Réalisé par
El YOUNSI Abdellah
MORSLI Omar

3°A FIG ISA Brest
Etudiants en troisième année à l'IMT Atlantique – campus de Brest
Sous la tutelle de Romain Billot, Philippe Lenca, Sorin Moga, enseignants-chercheurs à l'IMT Atlantique – campus de Brest

Rapport UV 213A : Fouille de données

Google Analytics Customer Revenue Prediction

Rapport
Version 1
Le 27/11/2018

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Contents

LISTE DES FIGURES

INTRODUCTION

Après une analyse de données fiscales de l'Angleterre, Russie, France, Suisse, Italie et Prusse, l'économiste et sociologue italien Vilfredo Pareto remarque que, dans chacun de ces pays, 80% des richesses sont détenues par 20% de la population. Ces observations l'ont conduit à annoncer une loi connue aujourd'hui sous le nom de 'la règle des 80/20' qui postule que 80% des effets sont obtenus par seulement 20% des causes.
Appliqué au revenu d'une entreprise, ce principe nous montre que seul un faible pourcentage de clients produit l'essentiel du revenu, ce qui invite les équipes marketing à relever le défi de trouver des stratégies promotionnelles les plus appropriées.

Dans ce rapport, nous analysons les données de Google Marchandise Store, GStore en abrégé. Ces données sont disponibles sur la plateforme américaine Kaggle. Sur ce site web, les entreprises proposent des problèmes en data science et offrent un prix aux participants ayant proposé les meilleurs algorithmes. Pour chaque compétitions, Kaggle fixe une métrique bien définie afin de mesurer la performance de chaque participant. La durée de chaque compétition s'étale généralement sur une durée allant de 2 à 3 mois.

Pour être rigoureux et méthodique, nous suivons les six étapes de la méthode Cross-Industry Standard Process for Data Mining (CRISP-DM) qui se résume en : Business understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment. La figure suivante illustre la méthodologie.

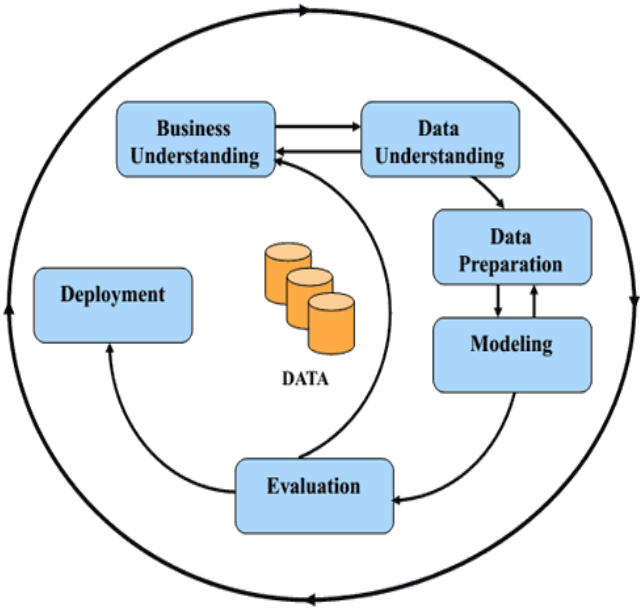

Figure 1. Le cycle de vie de l'exploration des données

I. BUSINESS UNDERSTANDING

Le but de ce challenge est de prédire le revenu total par client. Cette prédiction permettra à l'entreprise de réduire les coûts colossaux des campagnes de promotions, mieux gérer les budgets marketing et ainsi segmenter la clientèle pour mieux cibler les clients.

Ainsi, notre but est, à partir de l'historique des transactions de nombreux visiteurs de GStore, proposer un modèle du Machine Learning qui permettra de prédire le revenu d'un client donné. Et plus précisément, nous cherchons à prédire la variable suivante :
$y_{client} = \ln(1 + \sum_{i \geq 1} transaction(client, i))$ Correspond à la $i_{ème}$ transaction du client "client". Ainsi, pour éviter de manipuler des grandeurs qui varient sur une grande échelle et pour converger rapidement aux résultats souhaités l'utilisation du logarithme nous semble nécessaire.

