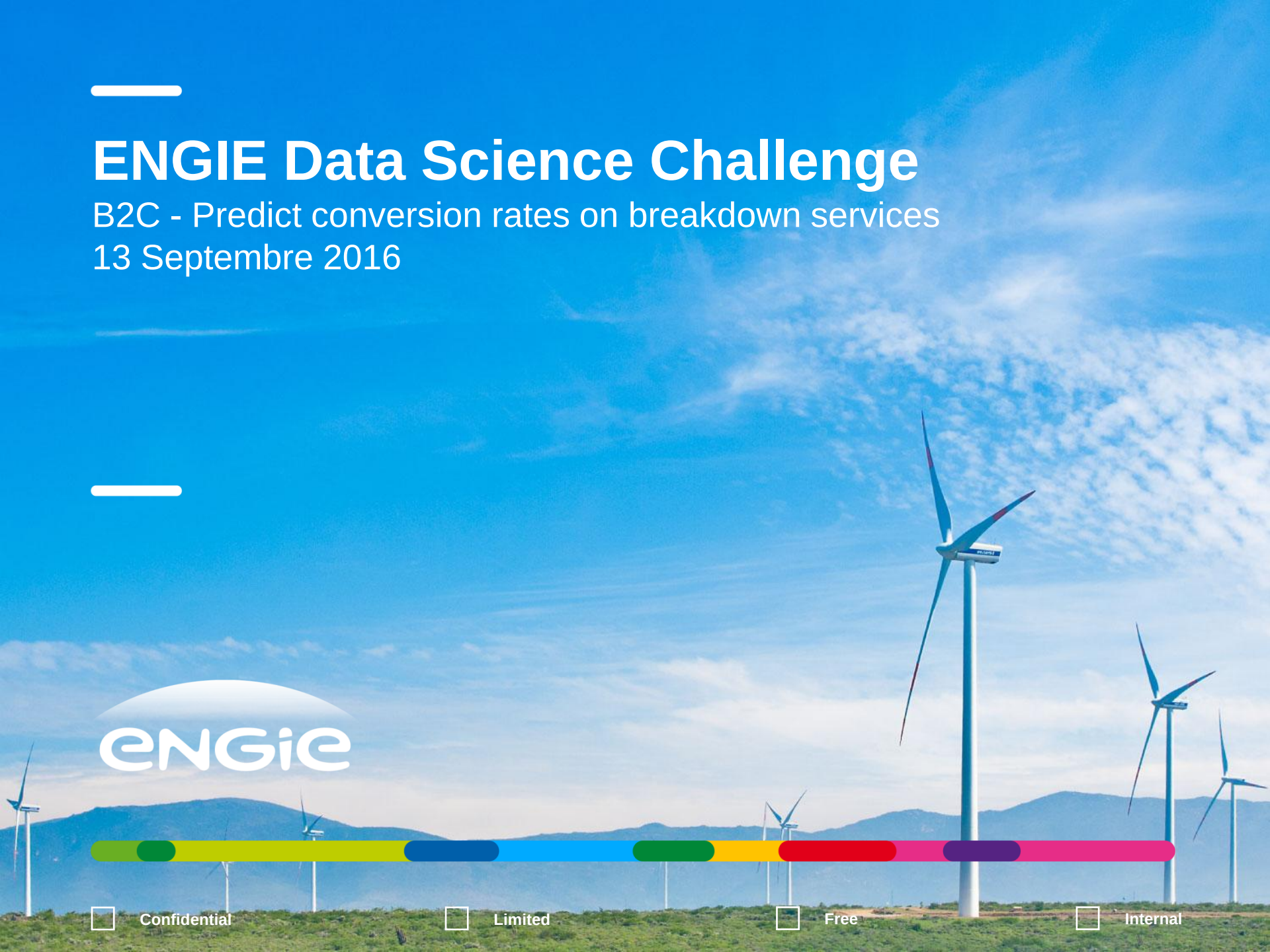
ENGIE Data Science Challenge

B2C - Predict conversion rates on breakdown services

13 Septembre 2016



☐ Confidential ☐ Limited ☐ Free ☐ Internal

Table of content

Pre-processing

- 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+
  - Pourquoi cette nouvelle variable ?
    Je fais l'hypothèse que la préparation des appels se fait à J – 1 et que potentiellement cette date de préparation peut influer sur la probabilité de souscrire aux contrats.
- 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.
  - Pourquoi ces splits de dates?
    Je considère que nous fournissons plus d'énergies dans notre travail le lundi que le vendredi et qu'il y a des périodes de l'année où nous sommes plus efficaces (en fonction des vacances, de la météo, des fêtes, etc…) . Ces granularités vont aider le modèle prédictif à discerner des schémas plus fin durant son apprentissage.

Models

J'utilise le gradient boosting machine (gbm) du framework de machine learning h2o.
L'idée est de moyenner un ensemble d'arbres de décisions où chaque arbre va réduire les erreurs de l'arbre précédant.
Un arbre de décision permet de séparer nos clients suivants une série de questions posées automatiquement par l'algorithme. Ici chaque arbre est construit à partir d'un sous-ensemble aléatoire (avec remise) du dataset initial.
Le gbm est une méthode itérative. Ici il est combiné à la méthode du gradient stochastique afin d'améliorer la généralisation.

- La 1ère itération construit un 1ère arbre de décision sur un échantillon E1 et fournit une 1ère erreur de prédiction ➔ algo faible 1
- La 2ème itération construit un 2ème arbre de décision sur un échantillon E2 avec l'objectif de réduire l'erreur commise à la 1ère itération ➔ algo faible 2
- La 3ème itération construit un 3ème arbre de décision sur un échantillon E3 avec l'objectif de réduire l'erreur commise à la 2ème 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.

Paramètres les plus importants : nombre d'arbres, profondeur des arbres et taux d'apprentissage.
Plus on diminue le taux d'apprentissage, plus on augmente le nombre d'arbre et vice versa.

La dépendance des arbres rend le model vulnérable aux valeurs aberrantes et au sur-apprentissage (overfitting). Cependant l'échantillonnage aléatoire du dataset initial permet de réduire cette faiblesse.

Ici j'entraine une grille d'algorithmes gbm sur 80 % des données de training et je teste la capacité de généralisation sur les 20% de données restant.
Les gbm sont générés automatiquement suivant un balayage aléatoire de l'espace des hyper-paramètres.

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 précédent 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 2ème 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
- ✓ Tidyr

Source Code

Final_script.R