Fiche tecnique

Sommaire

- Intégration avec Azure Insights pour la surveillance et le diagnostic
- Mise en place d'une règle d'alerte pour les feedbacks négatifs

1. Introduction

Le Machine Learning Operations (MLOps) est une approche qui vise à combler le fossé entre les équipes de développement et de production des modèles d'apprentissage automatique. Il s'agit de mettre en place des pratiques et des outils pour automatiser et faciliter le déploiement, la gestion, et la surveillance des modèles ML dans un environnement de production.

Dans cette fiche technique, nous présentons un projet d'analyse de sentiment sur les réseaux sociaux, en mettant en avant l'importance de cette analyse dans le contexte actuel. Nous abordons également la gestion de version avec Git, le traitement du texte, la modélisation avec différents algorithmes tels que Gaussian Naive Bayes, Régression Logistique, XGBoost, et des modèles Keras. Nous explorons également l'utilisation de MLFlow pour enregistrer les métriques des modèles et présentons le modèle BERT.

Enfin, nous décrivons l'API développée pour ce projet, les fonctions principales de l'API, l'interface utilisateur, et son intégration avec Azure Application Insight pour la collecte des feedbacks. Nous détaillons également le déploiement sur Azure, en utilisant des outils tels que Docker et Docker Hub, et l'intégration avec Azure Insights pour la surveillance et le diagnostic. Une règle d'alerte est mise en place pour les feedbacks négatifs, afin d'assurer une réactivité en cas de problème.

Ce projet met en lumière l'importance de l'approche MLOps pour garantir une gestion efficace des modèles ML en production, en combinant des pratiques d'ingénierie logicielle avec des techniques d'apprentissage automatique avancées.

Objectif du Projet

L'objectif principal du projet est de développer un modèle d'analyse de sentiment capable de classifier les tweets de l'entreprise Air Paradis en fonction de leur tonalité émotionnelle, qu'elle soit positive, négative ou neutre. Ce modèle sera entraîné sur un large corpus de tweets annotés, afin d'apprendre à reconnaître les différents schémas linguistiques associés à chaque type de sentiment. Une fois entraîné, le modèle sera intégré dans une API web, offrant ainsi une interface simple et intuitive pour les utilisateurs finaux.

2. Gestion de Version avec Git

Avant d'entamer le processus de développement et de déploiement de notre modèle d'analyse de sentiment et de son API sur le service cloud Azure, nous établissons une gestion de version robuste à l'aide de Git. Git est un système de contrôle de version distribué largement utilisé qui permet de suivre les modifications apportées au code source et de collaborer efficacement au sein de l'équipe de développement.

- **Initialisation du Référentiel Git**: Nous initialisons un référentiel Git pour notre projet, ce qui nous permet de suivre l'évolution du code source, des configurations et de la documentation tout au long du cycle de développement.
- **Suivi des Modifications**: À chaque étape du processus de développement, nous effectuons des validations régulières de notre code et des configurations, en les ajoutant et en les validant dans Git. Cela garantit que nous conservons un historique complet des modifications apportées au projet et nous permet de revenir à des versions antérieures si nécessaire.
- **Gestion des Conflits**: En cas de conflits lors de la fusion de branches ou de la collaboration avec d'autres membres de l'équipe, nous utilisons les fonctionnalités de résolution de conflits de Git pour harmoniser les modifications apportées par différentes parties et maintenir l'intégrité du code.

3. Traitement du Texte

Dans cette section, nous nous concentrons sur la préparation et le prétraitement des données textuelles pour l'analyse.