L'évaluation du modèle se fait à travers la métrique Root Mean Squared Error (RMSE). Par la suite, nous choisissons le modèle avec le RMSE le plus faible.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

Avec : yi : le logarithme du vrai revenu de client i
yi_hat : l'approximation du yi

Pour trouver le meilleur modèle possible, nous allons tester plusieurs modèles Machine Learning : Régression Linéaire, Arbre de Décisions, Random Forests et Gradient Boosting.

II. DATA UNDERSTANDING

1. Collecte des données

Dans le cadre de cette compétition, Google nous a mis à disposition deux jeux de données au format CSV. De plus, des ressources matérielles suffisantes pour traiter le sujet (CPU, GPU, RAM de 17 Go, Stockage).

2. Description des données

train.csv : dataset que nous allons utiliser par la suite pour entraîner les modèles après l'application d'un split : x_train, x_val, y_train, y_val.
test.csv : dataset d'évaluation.

```
Loaded train_v2.csv. Shape: (1708337, 59)
Loaded test_v2.csv. Shape: (401589, 58)
```

Figure 2. Train/test dataset shapes

Chaque ligne des jeux de données (1708337 pour le training set et 401589 pour le test set) représente une visite au magasin par un client (c.-à.-d. un client peut avoir plusieurs visites au magasin : transactions).

3. Exploration des données

Nous avons au total 12 variables prédictives :

- fullVisitorId <chr> - identifiant unique pour chaque utilisateur du magasin de marchandises Google.
- channelGrouping <chr> - Le canal par lequel l'utilisateur est venu au magasin.
- date <dbl> - Date à laquelle l'utilisateur a visité le magasin.
- device <chr> - périphérique utilisé pour accéder à la boutique.
- geoNetwork <chr> - Cette section contient des informations sur la géographie de l'utilisateur.
- sessionId <chr> - Identifiant unique pour cette visite au magasin.
- socialEngagementType <chr> - Type d'engagement, "socialement engagé" ou "non socialement engagé".
- totals <chr> - Cette section contient les valeurs globales de la session.
- trafficSource <chr> - Cette section contient des informations sur la source de trafic à l'origine de la session.

- visitId <dbl> - identifiant de la session. Cela fait partie de la valeur généralement stockée sous le cookie __utmb. Il est unique à l'utilisateur. Pour un identifiant totalement unique, vous devez utiliser une combinaison de fullVisitorId et de visitId.
- visitNumber <dbl> - Le numéro de session de cet utilisateur. S'il s'agit de la première session, sa valeur est 1.
- visitStartTime <dbl> - L'horodatage (exprimé en heure POSIX).

D'autre part, il est important de noter que les champs device, geoNetwork, trafficSource, totals sont au format json. Ce qui explique le passage de 12 à 58/59 colonnes.
Puisque, nous cherchons à estimer le logarithme naturel de la somme de toutes les transactions de l'utilisateur, nous pouvons calculer la somme des transactions par utilisateur. Puis, visualiser ceci à travers un diagramme de dispersion.

