Gestion automatique de Cookies

Maxence Neus

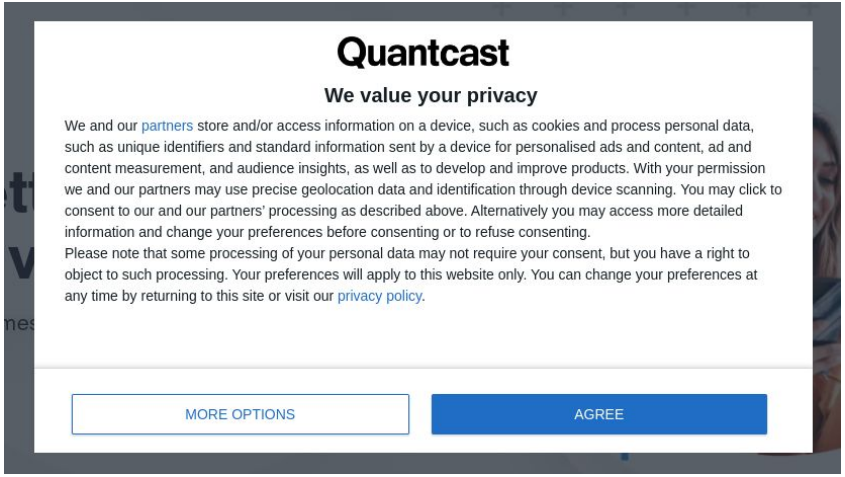
Tuteurs :
Jeremie Dequidt - Naif Mehanna - Walter Rudametkin

Sommaire

- ➔ Contexte
- ➔ Cahier des charges
- ➔ Processus de collecte de données
- ➔ Entraînement du modèle
- ➔ Démo
- ➔ Conclusion et critiques

Contexte

- Cadre de recherches web
  - Bannières cachent la page
  - Volonté de configurer les cookies

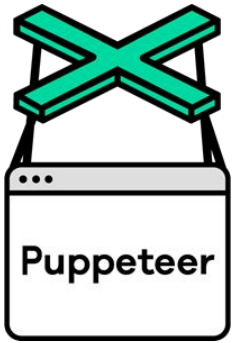


Contexte

- Catégories de cookies :
  - Mal défini légalement parlant
  - On propose d'utiliser la liste proposée par Consent-O-matic
    - Préférences et fonctionnalité
    - Performance et analytics
    - Stockage et accès à l'information
    - Sélection, transmission et reporting de contenu
    - Sélection, transmission et reporting de pubs
    - Autres

Crawler le web

Permet de visiter des sites automatiquement et d'exécuter des scripts sur la page



Cahier des charges

- Outil pour effectuer le paramétrage des cookies sur une page crawlée
  - extension chrome
  - Plugin puppeteer si le temps le permet
- Proposition d'une technologie Machine Learning (NLP) pour la classification des catégories

Processus de collecte de données

- Rappel de la soutenance intermédiaire

On parcours la liste des 1M sites les plus visités (tranco)

- Premier passage : On ne garde que les pages avec une bannière
  - On a utilisé les sélecteurs CSS pour détecter si une bannière est présente sur la page

- Second passage : On repasse manuellement sur les pages
  - On récupère les labels des catégories à la main

Processus de collecte de données

- **Résultats**

Catégorie	# de points
Performance And Analytics	14
Preferences And Functionality	7
Selection And Reporting Of Ads	18
Selection And Reporting Of Ads	3
Storage And Access To Information	4
Others	9

Entraînement du modèle
- Augmentation du dataset

- Notre dataset est bien trop petit pour obtenir un modèle correct en l'état.
- On a la possibilité d'augmenter le dataset en générant des variations des points :
  - En changeant des mots par leur synonymes
  - En changeant la position des mots dans la phrase
  - En changeant l'orthographe des mots
- Pour réaliser cette étape, on utilise un script décrit par Wei, Jason et Zou, Kai dans un papier de 2019, *EDA : Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks*

- On obtient finalement **540** points dans le dataset

Entraînement du modèle
- Préprocessing

- Pour pouvoir classifier par la suite, il faut convertir le dataset en valeurs numériques
  - On utilise l'état de l'art: BERT
  - On implémente pas manuellement l'algorithme, on utilise plutôt un modèle pré-entraîné (dans notre cas par Microsoft)

Entraînement du modèle
- Evaluation

- On peut utiliser le F1-score qui permet de quantifier la proportion de faux positifs et de faux négatifs pour donner un score à notre modèle
- L'accuracy correspond simplement au ratio des tests qui sont correctement classifiés

Modèle	Accuracy	F1-score
Hist Gradient Boosting	0.9444	0.9592
Gradient Boosting	0.9166	0.9118
Random Forest	0.9074	0.9071
Extra Trees	0.8981	0.9070

Entraînement du modèle
- Evaluation

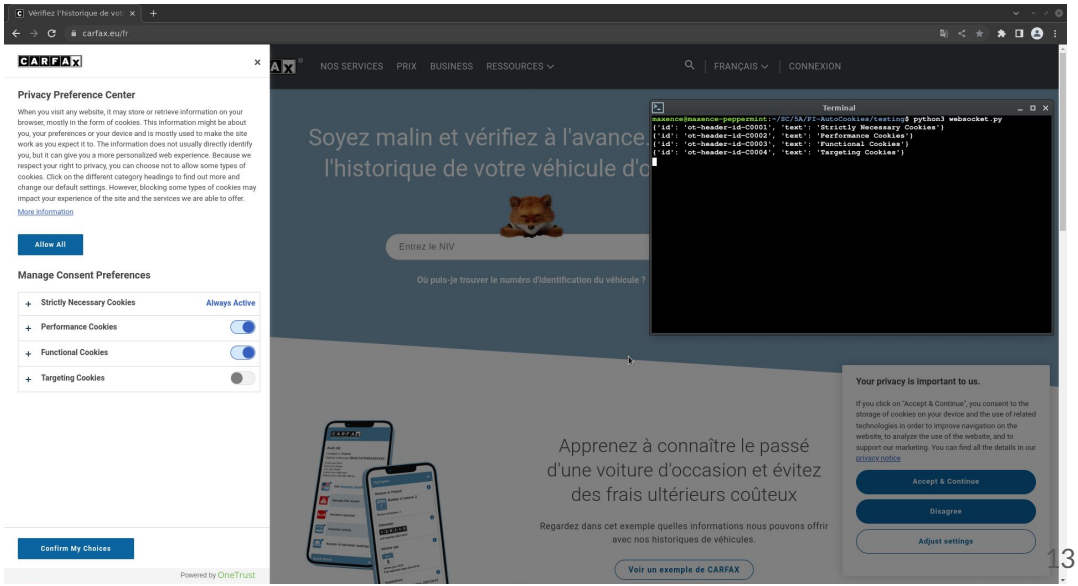
- On peut aussi montrer l'avantage d'augmenter le dataset : (on utilise ici l'algorithme Random Forest)

Dataset	Accuracy	F1-score
Augmenté	0.9074	0.9071
Non - Augmenté	0.57	

Démo

```
Connected to ws://localhost:8765/.
> {"id": "monid", "text": "Advertising Cookies"}
< {"id": "monid", "category": "SelectionAndReportingOfAds"}
Connection closed: 1000 (OK).
```

- Une API qui classifie le texte
- L'extension qui envoie le texte et interagit avec la page



Critiques du projet

- Pauvre communication de ma part

- Trop de temps passé à essayer d'avoir un dataset suffisant

  -> Trop peu d'avancées sur l'interaction avec les bannières

- Ordre des opérations peut-être amélioré