- **Nettoyage du Texte**: Il s'agit d'une étape cruciale pour préparer les données textuelles. Elle implique la suppression de caractères non pertinents tels que les caractères spéciaux et les chiffres, la conversion de tout le texte en minuscules pour assurer l'uniformité et l'élimination des espaces blancs inutiles.
- **Tokenisation**: Cette étape consiste à diviser le texte en unités plus petites appelées tokens, ce qui facilite l'analyse ultérieure.
- **Suppression des Mots Vides**: Les mots couramment utilisés mais peu informatifs, tels que "le", "la", "et", sont supprimés pour réduire la dimensionnalité des données.
- **Lemmatisation/Stemming**: Ces techniques réduisent les mots à leur forme de base pour améliorer l'efficacité des algorithmes d'apprentissage automatique.
- **Abréviations**: Les abréviations sont converties en leurs formes complètes pour une meilleure gestion du texte.

4. Modélisation

Nous construisons et évaluons des modèles d'apprentissage automatique pour l'analyse de texte.

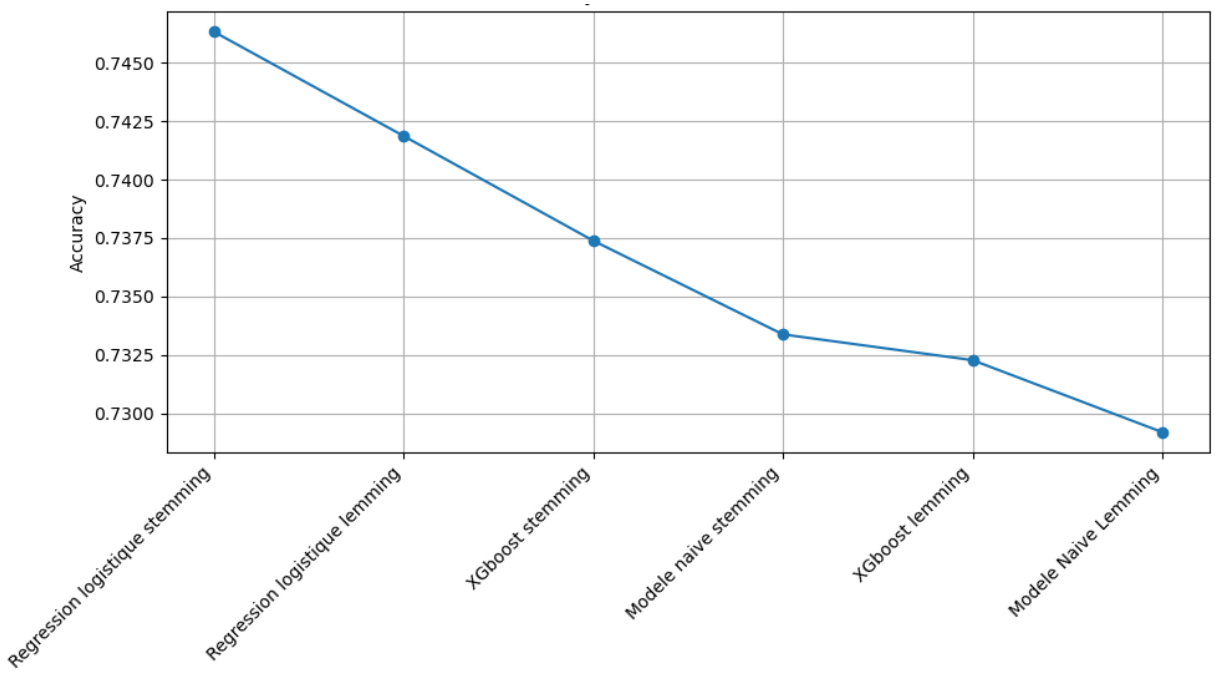
- **Division des Données**: Les données sont séparées en ensembles d'entraînement et de test pour éviter le surapprentissage.
- **Création du Modèle**: L'architecture du modèle est définie, y compris le nombre de couches et de neurones.
- **Entraînement du Modèle**: Les paramètres du modèle sont ajustés à l'aide de l'ensemble d'entraînement.
- **Évaluation du Modèle**: Les performances du modèle sont évaluées à l'aide de l'ensemble de test.
- **MlFlow**: Cet outil enregistre les métriques de chaque modèle pour permettre une comparaison efficace des performances.