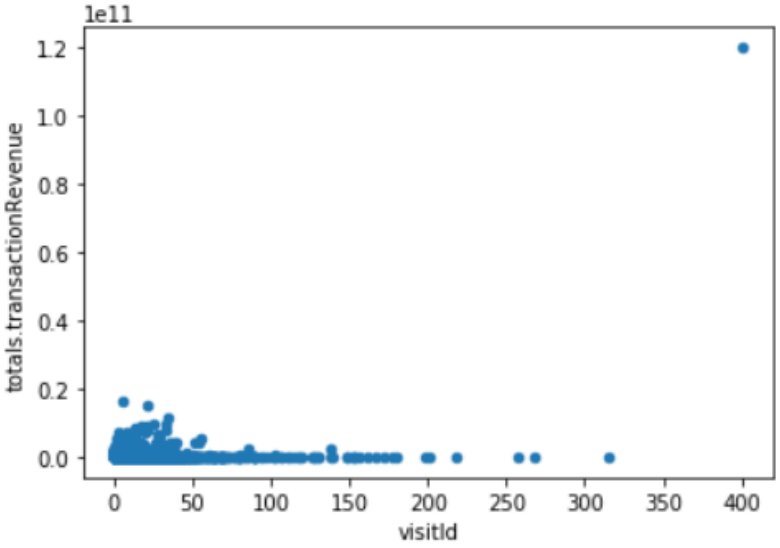

Figure 3. Distribution du Revenu

Pour plus de précision, nous avons procédé à un simple calcul des clients qui (ne) contribuent (pas) au revenu :

```
Total records:  1708337 customers do not contribute for revenue:  1689823 customers contribu
ting revenue:  18514
```

Figure 4. Clients (non) contributeurs au revenu

Le calcul, le graphe ci-dessus confirment les deux premières lignes de la vue d'ensemble de la compétition :

The 80/20 rule has proven true for many businesses–only a small percentage of customers produce most of the revenue. As such, marketing teams are challenged to make appropriate investments in promotional strategies.

Maintenant, nous pouvons visualiser la variation du revenu et des visites en fonction du temps:

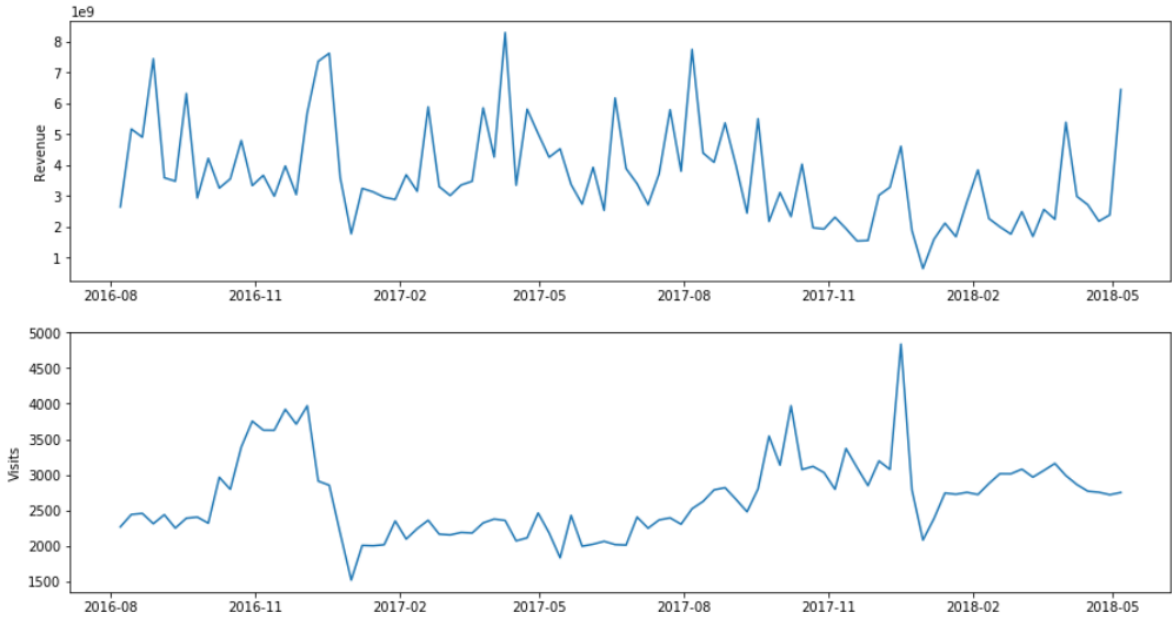

Figure 5. Variation du revenu et des visites en fonction du temps

Le nombre des visites a augmenté considérablement en Octobre 2016/2017 et a diminué en Décembre 2016/2017. Cependant, ces augmentations n'ont pas contribué à la croissance du revenu. Nous pouvons conclure que, les deux variables, nombre de visites et le revenu sont peu corrélées.

