

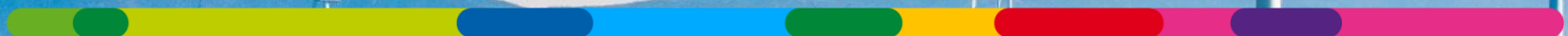
---

# ENGIE Data Science Challenge

B2C - Predict conversion rates on breakdown services

20 July 2016

---



Confidential



Limited



Free



Internal

---

# Table of content

---

- 1 Preprocessing
- 2 Feature creation / processing
- 3 Models
- 4 Features importance
- 5 Final blend
- 6 Tools and frameworks
- 7 Source code

# Preprocessing

- Fusion des 2 datasets train.csv et test.csv afin d'éviter de réécrire les mêmes opérations 2 fois.
- Conversion des dates au format yyyy-mm-dd :
  - ❖ variable **DT\_DEBT\_ASSR** : Date où le client a été contacté pour lui proposer DEGEX+ ou DEPEX+
  - ❖ variable **DATE\_EMMENAG** : Date d'emménagements dans son logement du client
  - ❖ variable **DATE\_ANC\_CLI** : Date du premier contrat du client chez ENGIE

## Feature creation / processing

- Ajout d'une variable **Prev\_DT\_DEBT\_ASSR** ( = DT\_DEBT\_ASSR – 1jour) pour matérialiser le jour précédent la date où client a été contacté pour lui proposer DEGEX+ ou DEPEX+
- Split des dates **DT\_DEBT\_ASSR** et **DATE\_ANC\_CLI** en jour, mois, année, trimestre , jour de la semaine.
- Split de la date **Prev\_DT\_DEBT\_ASSR** en jour de la semaine.

# Models

J'utilise le gradient boosting machine (gbm) reposant sur des arbres de décisions.

Un arbre de décision permet de séparer nos clients suivant une série de questions posées automatiquement par l'algorithme.

Le gbm est une méthode itérative :

- La 1<sup>ère</sup> itération construit un 1<sup>er</sup> arbre de décision et fournit une 1<sup>ère</sup> erreur de prédiction. → algo faible 1
- La 2<sup>ème</sup> itération construit un 2<sup>ème</sup> arbre de décision avec l'objectif de réduire l'erreur commise à la 1<sup>ère</sup> itération. → algo faible 2
- La 3<sup>ème</sup> itération construit un 3<sup>ème</sup> arbre de décision avec l'objectif de réduire l'erreur commise à la 2<sup>ème</sup> itération. → algo faible 3
- On continue ainsi le processus d'itération jusqu'à atteindre un point d'arrêt (paramétrable).

Le résultat final du gbm est une combinaison linéaire des algos faibles.

La dépendance des arbres rend le modèle vulnérable aux valeurs aberrantes et au sur-apprentissage (overfitting).

Ici j'entraîne une grille d'algorithmes gbm sur 80 % des données de training et je teste la capacité de généralisation sur les 20% de données restantes.

Les gbm sont générés automatiquement suivant un balayage aléatoire de l'espace des hyper-paramètres. Ensuite je classe ces gbm du plus performant au moins performant suivant la métrique AUC.



# Feature importance 1/2



Feature\_importance.zip

Variables les plus importantes pour le gbm le plus performant avec 80 % des données de training (gbm1)

Variable	Relative importance	Scaled importance	Percentage
Prev_DT_DEBT_ASSR	1054.119385	1	0.150306027
day_DT_DEBT_ASSR	924.2957764	0.876841646	0.131794584
weekday_Prev_DT_DEBT_ASSR	762.920166	0.723751197	0.108784167
ass_fact	427.4271851	0.40548271	0.060946495
weekday_DT_DEBT_ASSR	374.8261108	0.35558222	0.053446151
DT_DEBT_ASSR	257.723877	0.244492114	0.036748638

On constate que la variable la plus importante est celle que j'ai créé et qui correspond à la date précédant le jour où client a été contacté pour lui proposer DEGEX+ ou DEPEX+

## Feature importance 2/2

Variables les plus importantes pour la validation croisée du gbm le plus performant avec 100 % des données de training (cvgbm1) :

Variable	Relative importance	Scaled importance	Percentage
Prev_DT_DEBT_ASSR	1334.299927	1	0.159496897
day_DT_DEBT_ASSR	1149.224976	0.861294341	0.137373775
weekday_Prev_DT_DEBT_ASSR	923.6818237	0.692259518	0.110413245
ass_fact	557.9256592	0.418141115	0.06669221
weekday_DT_DEBT_ASSR	384.7984009	0.288389734	0.045997268
DT_DEBT_ASSR	273.8998718	0.20527609	0.032740899

Idem on remarque que la variable la plus importante est celle que j'ai créé et qui correspond à la date précédent le jour où client a été contacté pour lui proposer DEGEX+ ou DEPEX+

## Final Blend

Avec le tuning des hyper-paramètres, j'obtiens 500 modèles gbm.  
Je garde les 10 modèles les plus performants (AUC élevés).

Ensuite j'effectue une validation croisée (5-folds) sur l'ensemble du training set avec chacun des 10 modèles.

J'applique ces 10 nouveaux modèles issus de la validation croisée sur le testing set (test.csv) afin de prédire la probabilité de souscription au contrats DEGEX+ / DEPEX+.

J'obtiens ainsi 10 vecteurs de prédictions dont la moyenne sera mon résultat final à soumettre.

A noter que la moyenne arithmétique n'est pas forcément la combinaison linéaire la plus optimisée.  
Je n'ai pas eu le temps d'étudier l'application d'un 2<sup>ème</sup> niveau de machine learning (type glm, random forest, deeplearning ...) sur les 10 modèles gbm (piste d'amélioration).





---

# Tools and framework

---

Langage utilisé : R

Outil : Rstudio

Framework : H2O

Librairies pour le preprocessing et le feature engineering :

- ✓ Lubridate
- ✓ Dplyr
- ✓ Tidyrr

---

# Source Code

---



Final\_script.R