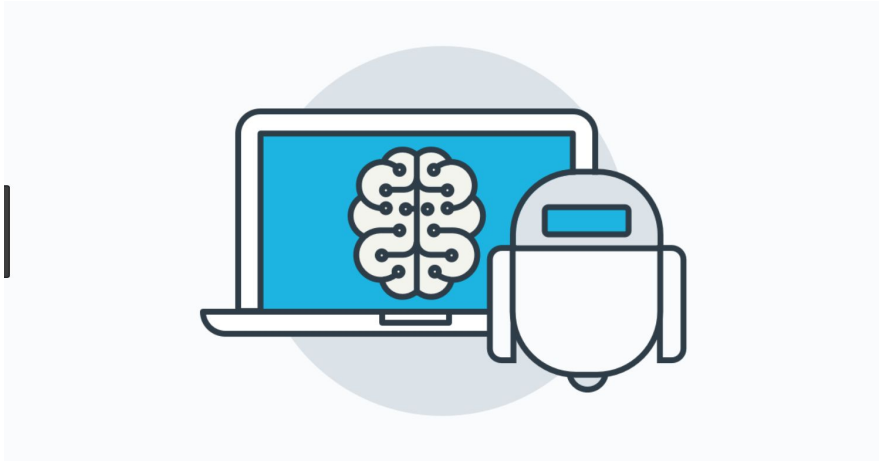
Advance Machine Learning

Modélisation des revenus

Denis OBLIN
Takwa ALDROE | Wissal BENJIRA | Nacima BEN SOUNA | Fajer YOUSAF
A5 - DIA2

SOMMAIRE

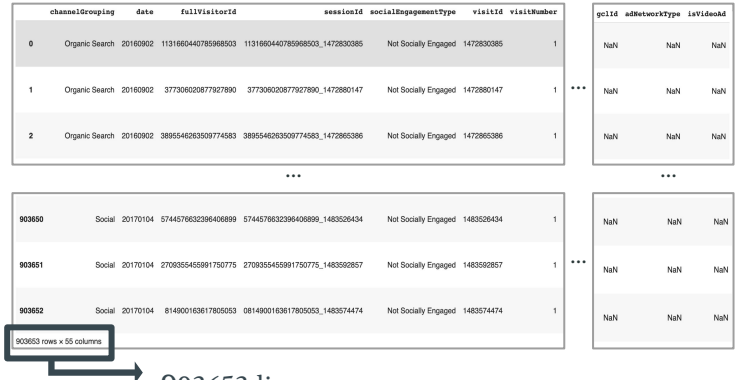


Introduction

Dans le cadre du projet du module *Advance machine Learning*, il nous a été mis à disposition un jeu de données répertoriant des transactions d'achat. L' objectif de ce projet est de modéliser et d'étudier les revenues générés par des personnes.

Pour ce faire nous avons définis une approche de régression et testé plusieurs algorithmes et modèles. Pour chacun d'entre eux nous avons apprécié leur performances, afin de conclure sur le meilleure modèle nous permettant de prédire au mieux la target : *transactionRevenue*, par personne.

Présentation du dataset



	channelGrouping	date	fullVisitorId	sessionId	socialEngagementType	visitId	visitNumber	gclId	adNetworkType	isVideoAd
0	Organic Search	20160902	1131660440785968503	1131660440785968503_1472830385	Not Socially Engaged	1472830385	1	NaN	NaN	NaN
1	Organic Search	20160902	3773060208779278900	3773060208779278900_1472880147	Not Socially Engaged	1472880147	1	NaN	NaN	NaN
2	Organic Search	20160902	3895546263509774583	3895546263509774583_1472865386	Not Socially Engaged	1472865386	1	NaN	NaN	NaN
...										
903650	Social	20170104	5744576632396406899	5744576632396406899_1483526434	Not Socially Engaged	1483526434	1	NaN	NaN	NaN
903651	Social	20170104	2709355455991750775	2709355455991750775_1483592857	Not Socially Engaged	1483592857	1	NaN	NaN	NaN
903652	Social	20170104	8149001636178050530	8149001636178050530_1483574474	Not Socially Engaged	1483574474	1	NaN	NaN	NaN

903653 rows × 55 columns

903653 lignes
55 colonnes

Nous disposons d'un jeu de données qui contient 903653 lignes et 55 colonnes dont 12 variables quantitatives et 43 qualitatives :

Quantitative: date, visitId, visitNumber, visitStartTime, visits, hits, pageviews, bounces, newVisits, isMobile,page, transactionRevenue

Qualitative: sessionId, socialEngagementType, datasplit, campaign, source, medium, keyword, isTrueDirect, referralPath ...