D'autre part, 'totals.pageviews' et 'totals.hits' pourront renforcer notre modèle (coefficient de corrélation=0.15) : Le revenu augmente lorsque le nombre des 'pageviews' et 'hits' augmente. Même avec une faible corrélation positive, ces variables peuvent avoir un rôle très important dans un modèle statistique.

4. Vérification de la qualité des données

Avant de manipuler les données et d'émettre plusieurs hypothèses, nous vérifions si les données sont bien manipulables. Ce qui nous intéresse principalement, c'est d'observer les colonnes constantes, les outliers et les valeurs manquantes.

```
Features with one unique values are :
Index(['socialEngagementType', 'device.browserSize', 'device.browserVersion',
       'device.flashVersion', 'device.language', 'device.mobileDeviceBranding',
       'device.mobileDeviceInfo', 'device.mobileDeviceMarketingName',
       'device.mobileDeviceModel', 'device.mobileInputSelector',
       'device.operatingSystemVersion', 'device.screenColors',
       'device.screenResolution', 'geoNetwork.cityId', 'geoNetwork.latitude',
       'geoNetwork.longitude', 'geoNetwork.networkLocation', 'totals.bounces',
       'totals.newVisits', 'totals.visits',
       'trafficSource.adwordsClickInfo.criteriaParameters',
       'trafficSource.adwordsClickInfo.isVideoAd',
       'trafficSource.isTrueDirect'],
      dtype='object')
```

Figure 6. Les colonnes constantes

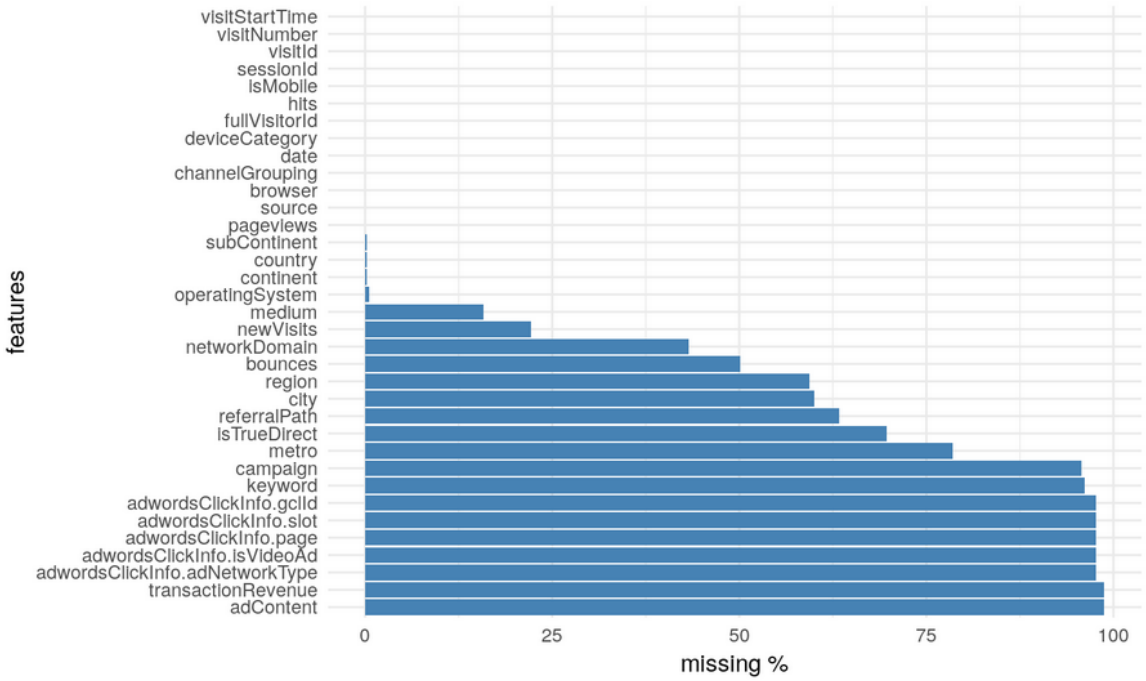

Figure 7. % des Missing values

Nous remarquons, que vous avons plusieurs champs constants et des champs avec des valeurs manquantes ou quasiment vides. Par la suite, nous expliquons comment nous traitons ces cas.

