Open in app

up Sign In

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Détecter les bad buzz grâce au Deep Learning

Nasr-edine Drai · Follow
5 min read · Just now

Share

Introduction

Dans le cadre d'un projet pour le compte d'Air Paradis, une compagnie aérienne, j'ai été chargé de développer un prototype d'un produit IA permettant de prédire le sentiment associé à un tweet. Pour ce faire, j'ai testé trois approches différentes :

- Une approche "Modèle simple", pour développer rapidement un modèle classique (ex : régression logistique) permettant de prédire le sentiment associé à un tweet.

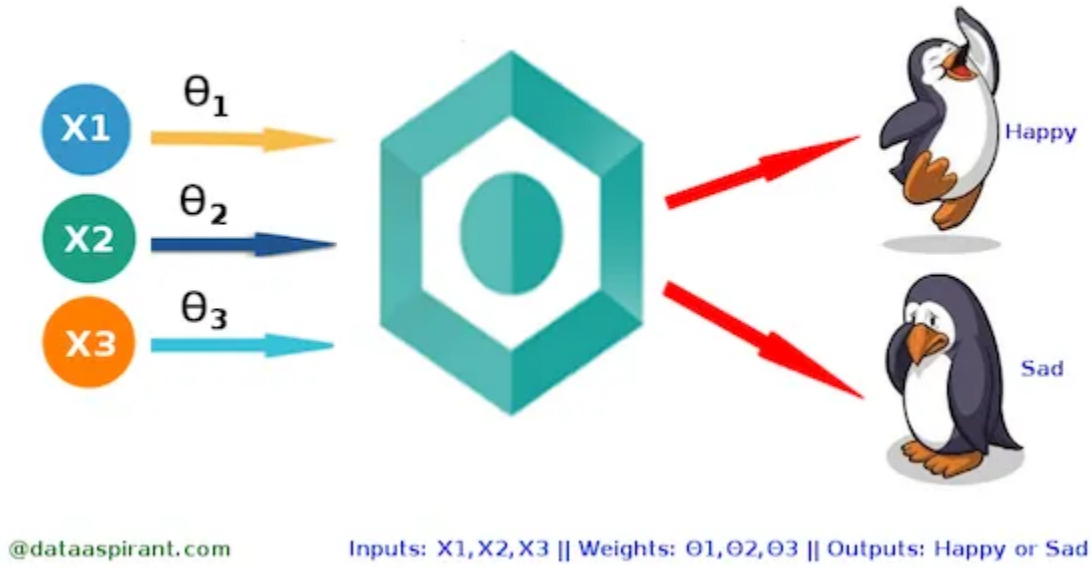
- Une approche "Modèle sur mesure avancé", pour développer un modèle basé sur des [cookie] ié à un tweet.

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

- Une approche "Modèle avancé BERT", pour développer un modèle basé sur un modèle pré-entraîné BERT, pour prédire le sentiment associé à un tweet.

Dans cet article, je présenterai les trois approches et je comparerai leurs performances. Je présenterai également la démarche orientée MLOps que j'ai mise en œuvre pour gérer le cycle de vie du modèle.