On souhaite prédire la variable "transactionRevenue". Ainsi, le but est de choisir le meilleur modèle de prédiction pas forcément avec les meilleures performances mais plutôt avec une approche structurée et réfléchie pour résoudre le problème.

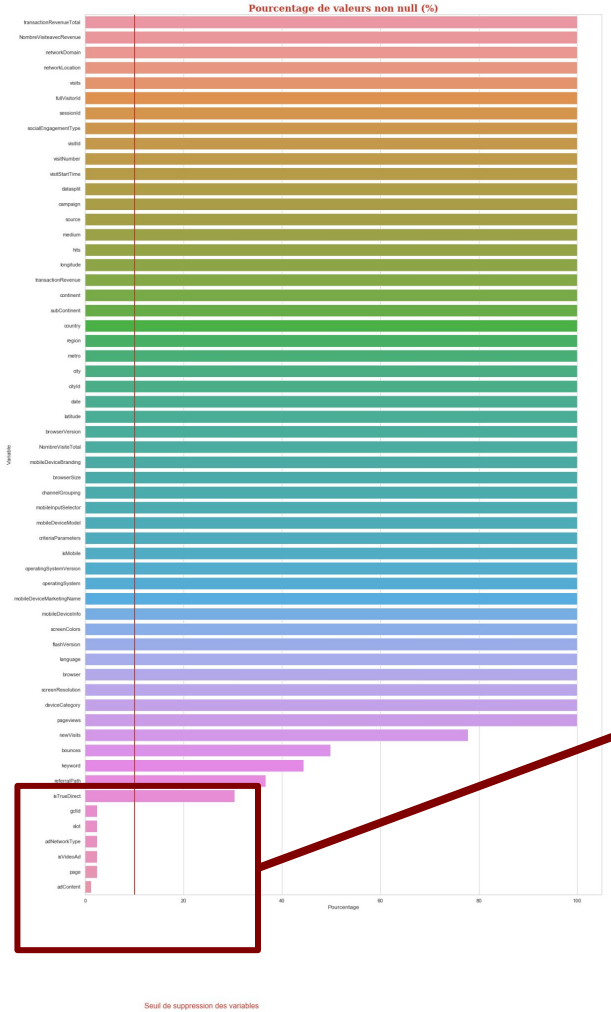
Il s'agit ici d'un problème de régression avec comme métrique R^2.

Présentation du dataset

Cette étude de cas est basée sur la règle 80/20 qui stipule que pour de nombreux événements, environ 80% des effets proviennent de 20% des causes. Sur la base de la règle 80/20, cette étude de cas vise à trouver ces 20% de visiteurs qui représentent 80% des transactions. Ce qui explique le grand nombre de valeurs nulles pour la variable dans notre dataset.

Par conséquent, nous allons examiner le nombre d'instances où le revenu n'est pas nul. On remarque que le dataset présente plusieurs lignes associées à un même utilisateur. Nous nous intéressons donc aux visiteurs uniques dans l'ensemble de formation.

Pour ce faire, nous avons donc créé une variable *transactionRevenueTotal* qui donne pour chaque utilisateur la somme de toutes ses transactions (qui peuvent être nulles).



Data pre-processing

Valeurs manquantes :

Nous nous sommes ensuite intéressé aux valeurs manquantes de chacune des variables de notre jeu de donnée.