III. DATA PREPARATION

L'objectif de cette étape est de trouver des solutions aux problèmes rencontrés dans l'étape de "Data Understanding", sélectionner les variables les plus pertinentes pour notre problématique de régression et les transformer en conséquence.
Nous avons décidé de ne pas retenir pour nos modèles, toute variable conséquente ou contenant des informations sur ce qu'on essaye de prédire (*Data Leakage). De plus, les attributs ID ne renforce pas notre modèle (ils sont peu corrélés à ce qu'on essaye de prédire) et risque de provoquer des data leakages. Par la suite, nous regroupons le dataset par la variable 'fullvisitid' - l'unique id du client.
La liste des variables non retenues : 'date', 'visitStartTime', 'visitid' et 'sessionid'.

1. Sélection des variables significatives

La première chose à vérifier dans des grands jeux de données est la conformité des attributs entre le training dataset et le test dataset. Nous avons trouvé une seule variable de plus dans le training dataset que nous avons supprimée par la suite. Ensuite, nous avons effacé tous les champs constants (23 variables) - ceux que nous avons exploré dans la phase du Data understanding.

Vu qu'il est difficile de prouver qu'une variable prédictive quasiment vide est corrélée avec la variable cible, nous n'avons pas retenu les colonnes trafficSource et geoNetwork.

2. Nettoyage des données

Le tableau ci-dessous résume les remplacements de valeurs et les changements de types que nous avons appliqués aux datasets.

Variable	Valeur à remplacer	Valeur	Type
totals.transactionRevenu	missing values	0	Float
trafficSource.adContent	missing values	NoAdContent	String
trafficSource.keyword	missing values	NA	String
totals.sessionQualityDim	missings values	0	Integer
totals.timeOnSite	missings values	0	Integer
totals.pageviews	-	-	Integer
totals.hits	-	-	Integer

*Data Leakage happens a lot when we are dealing with data that has Time Series characteristics. That is, we have a series of data points that are distributed chronologically.

3. Transformation des données

Ensuite, nous avons aussi pris le parti de transformer certaines données, car elles n'étaient pas exploitables telles qu'elles nous avaient été fournies.
Nous avons ainsi décidé de convertir les variables catégorielles qui ont moins de 40 valeurs uniques en variables factices (dummy variables). Pour ceux qui contiennent plus de 40 valeurs uniques nous les avons convertis en type énuméré (ce qui est utile pour obtenir une représentation numérique de ces champs - ce qui compte dans ce cas c'est d'identifier des valeurs distinctes). Le tableau suivant résume les transformations :

Variable	Nombre des valeurs uniques
channelGrouping	8
device.browser	161
device.deviceCategory	3
device.operatingSystem	26
geoNetwork.city	1097
geoNetwork.subContinent	23
trafficSource.adContent	101
trafficSource.campaign	42

D'autre part, nous avons calculé le logarithme de 'totals.transactionRevenue' (d'après les règles de la compétition on est obligé de faire cette transformation pour prédire la bonne valeur).
Finalement, nous avons normalisé le dataset (MinMaxScaler), Pour éviter d'avoir des variables sur différentes échelles, et que, certaines pondérations se mettent à jour plus rapidement que d'autres.

IV. MODELING

La représentation de l'entrée peut se faire à travers une fonction continue, c'est-à-dire que la variable cible – revenu par client – prend des valeurs continues dans le temps. En effet, le cas que nous traitons ici peut être modélisé par une régression.

Par ailleurs, dans l'étape suivante, nous expliquons les choix d'algorithmes pour faire face à cette problématique de régression. D'autre part, Kaggle a mis à disposition un jeu de données d'apprentissage sur lequel on peut entraîner nos modèles. Puis, nous allons les évaluer par un deuxième jeu de données.