Chaque itération d'entraînement et d'évaluation est effectuée à la fois sur un texte lemmatisé et un texte stemmatisé.

5. Modèles Simples

Nous commençons par la mise en place de trois modèles simples : GaussianNB, Régression Logistique et XGBoost.

- **Gaussian Naive Bayes (GaussianNB)** : Ce modèle repose sur le théorème de Bayes et suppose que les caractéristiques sont conditionnellement indépendantes. Il est souvent utilisé pour des données de classification et est efficace lorsque les caractéristiques sont continues.
- **Régression Logistique**: Il s'agit d'un modèle de régression utilisé pour prédire la probabilité qu'un résultat appartienne à une catégorie donnée. Il est largement utilisé dans la classification binaire et multiclasse.
- **XGBoost**: XGBoost est une implémentation de l'algorithme de boosting en gradient qui est efficace pour les problèmes de classification et de régression. Il combine plusieurs modèles faibles pour créer un modèle fort.



6. Modèles Keras

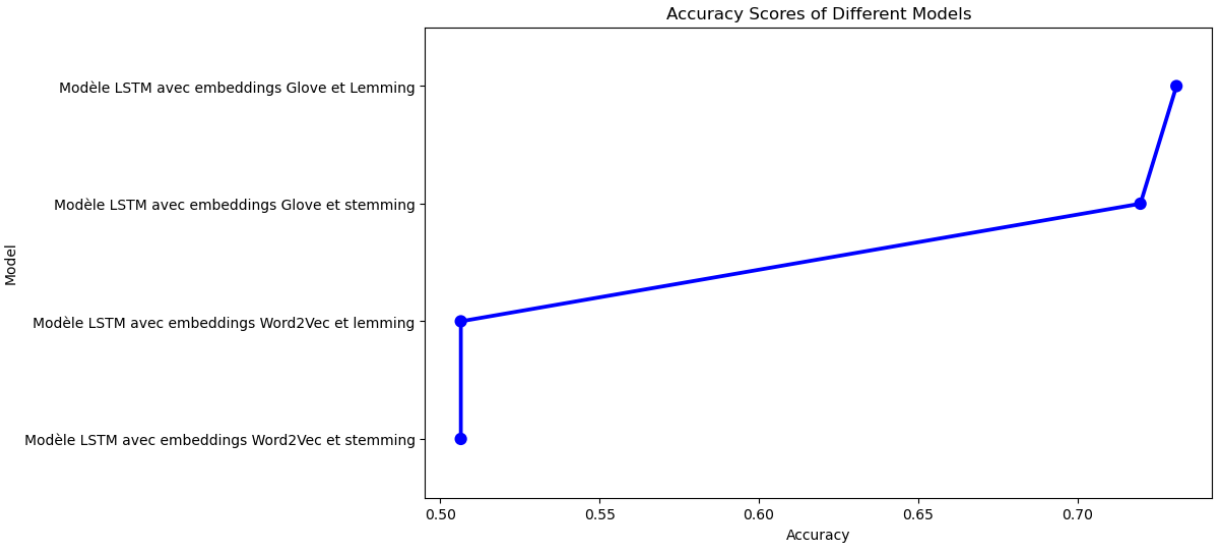
Nous poursuivons avec des modèles Keras utilisant une couche biLSTM pour mieux capturer les caractéristiques des textes. Les tests sont effectués avec les embeddings Word2Vec et GloVe.

Un modèle biLSTM est un type de modèle de réseau de neurones récurrents qui utilise deux couches de LSTM (Long Short-Term Memory) pour capturer les informations contextuelles des textes de manière bidirectionnelle. Cela permet au modèle de mieux comprendre les relations et les dépendances entre les mots dans une phrase.

Un **word embedding** est une technique utilisée en traitement automatique du langage naturel pour représenter des mots sous forme de vecteurs numériques. Ces vecteurs permettent de capturer la signification et la relation entre les mots, ce qui facilite la manipulation et l'analyse des textes.