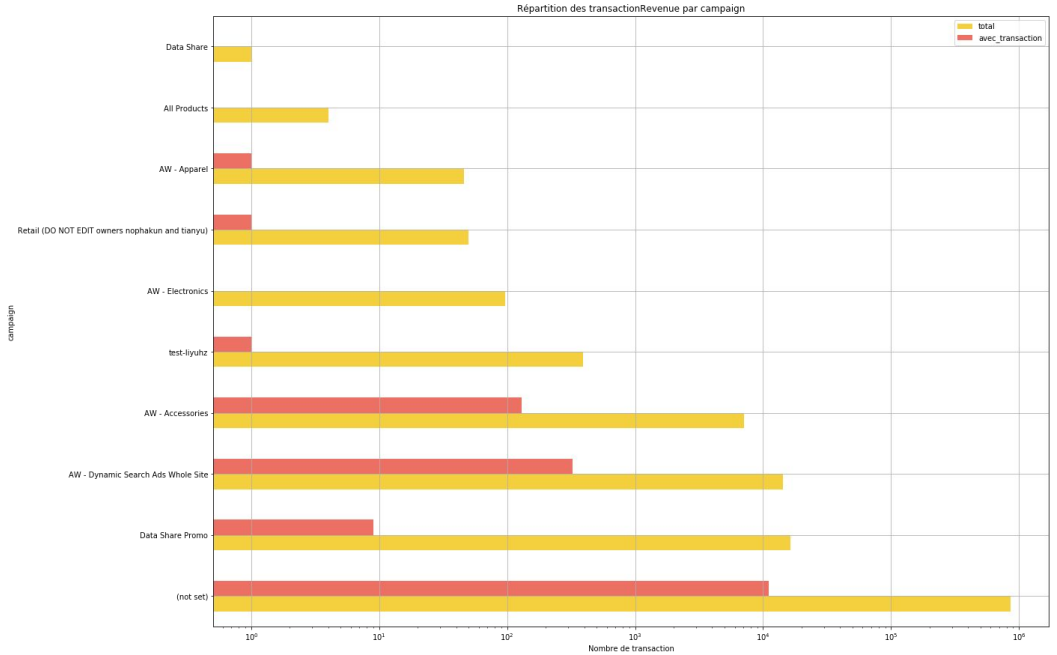
Ainsi, 11 variables ont des valeurs manquantes. Nous avons fixé un seuil de suppression de variable de 5‰ pour filtrer nos données. C'est de cette façon que nous avons retiré 6 variables de notre étude, qui contiennent plus de 95% de valeurs manquantes :

- *Glcid*
- *Slot*
- *AdNetwrok*
- *isVideoAd*
- *Page*
- *adContent*

Par exemple, nous avons étudié la variable *campaign*. Le graphe montre la répartition de cette dernière en fonction du nombre de transaction avec et sans revenue.

On note que plus de 50% des données appartiennent à la catégorie "*not set*".

C'est le cas de plusieurs autres variable dont les valeurs ne sont pas complètement renseignés. Nous allons voir dans la suite comment résoudre ce type de problème.



Data pre-processing

Après étude, nous remarquon qu'il semble y avoir de nombreuses colonnes contenant des informations constantes, telles que NaN ou "*not available in demo*" ou "*(not set)*". Si des colonnes entières contiennent les mêmes informations comme celles-ci, elles ne seront pas utiles.

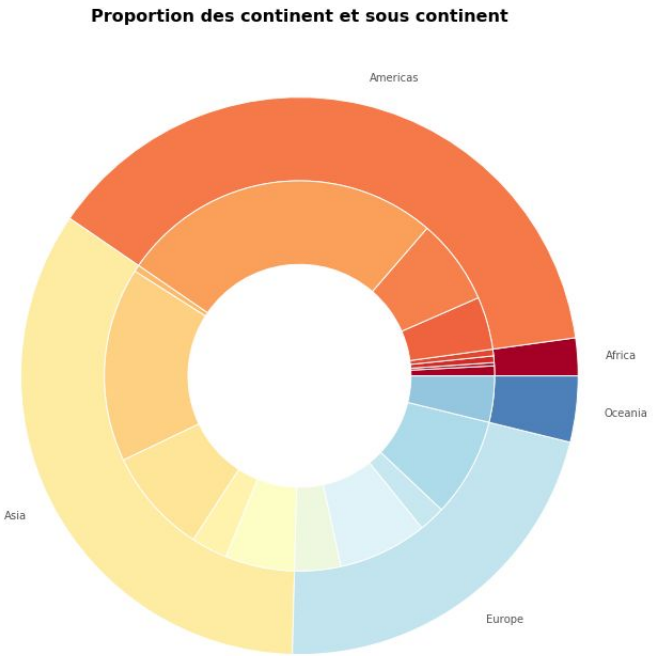
De ce fait, nous supprimons les colonnes qui n'ont qu'une seule valeur unique. De même pour les variables qualitatives qui contiennent beaucoup trop de valeur différentes et qui poseraient un problème de complexité pour lors de l'encoding.

