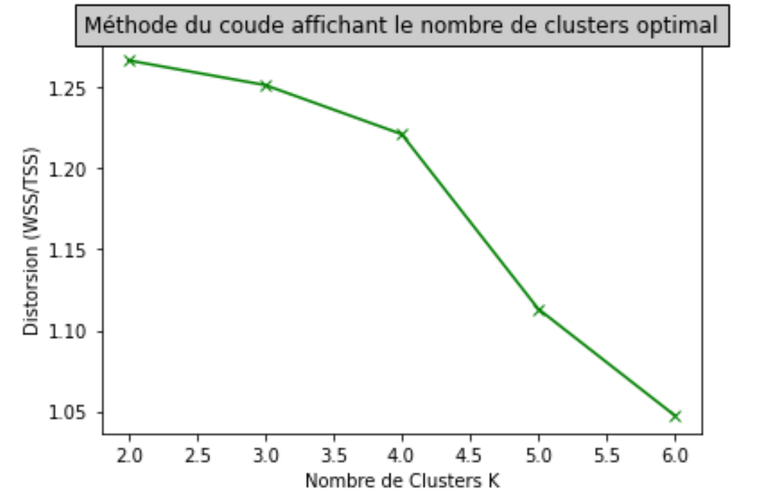
Fichier 3-Projet Ecommerce Clustering.ipynb

Constats

- Un boxplot sur les variables explicatives nous indique que des utilisateurs ont des statistiques de visualisation ou d'achat qui sortent de l’ordinaire.

- On réalise alors la méthode du coude afin de déterminer le juste nombre de clusters nécessaire puis on visualise les distorsions en fonction du nombre de clusters. 4 clusters seront définis pour la suite des travaux.



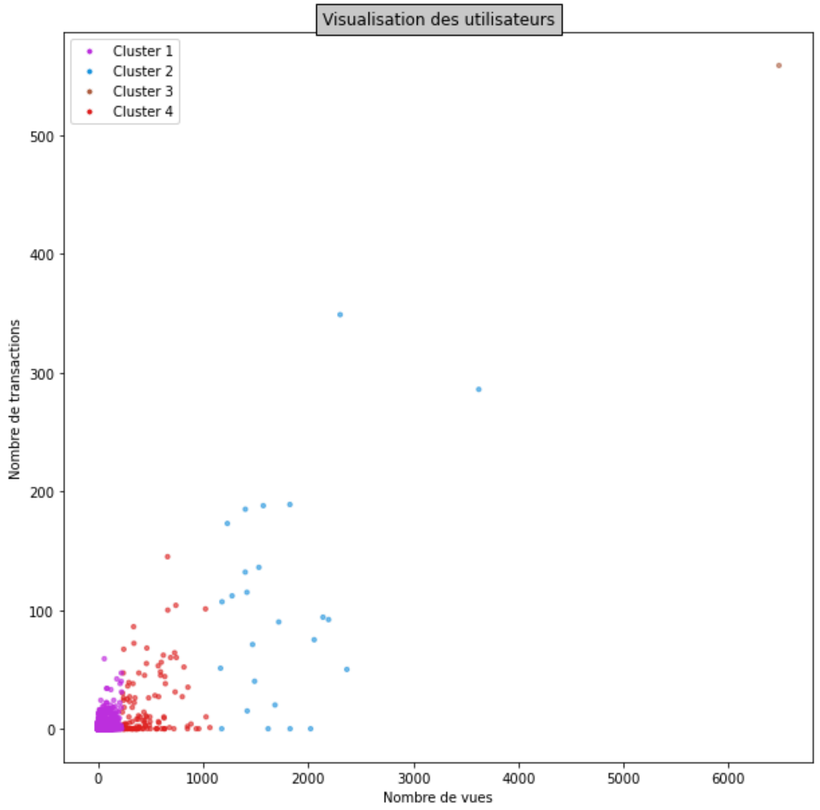


Notebook clustering

Fichier 3-Projet Ecommerce Clustering.ipynb

Constats

- On met ensuite en place l'algorithme des K-means avec les 4 clusters et on visualise.
- Grâce au clustering on observe 4 clusters distincts qui nous permettent d'isoler les utilisateurs ayant un comportement anormal.
- On peut émettre alors plusieurs hypothèses sur les utilisateurs présents dans le 3ème et 4ème clusters :
  - Il y a un bug et plusieurs clients utilisent le site avec le même i uilisateur.
  - Certains utilisateurs sont des clients b2b et font de très grosse commandes sur le site.
  - Certains utilisateurs sont en réalité des bots.