Word2Vec et GloVe sont deux algorithmes populaires pour créer des embeddings de mots. Word2Vec est basé sur des modèles de réseaux de neurones qui apprennent à prédire un mot en fonction de son contexte. GloVe, quant à lui, utilise des statistiques de co-occurrence pour créer des embeddings en considérant la fréquence à laquelle les mots apparaissent ensemble.

Ces embeddings sont utilisés dans de nombreux modèles d'apprentissage automatique pour des tâches telles que la classification de texte, la traduction automatique et l'analyse de sentiment. Ils permettent d'améliorer la performance de ces modèles en capturant la sémantique des mots et en réduisant la dimensionnalité des données textuelles.



7. Modèle BERT

Le modèle BERT (Bidirectional Encoder Representations from Transformers) est un modèle de langage pré entraîné développé par Google. Il est capable de capturer les relations entre les mots dans une séquence de texte grâce à son architecture transformer bidirectionnelle. BERT a révolutionné de nombreux domaines du traitement du langage naturel en raison de sa capacité à produire des représentations contextuelles des mots.

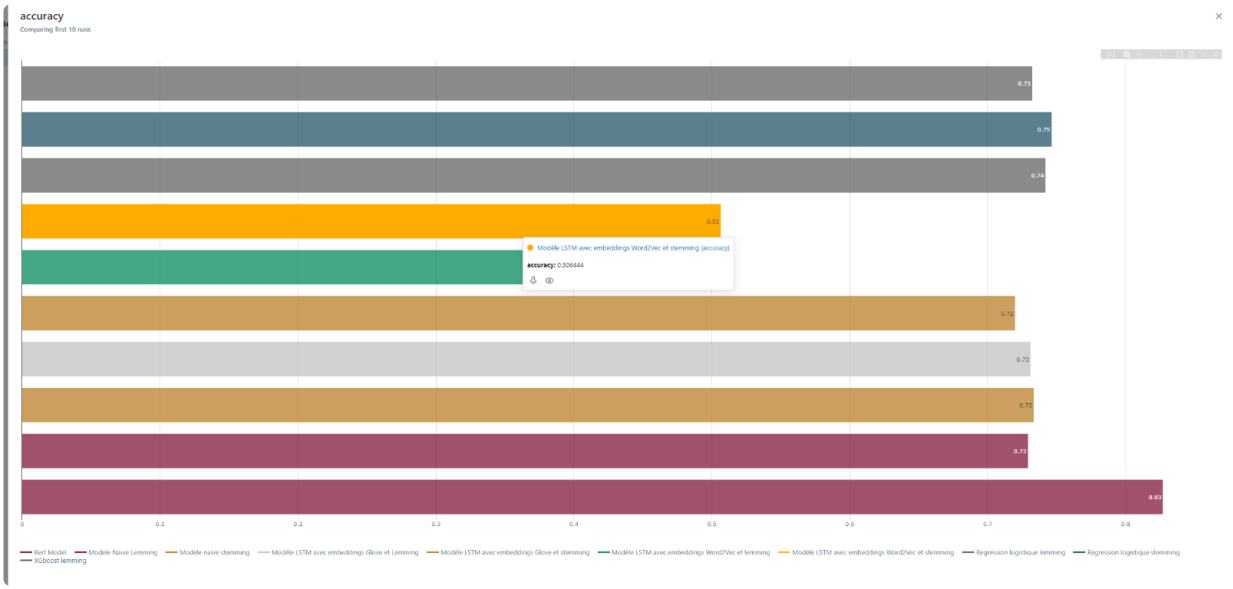
```
Epoch 1
Training loss: 0.34839468023180964
Validation loss: 0.39067043113708494
Accuracy: 0.827
```

8. MLFlow vue d'ensemble

Ici, vous pouvez voir une vue de l'exécution de tous les modèles dans l'interface utilisateur de MLFlow.

