



14/11/2019

Kaggle IEEE-CIS Fraud Detection Challenge

Rapport



Benjamin DUBREU

PARCOURS ML ENGINEER- CENTRALE SUPELEC/
OPENCLASSROOMS

TABLE DES MATIERES

Contexte.....	4
IEEE-CIS	4
VESTA Corporation	4
La competition.....	4
LES données.....	5
Volumetrie	5
Les variables	5
STRATEGIE	6
Approche CClassique	6
Approche Inversee.....	6
Schema de validation	6
LA SELECTION DES FEATURES.....	8
Permutation importance	8
Modelisation	9
Grid Search.....	9
Ensembling	9
Résultats.....	9

CONTEXTE

IEEE-CIS

IEEE-CIS travaille sur une grande variété de thématiques liées à l'intelligence artificielle et au machine learning.

VESTA CORPORATION

VESTA Corporation, fondée en 1995 est le leader des solutions de paiement e-commerce sécurisées, et garantit plus de 18 milliards de dollars de transactions chaque année.

LA COMPETITION

IEEE-CIS et VESTA corporation ont mis à disposition des compétiteurs de Kaggle un jeu de données de transactions en ligne.

L'objectif est de déceler le maximum de transactions frauduleuses, tout en minimisant le nombre de transaction classifiées comme frauduleuses alors qu'elles ne le sont pas.

LES DONNEES

VOLUMETRIE

- 435 Features ou Variables
- 590 000 Entrées

LES VARIABLES

Transaction Table

- TransactionDT : timedelta depuis le début de la période de référence
 - TransactionAMT : montant de la transaction (USD)
 - ProductCD : product code (le produit concerné par chaque transaction)
 - Card1 - card6 : informations sur la carte de paiement (type de carte, banque d'origine.)
 - addr : addresses
 - dist : distance
 - P_ and (R__) emaildomain : nom de domaine des adresses email
 - C1-C14 : Comptes (comme : « combien d'adresses sont associées avec cette carte de paiement », etc). La signification est masquée !
 - D1-D15 : timedeltas (comme le nombre de jours depuis la dernière transaction...)
 - M1-M9 : match (« même nom sur la carte de paiement et l'adresse ? », etc).
 - Vxxx: « Vesta engineered rich features », (rankings, countings, etc...)
-

Identity Table

Les variables de cette table sont des informations pouvant identifier l'utilisateur (adresse IP, Proxy, etc) ainsi que sa « signature digitale » (UA/Navigateur/OS/version/ etc) associées avec les transactions. Les noms des variables sont masqués (par id01, id02, etc) pour des raisons de protection de la vie privée.

STRATEGIE

APPROCHE CLASSIQUE

1. Obtenir les données
2. Exploration des données (EDA)
3. Visualisation
4. Nettoyage
5. Feature Engineering
6. Modélisation

APPROCHE INVERSEE

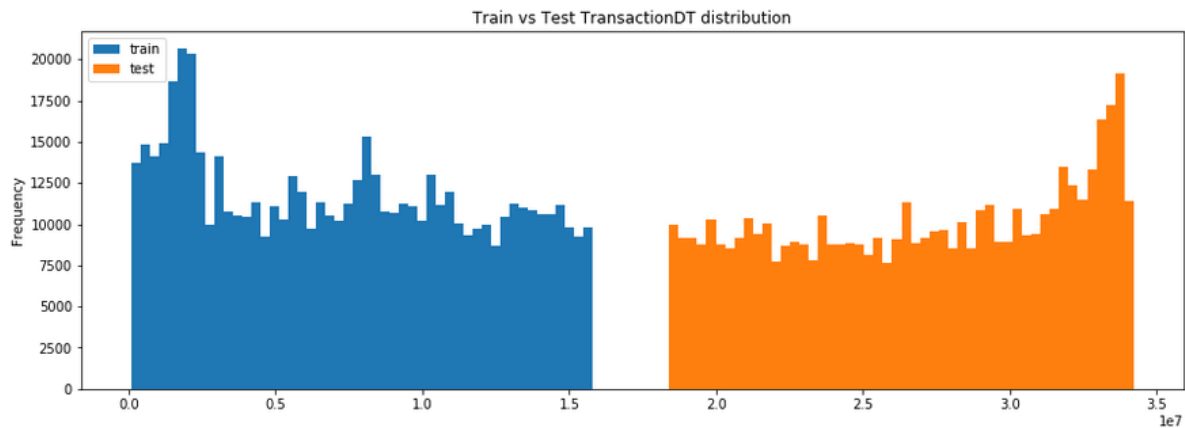
1. Obtenir les données
2. Nettoyage minimaliste pour permettre la modélisation
3. Modélisation
4. Feature Selection
5. Visualisation
6. Feature Engineering
7. Modélisation

Au vu du trop grand nombre de variables, faire une EDA classique aurait été trop chronophage. Cette approche un peu inversée permet de rapidement faire un premier tri. Les nouveaux algorithmes comme XGBoost ou LightGBM gèrent les valeurs manquantes, donc le nettoyage nécessaire est vraiment minimal.