Par ailleurs, les quelques valeurs manquantes restantes ont été transformées soit en 0 pour les variables numériques soit en *None* pour les autres.

De plus, pour les features apportant une donnée sur cadre temporelle tel que les dates, nous avons modifié leur format en DateTime pour pouvoir effectuer des visualisations adéquates.

Finalement, pour les features qualitatives restantes, nous les avons étudié puis transformé pour les regrouper en 4 ou 5 catégories principales représentant la majorité du dataset.

Data visualisation



continent	subContinent	sessionId
Africa	Eastern Africa	3
	Northern Africa	1
	Southern Africa	2
	Western Africa	2
Americas	Caribbean	16
	Central America	26
	Northern America	11143
	South America	98
Asia	Central Asia	2
	Eastern Asia	59
	Southeast Asia	32
	Southern Asia	11
	Western Asia	21
Europe	Eastern Europe	14
	Northern Europe	27
	Southern Europe	8
	Western Europe	30
Oceania	Australasia	14

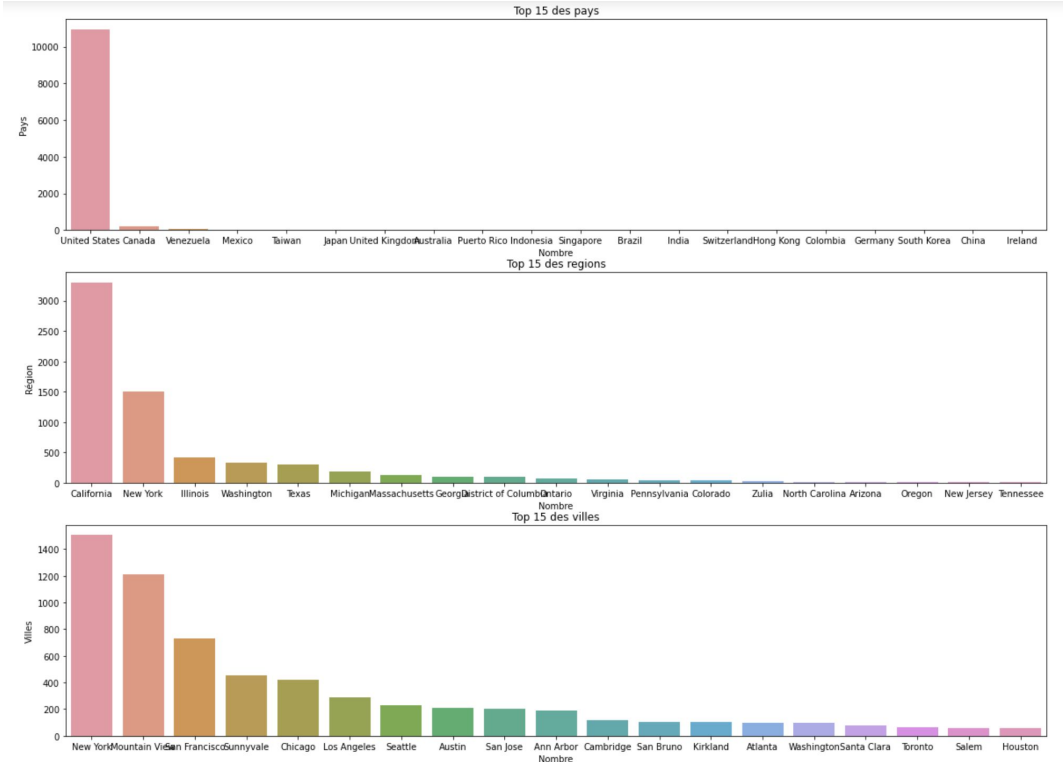
DONNÉES GÉOGRAPHIQUES

En s'intéressant aux données géographiques, on remarque un nombre de transaction avec revenu, beaucoup trop grand pour l'amérique du nord, comparé aux autres sous- continent.