Notebook clustering

Fichier 3-Projet Ecommerce Clustering.ipynb

Constats

- La suite du travail de clustering a pour objectif de créer des dataframes qui montrent les produits achetés, ajoutés au panier ou vus par les utilisateurs groupés par cluster.
- Pour commencer on crée un dataframe qui reprend certaines données de df_id : le cluster de chaque utilisateur avec tous les évènements qui le concernent.
- On observe une grande disproportion de visiteurs entre les clusters. Les clusters 1, 2 et 3 présentent des utilisateurs aux comportements extrêmes et qu'il faudra traiter séparement (traitement business, recherche d'un bug technique…étude à approfondir).
- Un calcul nous indique que :
  - 3 401 visiteurs, tous clusters confondus, sont venus sur le site sans regarder un seul item,
  - 1 369 858 visiteurs, tous clusters confondus, sont venus sur le site sans ajouter au panier un seul item
  - 1 395 861 visiteurs, tous clusters confondus, sont venus sur le site sans acheter un seul item

visitorid	cluster	Nombre de transactions	Nombre d'ajout au panier	Nombre de vues
1150086	2	559.0	719.0	6479.0
530559	1	286.0	419.0	3623.0
895999	1	50.0	56.0	2368.0
152963	1	349.0	371.0	2304.0
163561	1	92.0	124.0	2194.0
...	...	...	...	...
409939	0	1.0	0.0	0.0
684791	0	1.0	0.0	0.0
1246009	0	1.0	0.0	0.0
1214715	0	1.0	0.0	0.0
622280	0	1.0	0.0	0.0

1407580 rows × 4 columns

Notebook clustering

Fichier 3-Projet Ecommerce Clustering.ipynb

Constats

- Au regard du nombre de vues conséquent sur le cluster 0, il sera intéressant de le séparer en sous-clusters pour étudier la population de visiteurs plus finement. Ce qu'on fera ultérieurement.

- On créer un nouveau dataframe qui permet de lier les informations que l'on a sur chaque visiteur selon son cluster d'appartenance mais en y ajoutant l'itemid. Cela nous permet de voir quels produits ont été visités par visiteur.

	timestamp	visitorid	event	itemid	transactionid	day_of_week	events_addtocart	events_transaction	events_view	cluster	Nombre de transactions	Nombre d'ajout au panier	Nombi de vue
0	2015-06-02 00:02:12.117	257597	view	355908	NaN	Tuesday	0	0	1	0	0.0	0.0	2
1	2015-06-08 17:00:21.247	257597	view	302696	NaN	Monday	0	0	1	0	0.0	0.0	2
2	2015-06-02 00:50:14.164	992329	view	248676	NaN	Tuesday	0	0	1	0	0.0	0.0	30
3	2015-06-02 00:57:52.007	992329	view	193150	NaN	Tuesday	0	0	1	0	0.0	0.0	30
4	2015-06-02 01:12:35.976	992329	view	246453	NaN	Tuesday	0	0	1	0	0.0	0.0	30

Notebook clustering

Fichier 3-Projet Ecommerce Clustering.ipynb

Constats

- Nombre total d'événements en fonction des clusters et de chaque produit :

cluster	0	1	2	3	Total
itemid					
187946	3411	0	0	1	3412
461686	2810	61	29	78	2978
5411	2320	4	0	10	2334
370653	1853	0	0	1	1854
219512	1736	30	11	23	1800
...	...	...	...	...	...
356847	1	0	0	0	1
356844	1	0	0	0	1
356843	1	0	0	0	1
79432	1	0	0	0	1
197289	1	0	0	0	1

- Nombre de vues par cluster :

cluster	0	1	2	3	Total
itemid					
187946	3409	0	0	1	3410
461686	2514	58	25	75	2672
5411	2312	4	0	9	2325
370653	1853	0	0	1	1854
219512	1691	28	10	23	1752
...	...	...	...	...	...
154410	1	0	0	0	1
71262	1	0	0	0	1
309388	1	0	0	0	1
286303	1	0	0	0	1
380635	1	0	0	0	1