Modèle simple avec la régression logistique

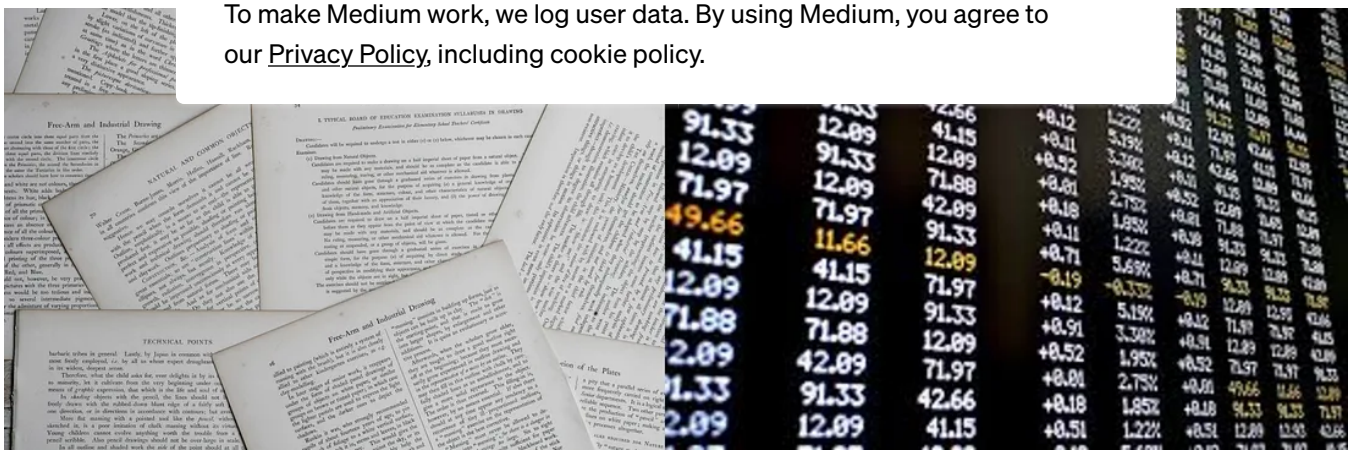


Prétraitement des données

La première étape a consisté à prétraiter les données des tweets, y compris le nettoyage du texte et la tokenisation. Le nettoyage du texte a consisté à supprimer les caractères non alphanumériques, les liens et les usernames. La tokenisation a consisté à diviser le texte en mots et expressions clés.

Extraction des caractéristiques

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.



Les caractéristiques des tweets ont été extraites à l'aide de la vectorisation TF-IDF. La vectorisation TF-IDF attribue un poids à chaque mot ou expression en fonction de sa fréquence d'occurrence dans le corpus et de sa rareté dans le corpus.

Entraînement du modèle

Le modèle de régression logistique a été formé sur les données d'entraînement. Le modèle a appris à prédire la classe d'un tweet en fonction de ses caractéristiques.

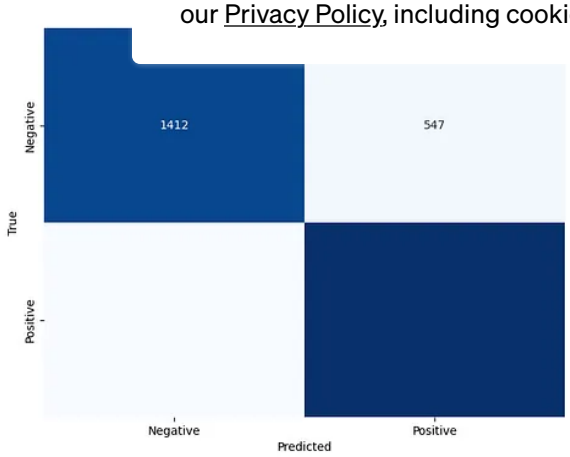
Évaluation du modèle

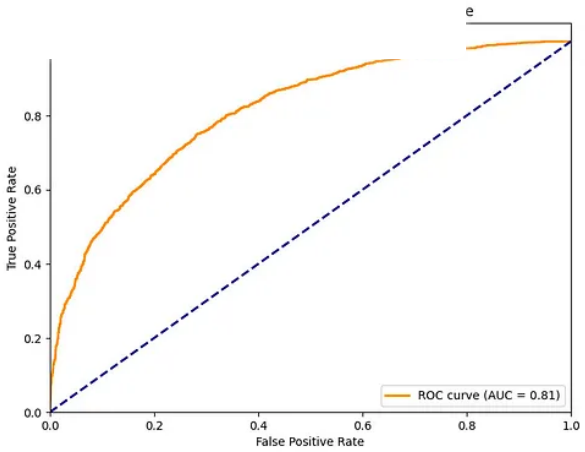
La performance du modèle a été évaluée sur les données de test. Le modèle a atteint une précision de 73,18 %, ce qui signifie qu'il a correctement classé 73,18 % des tweets.