Nous ne pouvions pas négligés les autres zones géographiques, c'est pourquoi nous notons cet écart mais souhaitons tout de même observer la répartition des transactions dans les autres continents/sous continent.

Après l'Amérique du nord on remarque que les autres sous continent de l'Amérique et l'Asie suivis de l'Europe sont les zones géographiques comptant le plus de transaction avec revenues.

Data visualisation



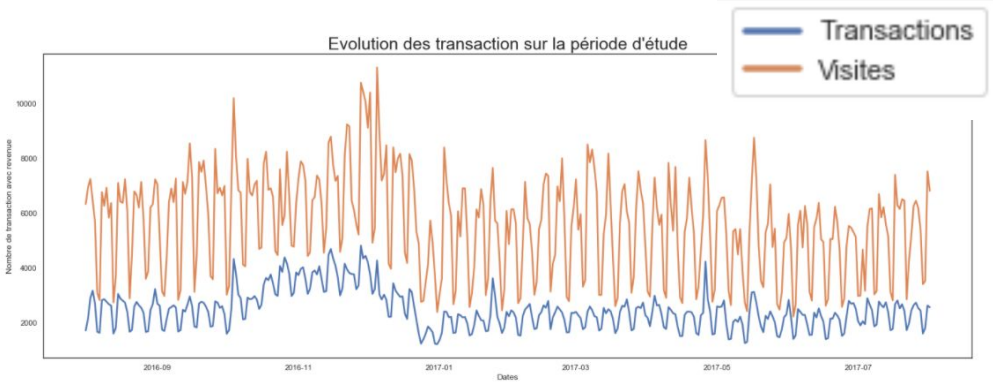
DONNÉES GÉOGRAPHIQUES

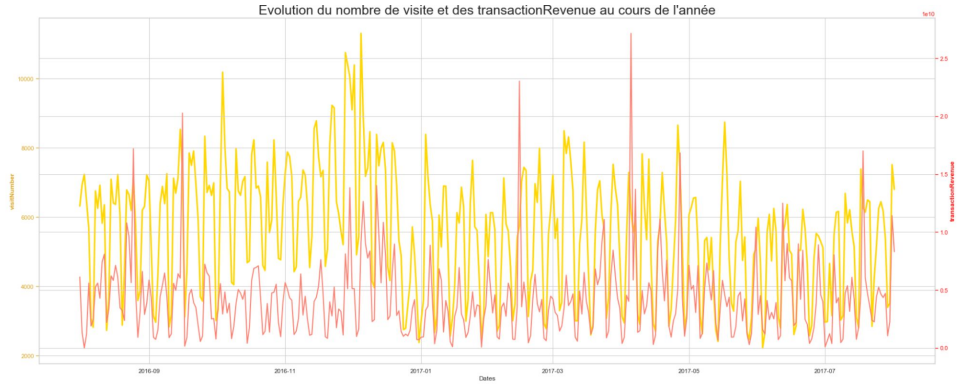
Nous disposons également des pays, des régions et des villes comme données géographiques. Après analyse de celle-ci, nous remarquons qu'il s'agit toujours des régions et villes des Etats-Unis qui regroupent la plupart des transactions avec revenus.

Ainsi, ce sont effectivement les clients d'Amérique du nord qui contribue le plus au visite de site et surtout qui effectue des achats. Les asiatiques contribuent égalements à quelques visite en ligne avec achats, mais il est à noter que leur contribution est faible comparé à celle des Etat-Unis.

Au vu de l'analyse des zones géographiques effectué sur notre dataset et du grand nombre de ville, région et pays qu'il contient, nous avons décidé de conserver seulement les sous continents comme donnée géographique pour notre analyse.