- Nombre d'ajouts au panier par cluster :

cluster	0	1	2	3	Total
itemid					
461686	421	6	6	6	439
312728	172	4	0	32	208
409804	166	10	2	13	191
320130	151	8	2	13	174
29196	123	0	2	30	155
...	...	...	...	...	...
284252	1	0	0	0	1
284259	1	0	0	0	1
284265	1	0	0	0	1
284290	1	0	0	0	1
65284	1	0	0	0	1

- Nombre de transactions par cluster :

cluster	0	1	2	3	Total
itemid					
461686	125	3	2	3	133
119736	14	45	1	37	97
213834	65	11	2	14	92
312728	38	0	0	8	46
7943	41	0	1	4	46
...	...	...	...	...	...
190034	1	0	0	0	1
190095	1	0	0	0	1
190243	1	0	0	0	1
190246	1	0	0	0	1
466861	1	0	0	0	1

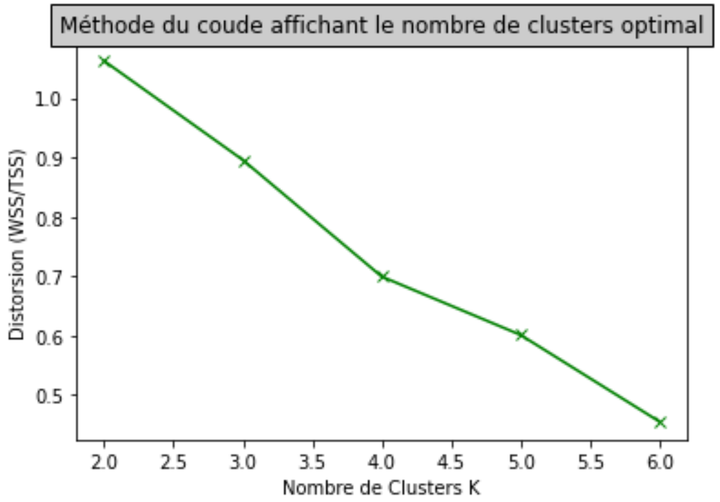
Notebook clustering

Fichier 3-Projet Ecommerce Clustering.ipynb

Constats

- Afin d'augmenter la précision de nos analyses, nous allons nous concentrer sur le cluster 0 qui concentre 99.98% des utilisateurs.
- Pour cela nous créons un nouveau dataframe contenant uniquement les utilisateurs du cluster 0 et on réalise à nouveau la méthode du coude afin de déterminer le juste nombre de clusters nécessaires en visualisant les distorsions en fonction du nombre de clusters

visitorid	Nombre de vues	Nombre d'ajout au panier	Nombre de transactions
317355	228.0	5.0	0.0
1035814	224.0	42.0	31.0
438441	222.0	42.0	47.0
797409	221.0	48.0	40.0
1381830	221.0	6.0	2.0