Visualisation des performances du modèle

Les performances du modèle ont été visualisées à l'aide d'une matrice de confusion et d'une courbe ROC. La matrice de confusion montre le nombre de tweets correctement et incorrectement classés. La courbe ROC montre la capacité du modèle à distinguer les sentiments positifs des sentiments négatifs.

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

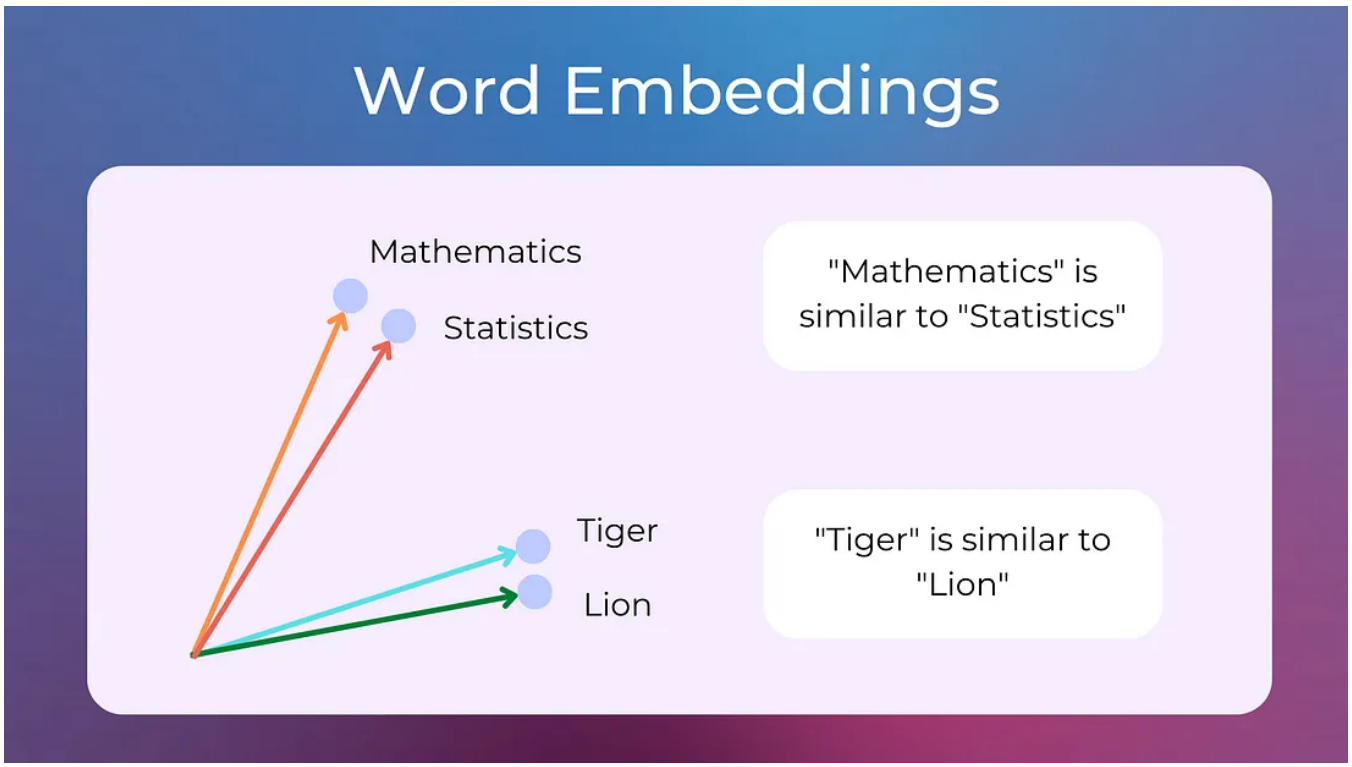




Modèle sur mesure avancé

Ce second model utilise deux techniques principales : les embeddings de mots GloVe et une couche LSTM.