Data visualisation





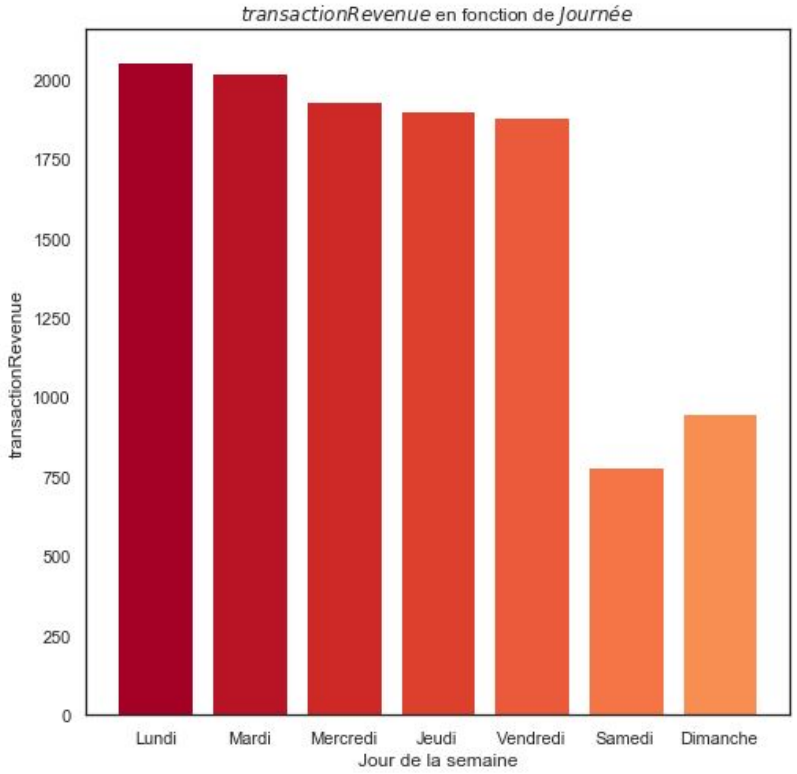
Nous avons ensuite analyser les données temporelles que nous disposions et notamment les dates de visites des transactions avec revenues. A partir de ces données la, nous avons extrait les jours et les mois pour approfondir notre analyse.

Nous observons que le nombre de transactions est en effet élevé en fin d'année (Novembre, décembre). Il s'agit des périodes ou les achats sont massives en raisons du BlackFriday et des fêtes de fin d'années.

D'autant plus, concernant l'évolution du nombre de visite et des transactions avec revenus, globalement il ne semble pas y avoir de tendance à la hausse sur une année.

On note également que l'affluence des visites est proportionnel au nombre de transactions

Data visualisation

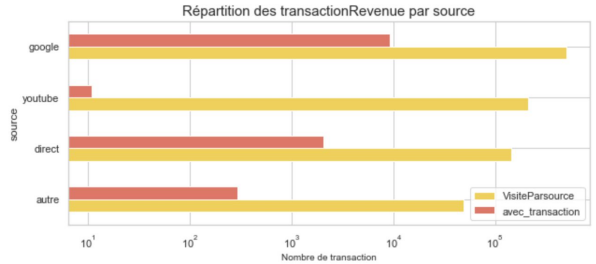


Nous nous sommes ensuite intéressé au jours de la semaine. Grâce à un histogramme on a pu observer quels sont les jours de la semaine où les utilisateurs ont effectué le plus d'achats.

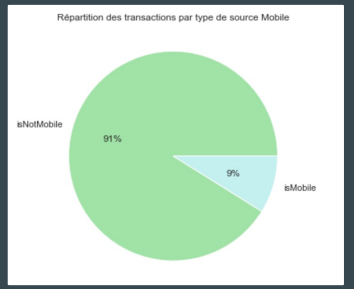
Avant cette étude on avait tendance à penser que c'était principalement les jours de week-end où les personnes effectuent le plus d'achat. Mais le graphique nous montre que c'est en semaine qu'il y a le plus de transactions effectuées.

Cela pourrait s'expliquer par le fait que les personnes sortent plus souvent le week-end, donc on moins le temps de faire des achats en ligne.

Data visualisation



Concernant les appareils permettants aux utilisateurs de visiter des sites, seulement 9% d'entre eux sont des mobile. Le reste des visites sont donc effectués avec d'autres appareils : les ordinateurs et tablettes.