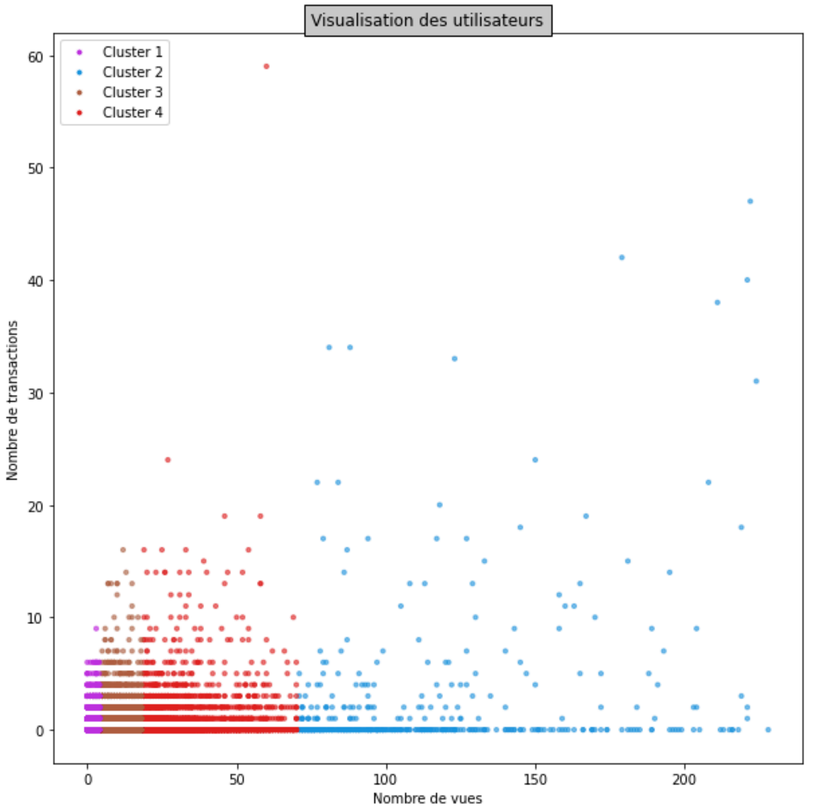


Notebook clustering

Fichier 3-Projet Ecommerce Clustering.ipynb

Constats

- On met ensuite en place l'algorithme des K-means avec les 4 clusters et on visualise les nouveaux clusters créés a partir du cluster 0.
- On observe une distribution beaucoup plus homogène entre les clusters.
- Cette distribution va nous permettre d'établir une segmentation plus précise, et d'augmenter la précision de notre algorithme de machine learning.



DataScientest • com

```
fig, ax = plt.subplots(1, figsize=(10,10))

# plot des data
plt.scatter(df_new['Nombre de vues'], df_new['Nombre

# Legende
legend_elements = [Line2D([0], [0], marker='o', colo
                  markerfacecolor=mcolor, markersize=5

# plot de la legende
plt.legend(handles=legend_elements, loc='upper left

# titre et labels
plt.title("Visualisation des utilisateurs", bbox={'
plt.xlabel('Nombre de vues')
plt.ylabel('Nombre de transactions')
```

Machine learning

Bootcamp Data Analyst • Mars 2021

Notebook ML

Fichier 4-Projet Ecommerce Machine learning.ipynb

Constats

- **Régression linéaire :**
  - La régression linéaire que l'on souhaite réaliser tente de prédire si les visiteurs vont acheter ou non des articles.
  - Nous créons au préalable une fonction qui va créer un dataframe identifiant de visiteur, nombre d'éléments visionnés, nombre total de vues, transactions ou non
  - Puis nous créons un dataframe buying_visitors qui reprend cette fonction pour une vue claire des événements par visiteur et un autre avec le nombre de visiteurs uniques.

	visitorid	num_items_viewed	view_count	bought_count	purchased
0	172	22	33	2	1
1	186	1	2	1	1
2	264	2	3	2	1
3	419	3	4	1	1
4	539	1	4	1	1
...	...	...	...	...	...
11714	1406787	3	20	1	1
11715	1406981	4	4	1	1
11716	1407070	1	1	1	1
11717	1407110	2	7	1	1
11718	1407398	0	0	1	1

11719 rows × 5 columns

	visitorid	num_items_viewed	view_count	bought_count	purchased
0	0	3	3	0	0
1	1	1	1	0	0
2	2	4	8	0	0
3	3	1	1	0	0
4	4	1	1	0	0
...	...	...	...	...	...
27815	28028	1	1	0	0
27816	28029	1	1	0	0
27817	28030	3	3	0	0
27818	28031	1	1	0	0
27819	28032	1	1	0	0

27820 rows × 5 columns

Notebook ML

Fichier 4-Projet Ecommerce Machine learning.ipynb

Constats

- Ces dataframes sont ensuite fusionnés pour préparer la régression linéaire et nous appliquons une régression linéaire sur un ensemble de tet (30%) qui nous indique une précision de notre modèle de **79,29 %**

- **Machine learning :**

  - On repart de la création d'un dataframe qui contient les utilisateurs qui ont réalisé au moins 10 interactions afin d'éliminer les utilisateurs les moins significatifs et nous conservons ceux qui ont réalisé à minima un achat.

