

Développement et déploiement d'un outil de management de la brand safety dans DV360 avec Google Apps script et Google Sheet & Prédiction du nombre d'impressions d'une campagne

Mamadou DIOUF

GroupM, France, Paris

31 Août 2022

Plan

- 1 Introduction
 - Contexte
 - Objectifs
- 2 Automatisation des processus
 - Outils
 - Développement
 - Problèmes rencontrés
 - Déploiements
- 3 Apprentissage Statistique
 - Analyse de données
 - Base de données
 - Régression linéaire
 - Amélioration du Gradientboosting régression
- 4 Conclusion

Sommaire

- 1 Introduction
 - Contexte
 - Objectifs
- 2 Automatisation des processus
 - Outils
 - Développement
 - Problèmes rencontrés
 - Déploiements
- 3 Apprentissage Statistique
 - Analyse de données
 - Base de données
 - Régression linéaire
 - Amélioration du Gradientboosting régression
- 4 Conclusion

Le programmatique est un modèle d'achat d'espace média automatisé, impression par impression, offrant la possibilité de cibler précisément des utilisateurs afin de leur adresser le bon message, au bon moment, au prix juste tout en permettant un contrôle complet sur la diffusion.

Dans le cadre du ciblage, il est essentiel de vérifier et sélectionner les sites vers lesquels nous souhaitons diffuser ou pas du contenu pour la sécurité de la marque (brand safety) et pour la visibilité.

Contexte

- En brand safety, il y a le commerce direct avec les listes d'exclusions et d'inclusions de sites ou mots clés.

Contexte

- En brand safety, il y a le commerce direct avec les listes d'exclusions et d'inclusions de sites ou mots clés.
- Une impression est considérée comme visible si elle s'est affichée sur le navigateur d'un internaute et si elle a eu l'occasion d'être vue.

Contexte

- En brand safety, il y a le commerce direct avec les listes d'exclusions et d'inclusions de sites ou mots clés.
- Une impression est considérée comme visible si elle s'est affichée sur le navigateur d'un internaute et si elle a eu l'occasion d'être vue.

Sommaire

- 1 Introduction
 - Contexte
 - Objectifs
- 2 Automatisation des processus
 - Outils
 - Développement
 - Problèmes rencontrés
 - Déploiements
- 3 Apprentissage Statistique
 - Analyse de données
 - Base de données
 - Régression linéaire
 - Amélioration du Gradientboosting régression
- 4 Conclusion

Les objectifs de notre stage sont de :

Objectifs

- Créer un outil d'automatisation pour automatiser la création et l'alimentation des canaux au niveau du plateforme Google Display & Video 360.

Objectifs

- Créer un outil d'automatisation pour automatiser la création et l'alimentation des canaux au niveau du plateforme Google Display & Video 360.
- Explorer les données de campagnes pour trouver une perspective de machine learning.

Objectifs

- Créer un outil d'automatisation pour automatiser la création et l'alimentation des canaux au niveau du plateforme Google Display & Video 360.
- Explorer les données de campagnes pour trouver une perspective de machine learning.

Sommaire

- 1 Introduction
 - Contexte
 - Objectifs
- 2 Automatisation des processus
 - Outils
 - Développement
 - Problèmes rencontrés
 - Déploiements
- 3 Apprentissage Statistique
 - Analyse de données
 - Base de données
 - Régression linéaire
 - Amélioration du Gradientboosting régression
- 4 Conclusion

Pour développer notre outil, nous utiliserons Google Sheet comme frontend, pour réaliser les actions dans Google Display & Video 360. Et Google Apps script pour le développement des fonctions qui réaliseront ces actions.

Objectifs

Dans Google Apps script, nous avons utilisés :

- Google Drive pour récupérer, modifier et stocker les listes sites d'exclusions ou d'inclusions et la liste des mots clés
- Google Cloud Plateforme est une plateforme de cloud computing fournie par Google, proposant un hébergement sur la même infrastructure que celle que Google utilise en interne pour des produits tels que son moteur de recherche

Sommaire

- 1 Introduction
 - Contexte
 - Objectifs
- 2 Automatisation des processus
 - Outils
 - Développement
 - Problèmes rencontrés
 - Déploiements
- 3 Apprentissage Statistique
 - Analyse de données
 - Base de données
 - Régression linéaire
 - Amélioration du Gradientboosting régression
- 4 Conclusion

Nous avons développé pour chaque type de listes exclusions ou inclusions et celle des mots clés des boutons qui gèrent les actions suivant :

- Lister les identifiants des canaux de partenaires et/ou annonceurs
- Remplacer les contenus des canaux
- Ajouter des éléments dans les canaux
- Supprimer des éléments dans les canaux
- Mettre à jour les noms des canaux
- Vérifier un élément dans une liste d'exclusion ou d'inclusion
- Mettre à jour le fichier d'une liste sur drive