1. Algorithmes choisis

Il existe divers algorithmes de régression pour répondre à ce problème. Dans notre cas, nous avons choisi de tester les performances de 4 algorithmes :

- Régression linéaire
- Arbre de décisions
- Random Forests
- Gradient boosting

D'abord, nous avons testé un modèle de régression linéaire simple qui ne demande aucun tuning des hyperparamètres. Ensuite, un arbre de décisions, en utilisant les paramètres par défaut. Cette approche présente des meilleurs résultats par rapport au premier modèle sur la base d'entrainement. Cependant, nous avons remarqué qu'il présente un problème de sur-apprentissage c'est-à-dire qu'il établit une fonction de mapping entre les variables significatives et la variable cible sur le training dataset et non pas sur le test dataset. Vu que, les méthodes ensemblistes permettent de remédier au overfitting, nous avons évalué deux approches : Random Forests et Gradient Boosting.

Afin, d'obtenir une meilleure performance avec le Gradient Boosting, nous avons effectué une recherche exhaustive sur un espace réduit des hyperparamètres par la méthode de « gridSearch ». Sous ce prétexte, nous avons constaté une légère amélioration des performances.

V. EVALUATION

1. Évaluation des résultats

Le tableau suivant résume la performance de chaque modèle.

Modele	RMSE
Regression Lineaire	1.8514
Arbre de decisions	1.6289
Random Forests	1.6122
Gradient Boosting	1.5908

Nous remarquons que la régression linéaire performe moins bien que tous les autres algorithmes. Ceci s'explique par la non-existence d'une relation linéaire entre les variables significatives et la variable cible. Le Gradient Boosting obtient le meilleur score une erreur de 1.5908 ce qui est légèrement inférieure à celle de Random Forests. Avec ce modèle, nous arrivons les 542 premiers. Comme Kaggle l'explique, certains participants avaient accès à la base de test, ce qui explique un « RMSE » nul pour certains d'entre eux. Compte tenu de ce fait, le premier dans le classement arrive avec un score de 1.2954.

2. Axes d'Amélioration

L'ajout des variables significatives, autre que celles fournies par Kaggle aurait amélioré la performance des modèles. En effet, cette étape, consiste à créer des variables à d'opérations arithmétiques simples comme l'addition, soustraction, multiplication et division de plusieurs variables.

Un autre aspect que nous aurions peut-être mieux développé est l'aspect temporel de la variable cible. En effet, notre cible est le revenu d'un client sur une période donnée. Ainsi, une étude robuste nécessite des modèles de Forecasting beaucoup plus sophistiqués, c'est-à-dire qu'ils prennent en compte la dépendance temporelle de la variable cible.

D'autre part, dans notre cas, nous ne pouvions pas étudier un « stacking » de modèle, à cause de la corrélation inter-modèles. Le tableau suivant résume les corrélations entre les modèles.

	linearRegression	decisionTree	randomForest	gradientBoosting
linearRegression	1	0.922581	0.92498	0.924177
decisionTree	0.922581	1	0.999227	0.999309
randomForest	0.92498	0.999227	1	0.999519
gradientBoosting	0.924177	0.999309	0.999519	1

3. Processus de révision

A travers ce projet, nous avons compris que la partie la plus longue d'un projet de fouille de données est effectivement la phase de préparation des données. Après avoir testé plusieurs modèles et approches nous nous sommes rendu compte que nous pouvons encore améliorer les résultats. En effet, nous n'avons pas intégré la totalité des informations à notre disposition. Nous aurions aussi pu passer plus de temps sur la phase de sélection des variables et essayer de créer de nouvelles variables (produit de 2 données…).

BIBLIOGRAPHIE

[1]. Google Analytics Customer Revenue Prediction, disponible sur : https://www.kaggle.com/c/ga-customer-revenue-prediction

[2]. Méthodologie CRISP-DM, disponible sur : https://www.dummies.com/programming/big-data/

[3]. Data Leakage, disponible sur : https://towardsdatascience.com/data-leakage-part-i-think-you-have-a-great-machine-learning-model-think-again-ad44921fbf34?fbclid=IwAR2EJXr5NmFZBd2T8zZiZ0cmg_DaBkCXz_TcK1wJzj_vjT1hmtCd3LSITG8

[4]. Gradient Boosting from scratch, disponible sur : https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d