  - On créer un autre dataframe qui contient la liste des items achetés par chaque visiteur avec ajout de son cluster d'appartenance

	visitorid	Item acheté	cluster
0	599528	[356475]	0
1	121688	[15335, 380775, 237753, 317178, 12836, 400969,...	0
2	189384	[310791, 299044]	0
3	350566	[54058, 284871, 251130, 268335, 183049, 261940...	1
4	404403	[150100, 50934, 36013, 26210, 118199, 234199, ...	3

Notebook ML

Fichier 4-Projet Ecommerce Machine learning.ipynb

Constats

- On applique l'algorithme **Apriori** afin de créer un moteur de suggestion produits sur la base des produits déjà achetés :

- **L'idée principale d'Apriori est la suivante :**

  - Tous les sous-ensembles non vides d'un ensemble d'éléments fréquents doivent également être fréquents. Il sagit d'une approche ascendante. Nous sommes partis de chaque item de la liste d'items. Ensuite, les candidats sont générés par auto-adhésion. Nous étendons la longueur des itemsets un item à la fois.

  - Le test de sous-ensembles est effectué à chaque étape et les ensembles d'items qui contiennent des sous-ensembles peu fréquents sont élagués. Nous répétons le processus jusqu'à ce qu'il ne soit plus possible de dériver d'itemsets à partir des données.

  - On doit d'abord transformer les données en matrice qui renvoient des valeurs True/False pour être exploitées

Notebook ML

Fichier 4-Projet Ecommerce Machine learning.ipynb

Constats

- Puis on créer un dataset qui regroupe les objets les plus souvent achetés ensemble et leur nombre.
- Dans l'algorithme Apriori, deux eléments sont primordiaux :
- **Le support :** Ici cela represente la probabilité qu'un produit ait été acheté, ou qu'un ensemble de produits ait été acheté. Plus sa valeur est proche de 1 plus ce produit ou cet ensemble est important.
- **La confiance** : Ici cela représente la probabilité conditionnelle qu'une transaction qui contient le produit {A} contienne aussi le produit {B}. Si la confiance est de 1 cela signifie que le produit {A} a toujours été acheté avec le produit {B}.
- Sur un Total de 12025 produits acheté et 22457 transactions, le produit le plus acheté {461686} a été acheté 133 fois. Le seuil de prise en charge (support) que nous allons définir va donc être faible car aucun produit a une très forte influence sur l'ensemble de données.

Notebook ML

Fichier 4-Projet Ecommerce Machine learning.ipynb

Constats

- Le produit qui revient le plus souvent a une probabilité de 1.9525 % d'être acheté.
- On peut aussi observer la faible importance des achats groupés sur l'ensemble des achats.
- On affiche les résultats, avec pour chaque produit acheté (antecedents) une ou plusieurs propositions de produits (consequents) susceptibles d’intéresser les acheteurs.
- Afin d'affiner les résultats, on defini un seuil (min_threshold) que l'on applique a un indicateur donné (ici l'indicateur est la confiance. Le seuil a été défini sur 0.2 donc 20%, cela signifie que l'algorithme va retourner les résultats pour les produits qui ont au minimum déjà été acheté ensemble dans 20% des cas.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(546)	(119736)	0.003166	0.010554	0.001319	0.416667	39.479167	0.001286	1.696193
1	(546)	(248455)	0.003166	0.004222	0.001055	0.333333	78.958333	0.001042	1.493668
2	(248455)	(546)	0.004222	0.003166	0.001055	0.250000	78.958333	0.001042	1.329112
3	(546)	(338660)	0.003166	0.002639	0.001055	0.333333	126.333333	0.001047	1.496042
4	(338660)	(546)	0.002639	0.003166	0.001055	0.400000	126.333333	0.001047	1.661390
...	...	...	...	...	...	...	...	...	...
141	(384302)	(119736, 338660)	0.002111	0.001319	0.001055	0.500000	379.000000	0.001053	1.997361
142	(213834, 268883)	(445351)	0.001055	0.005805	0.001055	1.000000	172.272727	0.001049	inf
143	(213834, 445351)	(268883)	0.005013	0.004222	0.001055	0.210526	49.868421	0.001034	1.261319
144	(268883, 445351)	(213834)	0.001055	0.011082	0.001055	1.000000	90.238095	0.001044	inf
145	(268883)	(213834, 445351)	0.004222	0.005013	0.001055	0.250000	49.868421	0.001034	1.326649