Les étapes sur Apps script

Après avoir créer notre Google sheet, nous allons sur l'Apps script de ses extensions pour développer nos fonctions pour les boutons. Mais d'abord, il faut le configurer :

- Ajouter les champs d'applications des autorisations
- Ajouter l'identification du projet google Cloud
- Ajouter les bibliothèques

Sommaire

- 1 Introduction
 - Contexte
 - Objectifs
- 2 Automatisation des processus
 - Outils
 - Développement
 - **Problèmes rencontrés**
 - Déploiements
- 3 Apprentissage Statistique
 - Analyse de données
 - Base de données
 - Régression linéaire
 - Amélioration du Gradientboosting régression
- 4 Conclusion

Les principaux problèmes rencontrés sont ceux :

Problèmes rencontrés

- Maximum de temps d'exécution

Problèmes rencontrés

- Maximum de temps d'exécution
- Problèmes de serveurs internes

Sommaire

- 1 Introduction
 - Contexte
 - Objectifs
- 2 Automatisation des processus
 - Outils
 - Développement
 - Problèmes rencontrés
 - Déploiements
- 3 Apprentissage Statistique
 - Analyse de données
 - Base de données
 - Régression linéaire
 - Amélioration du Gradientboosting régression
- 4 Conclusion

Pour le déploiement de l'outil, nous devons :

Déploiements

- Déployer une version bibliothèque du projet Apps script

Déploiements

- Déployer une version bibliothèque du projet Apps script
- Créer un nouveau Google sheet en copiant le premier

Déploiements

- Déployer une version bibliothèque du projet Apps script
- Créer un nouveau Google sheet en copiant le premier
- Configurer son apps script en ajoutant les mêmes champs d'autorisations et bibliothèques, mais aussi celui de notre projet afin de créer les fonctions

Déploiements

- Déployer une version bibliothèque du projet Apps script
- Créer un nouveau Google sheet en copiant le premier
- Configurer son apps script en ajoutant les mêmes champs d'autorisations et bibliothèques, mais aussi celui de notre projet afin de créer les fonctions

Sommaire

- 1 Introduction
 - Contexte
 - Objectifs
- 2 Automatisation des processus
 - Outils
 - Développement
 - Problèmes rencontrés
 - Déploiements
- 3 Apprentissage Statistique
 - Analyse de données
 - Base de données
 - Régression linéaire
 - Amélioration du Gradientboosting régression
- 4 Conclusion

Grâce à BigQuery, nous avons connecté nos données de campagnes à un google sheet afin de les explorer et de créer une base de données.

Elle contient les éléments de campagne selon les environnements, nous avons comme variables :

Base de données

- Environnement : Connected Tv, Desktop, Mobile In-App, Mobile WebM

Base de données

- Environnement : Connected Tv, Desktop, Mobile In-App, Mobile WebM
- Catégorie d'annonceurs : Banque/Assurance, Santé, Automobile..

Base de données

- Environnement : Connected Tv, Desktop, Mobile In-App, Mobile WebM
- Catégorie d'annonceurs : Banque/Assurance, Santé, Automobile..
- Mode d'achat : Open Auction, Private Auction, Preferred Deal, Programmatic Guaranteep

Base de données

- Environnement : Connected Tv, Desktop, Mobile In-App, Mobile WebM
- Catégorie d'annonceurs : Banque/Assurance, Santé, Automobile..
- Mode d'achat : Open Auction, Private Auction, Preferred Deal, Programmatic Guarantee
- Coût média

Base de données

- Environnement : Connected Tv, Desktop, Mobile In-App, Mobile WebM
- Catégorie d'annonceurs : Banque/Assurance, Santé, Automobile..
- Mode d'achat : Open Auction, Private Auction, Preferred Deal, Programmatic Guarantee
- Coût média
- Impressions

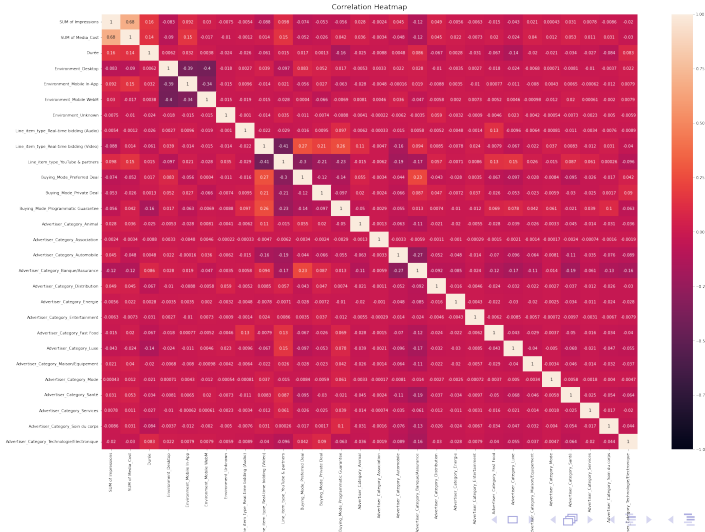
Base de données

- Environnement : Connected Tv, Desktop, Mobile In-App, Mobile WebM
- Catégorie d'annonceurs : Banque/Assurance, Santé, Automobile..
- Mode d'achat : Open Auction, Private Auction, Preferred Deal, Programmatic Guarantee
- Coût média
- Impressions
- Clicks, durée, mois du début et mois de la fin du campagne