Cette méthode est préconisée par Jérémie Howard, ancien N°1 de Kaggle et aujourd'hui professeur à Stanford : <http://course18.fast.ai/ml>

SCHEMA DE VALIDATION

Les données ont une dimension temporelle. Il est possible qu'au cours du temps, les fraudeurs utilisent, par exemple, de nouveaux navigateurs, ou leurs dernières versions. D'ailleurs, certaines données ont des valeurs qui évoluent au cours du temps (la moyenne de cette variable pour le mois d'Aout sera différente de celle pour le mois de Mars).

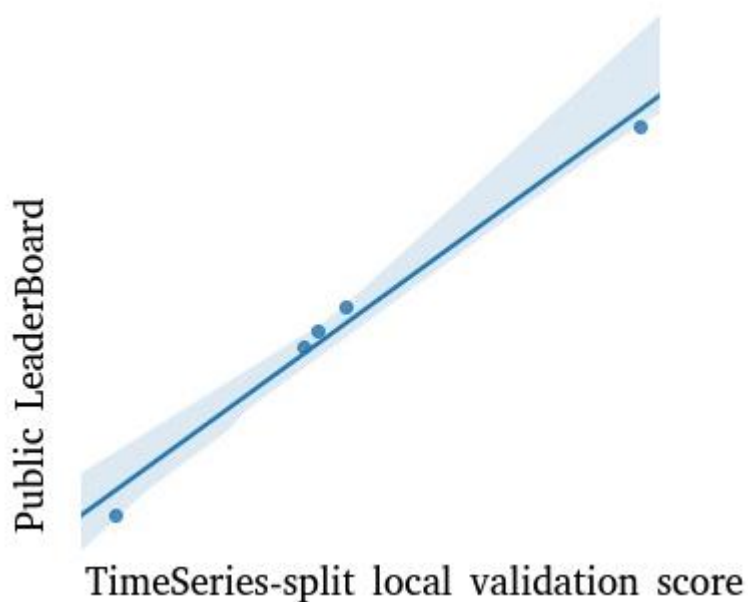


Comme le jeu de données test (« public » et « private » « leaderboards ») se situe *plus tard dans le temps* que le jeu de données d'entraînement, il faut une stratégie de validation qui suive cette logique.

Dans ce sens, une approche a été développée qui consiste à :

- Entraîner le modèle sur les données jusqu'à Mai (non inclus)
- Laisser de côté les données du mois de Mai
- Valider sur les données du mois de Juin

Les scores obtenus sur le mois de Juin sont très fortement corrélés aux scores obtenus sur le leaderboard public :



Cette stratégie de validation est retenue pour la compétition.

LA SELECTION DES FEATURES


PERMUTATION IMPORTANCE

La méthode de feature importance implémentée par défaut dans scikit-learn est biaisée (voir <https://explained.ai/rf-importance/index.html>).

Pour contrebalancer le biais en faveur des variables à haute cardinalité, il est préférable d'utiliser une méthode appelée « permutation importance ».

Elle consiste à obtenir une baseline, puis, pour chaque variable disponible, en mélanger les valeurs comme suit :

Height at age 20 (cm)	Height at age 10 (cm)	...	Socks owned at age 10
182	155	...	20
175	147	...	10
...
156	142	...	8
153	130	...	24



Une fois les valeurs permutées aléatoirement, on réalise à nouveau la prédiction sur le jeu de validation. Si le score diminue, la variable est importante. S'il reste le même (ou s'améliore), on peut la supprimer (les valeurs contenues dans ce champ n'apportent aucune aide à la décision).

MODELISATION

GRID SEARCH

La modélisation est trop longue pour envisager un grid search des meilleurs paramètres. En commençant cette compétition plus tôt (et en se focalisant uniquement sur celle-ci), cette méthode classique du machine learning aurait pu et dû être ajoutée.

ENSEMBLING








K Fold ensembling:

- 1) utiliser K-fold pour splitter l'ensemble du dataset (Mai-Juin inclus)
- 2) entrainer un nouveau modèle sur chacun de ces folds
- 3) faire une prediction sur le jeu de test à l'aide de chacun de ces modèles
- 4) réaliser la moyenne des prédictions

RESULTATS

Cette méthodologie de travail obtient un score sur le leaderboard privé de 0.925907. C'est une baisse notable par rapport au score sur le leaderboard public (0.951610), mais la plupart des compétiteurs ont connu une telle baisse (voir beaucoup plus grande).

On peut se féliciter que la stratégie de validation temporelle, couplée à une sélection drastique des features, ait porté ses fruits :

1343	▲153	Jupeeem		0.925928	6	1mo
1344	▼71	3A204		0.925925	3	2mo
1345	▼307	Bermuda's Δ		0.925924	306	1mo
1346	▲416	Benjamin Dubreu		0.925907	17	1mo
1347	▼663	kitaura		0.925907	2	1mo
1348	▼1005	Fredrik Jonsson		0.925901	77	1mo
1349	▼197	Yiolino		0.925892	138	2mo

En effet, beaucoup de compétiteurs ont perdu énormément de places. Cette stratégie, solide, a permis d'en gagner 416 lors du passage du leaderboard public au privé.