Notebook ML

Fichier 4-Projet Ecommerce Machine learning.ipynb

Constats

- Les colonnes antecedents et consequents contiennent les Itemid antecedent (le produit d'entrée, qui conduit à la recommandation) et consequent (le produit recommandé en fonction de l'antecedent)
- Il existe **3 indicateurs** de supports : l'indicateur de l'antecedent, du consequent ainsi que du couple antecedent/ consequent.
- La **confidence** représente la probabilité de voir le consequent dans une transaction donne qui contient également l'antecedent. Par exemple le produit (119736) a une probabilité de 41% d'etre acheté si le produit (546) a deja été acheté par cet utilisateur.
- Le **lift** est un indicateur utilisé pour mesurer combien de fois l'antecedent et le consequent seraient achetés ensemble si ils étaient statistiquement indépendants. Si {A} et {C} sont indépendants, le score lift sera exactement 1.
- Le **leverage** est un indicateur qui calcule la différence entre la fréquence observée de {A} et {C} apparaissant ensemble et la fréquence attendue si A et C sont indépendants. Une valeur de 0 indique l'indépendance.
- La **conviction** est un indicateur qui mesure la dépendance du consequent a l'antecedent. Une valeur de conviction élevée signifie que le consequent dépend fortement de l'antecedent. Par exemple, dans le cas d'un score de confiance parfait, le dénominateur devient 0 (en raison de 1 - 1) pour lequel le score de conviction est défini comme «inf». Comme le lift, si les éléments sont indépendants, la conviction est de 1.

DataScientest.com

```
fig, ax = plt.subplots(1, figsize=(10,10))

# plot des data
plt.scatter(df_new['Nombre de vues'], df_new['Nombre

# Legende
legend_elements = [Line2D([0], [0], marker='o', col
                  markerfacecolor=mcolor, markersize=5

# plot de la legende
plt.legend(handles=legend_elements, loc='upper left

# titre et labels
plt.title("Visualisation des utilisateurs", bbox={'
plt.xlabel('Nombre de vues')
plt.ylabel('Nombre de transactions')
```

Axes d'amélioration et conclusion

Bootcamp Data Analyst • Mars 2021

Conclusion

Site ecommerce Retailrocket

- Le moteur de recommandation donne des résultats intéressants et directement exploitables sur la base des transactions réalisées.
- Ce modèle est parfaitement transposable à d'autres événements et donc à d'autres visiteurs qui visualisent ou mettent au panier un ou plusieurs items.
- Peuvent être également considérées comme variables sur un même modèle transposable à la segmentation par cluster, également au timestamp qui permettrait de suggérer des produits selon l'heure de la journée (si on considère des produits alimentaires, midi ou 19h peuvent être des bons moments pour donner envie d'achat à des heures propices à l'appétit) mais d'autres variables peuvent être considérées dans le calcul selon la stratégie que veut développer l'entreprise.
- Il est donc intéressant d'intégrer que ce modèle de suggestion est duplicable et facilement modifiable.

Axes d’amélioration

Site ecommerce Retailrocket

- Parce que nous avons à notre disposition un grand nombre de données anonymisées, il est impossible de les remettre dans leur contexte (parle t-on de produits de grande consommation ? B2B, B2C ? Produits physiques ou digitaux…). Nous avons donc latitude à nous poser des tas de questions sans pour autant savoir si cela est pertinent en terme d’analyse.

- Notre phase d’exploration était donc conséquente par manque de contextualisation. Si nous avions eu à notre disposition des indications nous permettant de recontextualiser le site ecommerce cela nous aurait permis d’être plus précis quant à l’analyse et le moteur de recommandation même si celui-ci est, et c’est pour cela que nous l’avons choisi, assez simple à remettre dans un contexte plus détaillé.

- On aurait souhaité réaliser un calcul complémentaire en ajoutant les clusters de chaque utilisateur comme variable supplémentaire de notre outil de recommandation mais la solution technique ne nous a pas apparu évidente pour réaliser des calculs pertinents.