Descriptions

	Environment	Advertiser_Category	Line_item_type	Buying_Mode	SUM of Impressions	SUM of Clicks	SUM of Media_Cost	Durée	Start month	End month
count	28140	28140	28140	28140	2.814000e+04	28140.000000	28140.000000	28140.000000	28140.000000	28140.000000
unique	5	16	4	4	NaN	NaN	NaN	NaN	NaN	NaN
top	Desktop	Banque/Assurance	Real-time bidding	Open Auction	NaN	NaN	NaN	NaN	NaN	NaN
freq	8928	9164	11545	19131	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	5.833945e+05	941.630135	1992.014087	39.280313	4.969225	4.969225
std	NaN	NaN	NaN	NaN	2.013497e+06	5111.412045	4951.349274	36.586322	2.955378	2.955378
min	NaN	NaN	NaN	NaN	1.000000e+00	0.000000	0.000003	1.000000	1.000000	1.000000
25%	NaN	NaN	NaN	NaN	3.065500e+03	0.000000	16.638375	15.000000	3.000000	3.000000
50%	NaN	NaN	NaN	NaN	5.917000e+04	5.000000	303.135593	28.000000	5.000000	5.000000
75%	NaN	NaN	NaN	NaN	3.950468e+05	188.000000	1733.038092	55.000000	7.000000	7.000000
max	NaN	NaN	NaN	NaN	6.553471e+07	324628.000000	156315.987100	353.000000	12.000000	12.000000

Corrélations



Modèles

Nous avons transformé nos variables catégoriques en vecteurs numériques avant de créer notre modèle.

Nous importons les modèles LinearRegression depuis `sklearn.linear_model` et GradientBoostingRegressor depuis `sklearn.ensemble`. Nous avons une base de données de 28140 échantillons que nous allons en deux groupe 90% pour les données d'apprentissage et 10% pour les données de test . Les scores sont calculés avec R2 score ou MSE.


```
GradientBoostingRegressor()  
  Training time: 3.047s  
  Prediction time: 0.004s  
  Explained variance: 0.7529275163206657  
  Mean absolute error: 313255.90754175285  
  score in train: 0.7444312476742341  
  R2 score : 0.7527785397820699  
  
LinearRegression()  
  Training time: 0.024s  
  Prediction time: 0.000s  
  Explained variance: 0.5805585191751279  
  Mean absolute error: 485310.331365866  
  score in train: 0.4887753771218155  
  R2 score : 0.5805277561632876
```

Amélioration du Gradientboosting régression

Nous recherchons son meilleur modèle suivant ses hyperparamètres. Pour cela, nous utilisons GridSearchCV de `sklearn.model_selection` qui nous sélectionne le modèle, il utilise la cross-validation. La sélection avec les hyperparamètres :

```
[19] parameters = {'loss':['squared_error', 'absolute_error', 'huber', 'quantile'],'n_estimators' : [75,100],  
                  'criterion' : ['friedman_mse','squared_error'],'learning_rate':[0.1,0.2]  
  
                  }  
grid = GridSearchCV(GradientBoostingRegressor(),parameters)  
model = grid.fit(M_train,m_train)  
print(model.best_params_,'\n')  
print(model.best_estimator_,'\n')  
  
{'criterion': 'squared_error', 'learning_rate': 0.2, 'loss': 'squared_error', 'n_estimators': 75}  
  
GradientBoostingRegressor(criterion='squared_error', learning_rate=0.2,  
                           n_estimators=75)
```

```
[21] GBR = GradientBoostingRegressor(criterion='squared_error', learning_rate=0.2,n_estimators=75).fit(M_train,m_train)
```

```
[22] m_pred=GBR.predict(M_test)
```

```
print("\tExplained variance:", explained_variance_score(m_test, m_pred))
```

```
print("\tMean absolute error:", mean_absolute_error(m_test, m_pred))
```

```
print("\tR2 score :", r2_score(m_test, m_pred))
```

```
Explained variance: 0.7705358717431527
```

```
Mean absolute error: 305962.9211397919
```

```
R2 score : 0.7703969689907411
```

Grâce l'outil d'automatisation, plus de vingt cinq équipes de GroupM dans le monde peuvent effectuer le management de leur brand safety. Et la partie apprentissage statistique peut être améliorer en augmenter les données au fur du temps.

Merci de votre attention !
Avez-vous des questions ?