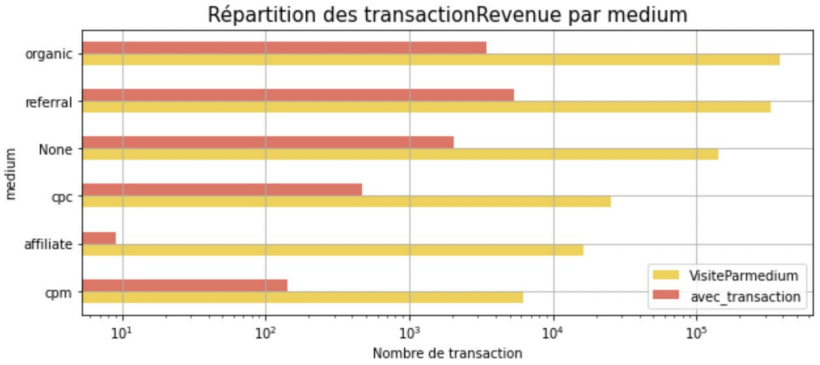


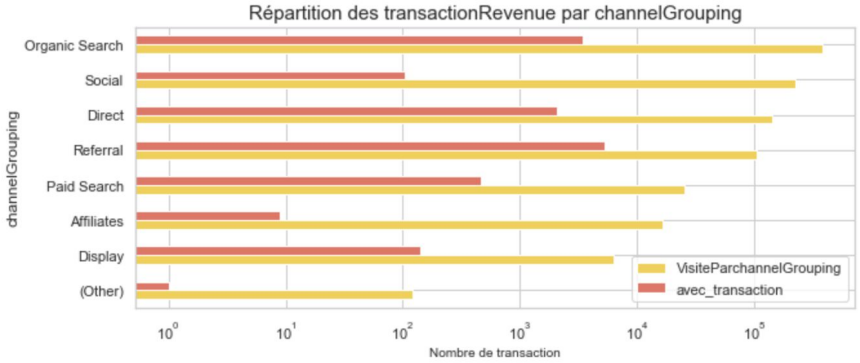
Les navigateurs web les plus utilisés pour les visites en lignes sont Chrome, Safari, Firefox, Internet Explorer et Edge. Il s'agit également des navigateurs les plus utilisés pour les transactions avec revenus.

Pour les systèmes d'exploitation, Windows, Macintosh, Android, IOS, Linux et Chrome OS sont les systèmes les plus utilisées pour des simples visites et pour des visites avec achats. Ces résultats sont plutôt cohérent le premier graphique ou la plupart des transactions ont été effectués sur Chrome ou Safari.

Enfin, la plupart des transactions avec revenus ont été faite soit par google soit directement via la plateforme concernée.

Data visualisation





Pour la variable medium, on remarque 6 valeurs différentes. On décide d'observer la répartition des visites ainsi que celle pour lesquelles un achat a été réalisé.

Pour avoir des résultats exploitables, on affiche pas la valeur réelle mais le logarithme. On peut voir que Organic et Referral ont presqu'autant de visites et de visite avec transaction, avec plus de 30 000 visites pour chacune et qu'il y a un écart qui va commencer à se créer pour les autres variables.

Pour la variable Channel Grouping, on effectue le même travail, ce qui nous amène à conclure que le nombre de visite n'est pas proportionnel selon qu'il y ai achat ou non.

Construction des modèles

Avant de procéder à la création de modèles, nous devons transformer nos données qualitatives en valeurs entières. Par conséquent, un OneHotEncoding a été effectué sur les variables que nous avons traité en amont.

Par la suite, sur la base de l'analyse que nous avons faite, nous pouvons lancé les modèles de régression. Pour cela, commençons par définir X et y. X correspondant au dataset auquel on retire les variables d'Id et y désigne la target.

Puisque la prédiction de *transactionRevenueTotal* est un problème de régression, nous avons expérimenté les modèles suivants :

- DecisionTreeRegressor
- BaggingRegressor
- GradientBoostingRegressor
- RandomForestRegressor

Pour cela, nous avons divisé l'ensemble de données en deux parties : l'une pour l'apprentissage et l'autre pour la prédiction. Le seuil de partage n'est pas fixé et est utilisé comme un hyperparamètre.

Pour tuner nos modèles, nous avons fait plusieurs tests en variant les hyperparamètres spécifiques et en utilisant une technique simple de validation croisée pour chaque algorithme.

Construction des modèles

Après l'obtention de 84 modèles en près de 12h de traitement, nous sélectionnons celui dont le R^2 sur le test est le plus élevé.

Le meilleur modèle est le Bagging Regressor avec comme split 40% et le nombre d'estimateurs fixé à 50.

