Data Challenge LinkValue

Introduction

Voici quelques consignes à propos des data challenges mis à disposition des candidats aux postes de Data Scientists chez LinkValue.

Quelque soit le score obtenu par le candidat à la fin, l'objectif de ces challenges est avant tout de permettre au candidat de démontrer sa capacité à réfléchir sur des problématiques concrètes en utilisant des techniques d'apprentissage statistique. Une attention particulière sera portée à la **clarté** et la **propreté du code**, ainsi qu'à la **justification** des directions choisies par le candidat.

Le candidat a une liberté totale quant au choix de la stack technique. Pour des langages autres que *R* et *Python*, il devra néamoins préparer une petite introduction présentative des avantages du langage choisi.

La présentation peut se faire sous n'importe quelle forme. On conseille néanmoins l'utilisation d'un notebook (jupyter) qui permet d'allier code et visualisation de manière propre et intelligente.

1 Prédiction de l'intérêt des avis sur les produits

PriceMinister est l'une des marketplaces de e-commerce les plus populaires en France. L'objectif de ce challenge est de prédire si l'avis d'un utilisateur sur un produit sera utile pour les autres utilisateurs. C'est un problème de classificatin binaire, évalué par le calcul de l'aire sous la courbe ROC (AUC). Les données sont en Français.

1.0.1 Data

Les données sont séparées en 3 fichiers.

train.csv : Les données d'entrainement (*reviews*). Fichier au format CSV (séparateur *;*) contenant les champs suivants :

- **ID** : Identifiant de l'avis
- **review_content** : Contenu de l'avis sur le produit rédigé par l'utilisateur
- **review_stars** : Titre de l'avis sur le produit rédigé par l'utilisateur
- **review_stars** : Note donné par l'utilisateur au produit (de 1 à 5)
- **product** : Identifiant du produit hashé
- **Target** : Classe de l'avis. 1 pour les avis jugés **utiles** (plus d'utilisateurs trouvent cet avis utile plutôt que pas utile) et 0 pour les avis pas utiles (plus d'utilisateurs trouvent cet avis pas utile plutôt que utile).

test.csv : Les données de test (*reviews*). Fichier au format CSV (séparateur *;*) contenant les même champs que *train.zip* sauf le champ *Target*. Il faut générer les probabilités que ces avis soient utiles (i.e. appartenant à la classe 1).

candidate_submission.csv : Un exemple du fichier à renvoyer contenant les probabilités générées par le modèle du candidat. Fichier au format CSV (séparateur *;*) contenant les champs suivants :

- **ID** : Identifiant de l'avis (doit correspondre aux identifiants du set de test).
- **Target** : Probabilité que l'avis soit de classe 1.

Exemple d'une entrée :

ID ;review_content ;review_title ;review_stars ;product ; Target
40 ;le jeu est très bon le seul défaut c'est que les pièces ne tiennent pas bien sur l'échiquier ;un jeux d'echec treès attrayant et complet bonne acquisition pour son prix défiant la concurrence ;5 ; 5bec8c1b067ab2a211d2.... ; 1

1.0.2 Scoring

Le score utilisé est la métrique *Area Under Curve (AUC).*

```
from sklearn.metrics import roc_auc_score
def score_function(y_true, y_pred):
    return roc_auc_score(y_true, y_pred)
```