Embeddings de mots GloVe

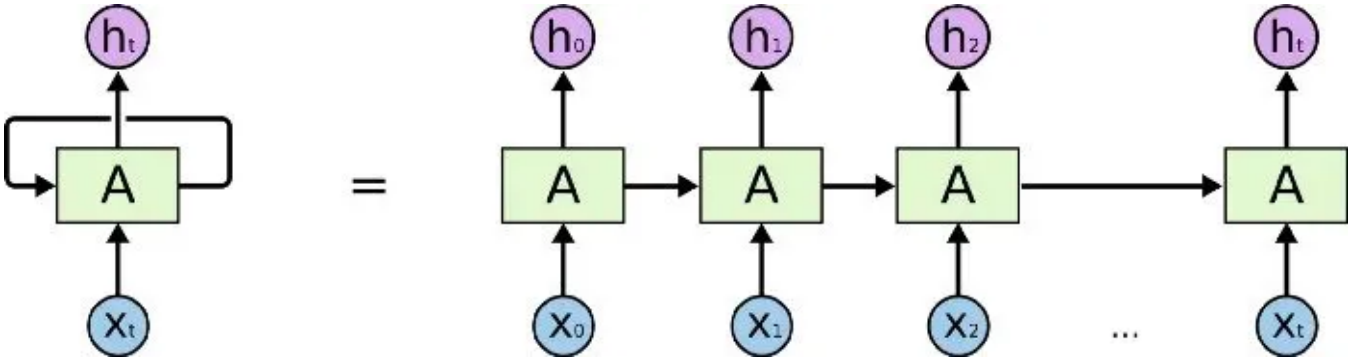


Les embeddings de mots GloVe sont des vecteurs pré-entraînés qui représentent les mots. Ils sont créés en analysant les cooccurrences de mots dans un corpus de texte massif. Les mots similaires ont des embeddings de mots similaires, ce qui permet au

modèle d'apprendre les relations sémantiques entre les mots

Couche LST

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.



Une couche LSTM est un type de réseau de neurones récurrent (RNN) qui est particulièrement bien adapté aux tâches de traitement du langage naturel. Les RNN sont des réseaux de neurones qui peuvent traiter des séquences de données, telles que des tweets. Les couches LSTM sont capables de conserver des informations à long terme, ce qui est important pour la classification du sentiment des tweets, car les sentiments peuvent être exprimés de différentes manières à différents endroits du tweet.

Comment les embeddings de mots GloVe et la couche LSTM sont utilisés dans le modèle

Utilisation des embeddings GloVe et de la couche LSTM : Les embeddings GloVe initialisent la couche d'embedding, convertissant les mots en vecteurs numériques pour que la couche LSTM apprenne les relations sémantiques. La couche LSTM prédit le sentiment des tweets en utilisant ces vecteurs. Ces techniques améliorent considérablement les performances du modèle, qui atteint une précision de 72,90 %.

Analyse de sentiment des tweets avec BERT

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

BERT, un modèle de langage pré-entraîné, peut être utilisé pour l'analyse de sentiment des tweets avec une grande précision.

Méthode

L'approche proposée se compose des étapes suivantes :

- Prétraitement des données : les tweets sont nettoyés et préparés pour l'analyse.
- Encodage des tweets : les tweets sont convertis dans un format que BERT peut comprendre.
- Entraînement du modèle : un modèle BERT est entraîné pour la classification de sentiment.
- Évaluation du modèle : les performances du modèle sont évaluées sur un ensemble de validation.

L'approche : aux résultats ol

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Utilisation d'Azure Web Apps et Github Action

J'utilise MLOps et Azure Web Apps pour déployer mon modèle d'analyse des sentiments pour les tweets.