	Methode	Modèle	Paramètres	R2_train	R2_test
0	BaggingRegressor	(DecisionTreeRegressor(criterion='mse', max_de...	{'split': 0.4, 'n_estimators': 50.0}	0.999626	0.999672
1	BaggingRegressor	(DecisionTreeRegressor(criterion='mse', max_de...	{'split': 0.4, 'n_estimators': 40.0}	0.999601	0.999642
2	BaggingRegressor	(DecisionTreeRegressor(criterion='mse', max_de...	{'split': 0.6, 'n_estimators': 40.0}	0.999576	0.999623
3	BaggingRegressor	(DecisionTreeRegressor(criterion='mse', max_de...	{'split': 0.4, 'n_estimators': 30.0}	0.999586	0.999619
4	BaggingRegressor	(DecisionTreeRegressor(criterion='mse', max_de...	{'split': 0.6, 'n_estimators': 30.0}	0.999586	0.999613
...	...	...	...	...	...
79	RandomForestRegressor	(DecisionTreeRegressor(criterion='mse', max_de...	{'split': 0.6, 'max_depth': 1.0, 'n_estimators...	0.982753	0.980126
80	RandomForestRegressor	(DecisionTreeRegressor(criterion='mse', max_de...	{'split': 0.6, 'max_depth': 1.0, 'n_estimators...	0.982753	0.980126
81	RandomForestRegressor	(DecisionTreeRegressor(criterion='mse', max_de...	{'split': 0.4, 'max_depth': 1.0, 'n_estimators...	0.982506	0.978987
82	DecisionTreeRegressor	DecisionTreeRegressor(criterion='mse', max_dep...	{'split': 0.4, 'depth': 1.0}	0.982506	0.978987
83	RandomForestRegressor	(DecisionTreeRegressor(criterion='mse', max_de...	{'split': 0.4, 'max_depth': 1.0, 'n_estimators...	0.982506	0.978987

84 rows × 5 columns

Construction des modèles

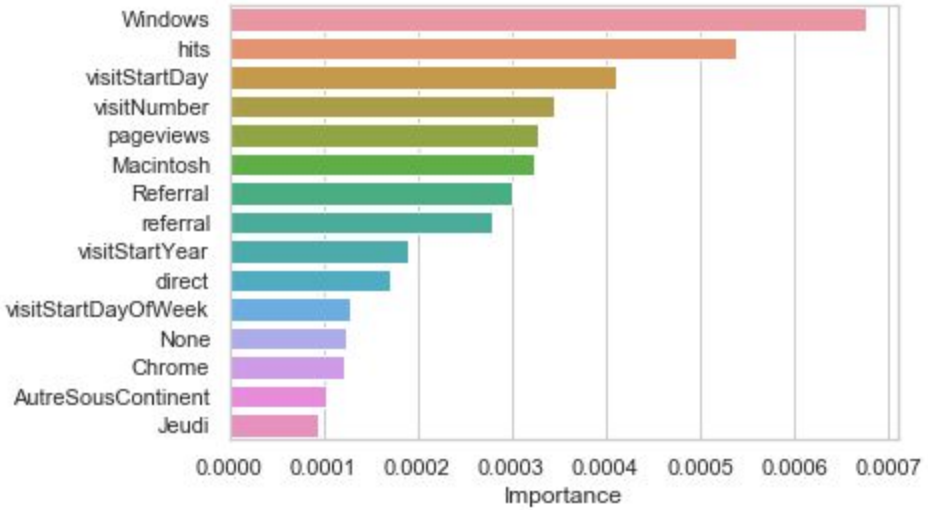
FEATURE IMPORTANCE :

Le modèle final sélectionné présente une méthode de représentation de l'importance de chaque variable dans la prédiction.

Le nombre de visite total est la variable qui a la plus grande importance pour la prédiction de la target.

En supprimant les variables liées aux visites et aux dates, nous allons voir l'importance des variables suivantes

Feature importance (après les variables de visites)



Conclusion

Comme pistes d'amélioration de notre étude, nous proposons les 2 idées suivantes :

- Étudier plus en détail les feature importance et tenter de les réduire dans le but de diminuer le temps de traitement des modèles sans pour autant diminuer la valeur du R^2

- Tester des algorithmes de DeepLearning en commençant par de simple Neural Network puis en ajoutant des couches si la performance augmente