Tout d'abord, j'ai entraîné un modèle d'analyse des sentiments à l'aide d'un modèle de langage pré-entraîné, tel que BERT. Ce modèle a été entraîné sur un ensemble de tweets, et il peut être utilisé pour l'analyse des sentiments.

Une fois le modèle entraîné, je l'ai déployé sur Azure Web Apps. Enfin, j'ai utilisé GitHub Actions pour automatiser le cycle de vie MLOps. Cela signifie que mon modèle est automatiquement déployé et testé chaque fois que je pousse des modifications à mon code.

Maintenant que mon modèle est déployé, je peux l'utiliser pour analyser les tweets en envoyant une requête POST au point de terminaison `/predict` de mon service Azure Web Apps. Le service retournera une réponse JSON avec le sentiment prédit du tweet.

Request: POST https://<your-azure-web-apps-service-url>/predict

```
{
 "tweet": "J'adore ce produit !"
}
```

Response:

```
{
  "sentiment": "Positif"
}
```

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Conclusic

Sur la base utres modèles. Cependant, il est important de noter que le temps d'entraînement et le temps de prédiction du modèle BERT sont plus longs que ceux des deux autres modèles.

Modèle	Précision
BERT	76,91%
Régression logistique	73,17%
GloVe avec couche LSTM	72,90%

Voici une comparaison plus détaillée des trois modèles :

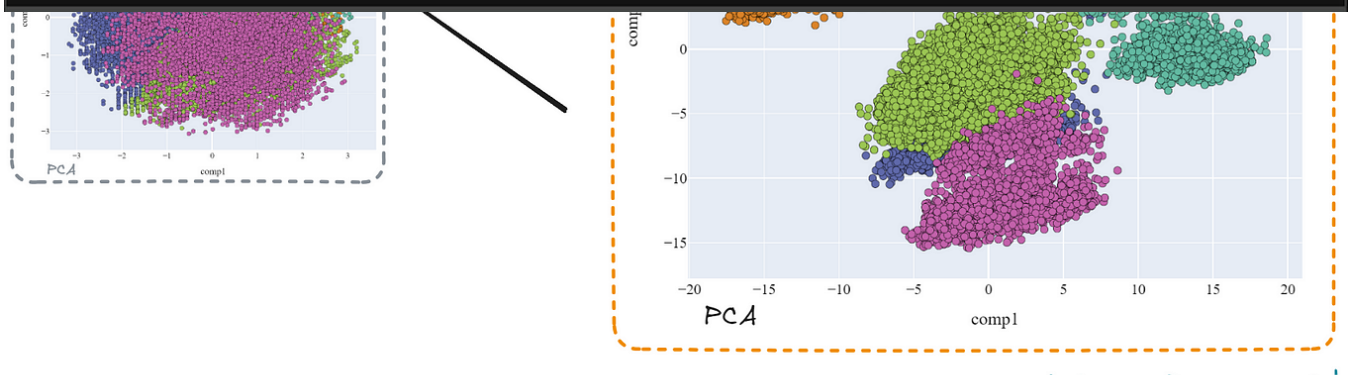
Follow

Written by Nasr-edine Drai

0 Followers

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Modèle		
BERT	Haute précision, peut apprendre les dépendances contextuelles	Long temps d'entraînement, long temps de prédiction
Régression logistique	Simple à entraîner et à interpréter, temps de prédiction rapide	Précision inférieure, ne peut pas apprendre les dépendances contextuelles
GloVe avec LSTM	Bonne précision, temps d'entraînement plus rapide que BERT	Temps de prédiction plus lent que la régression logistique, ne peut pas apprendre les dépendances contextuelles aussi bien que BERT



Damian Gil in Towards Data Science

Mastering Customer Segmentation with LLM

Unlock advanced customer segmentation techniques using LLMs, and improve your clustering models with advanced techniques

23 min read · 6 days ago

1.97K 21