Table Chart Evaluation Experimental

									Metrics
		Run Name	Created	Dataset	Duration	User	Source	Version	Models	accuracy
		Bert Model	1 day ago	-	21.1min	Zacca	C:\Users...	-	pytorch	0.827
		Modele Naive Lemming	1 day ago	-	3.1s	Zacca	C:\Users...	-	sklearn	0.72918926...
		Modele naive stemming	1 day ago	-	2.9s	Zacca	C:\Users...	-	sklearn	0.73337501...
		Modèle LSTM avec em...	1 day ago	-	37.2min	Zacca	C:\Users...	-	tensorflow	0.73109358...
		Modèle LSTM avec em...	1 day ago	-	1.1h	Zacca	C:\Users...	-	tensorflow	0.71977365...
		Modèle LSTM avec em...	1 day ago	-	19.0min	Zacca	C:\Users...	-	tensorflow	0.50644445...
		Modèle LSTM avec em...	1 day ago	-	25.2min	Zacca	C:\Users...	-	tensorflow	0.50644445...
		Regression logistique le...	1 day ago	-	10.7s	Zacca	C:\Users...	-	sklearn	0.74186842...
		Regression logistique st...	1 day ago	-	6.4s	Zacca	C:\Users...	-	sklearn	0.74633239...
		XGboost lemming	1 day ago	-	14.6s	Zacca	C:\Users...	-	sklearn	0.73227152...
		XGboost stemming	1 day ago	-	14.5s	Zacca	C:\Users...	-	sklearn	0.73738570...



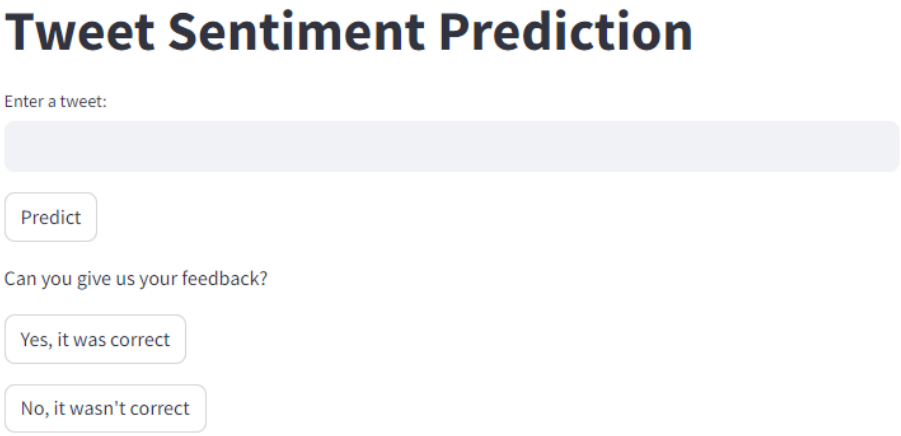
9. API

Nous avons développé une API pour permettre l'utilisation facile du modèle d'analyse de sentiment. Cette API est construite en utilisant Streamlit comme base.

Fonctions Principales:

- `preprocess(text)` : Cette fonction prend en entrée un texte et le nettoie en supprimant les liens, les mentions d'utilisateurs, les caractères spéciaux et les chiffres. Elle lemmatise ensuite les mots et supprime les mots vides.
- `predict()` : Cette fonction prend en entrée un tweet, le prétraite, le tokenize et le pad, puis utilise le modèle pour prédire le sentiment du tweet. Elle convertit ensuite la prédiction en 'POSITIVE' ou 'NEGATIVE'.

Interface Utilisateur:



L'interface utilisateur permet à l'utilisateur d'entrer un tweet, puis affiche la prédiction du sentiment du tweet. Elle demande ensuite à l'utilisateur si la prédiction était correcte et enregistre le tweet, la prédiction et les commentaires de l'utilisateur sont donc envoyés vers Azure Application Insight.

10. Deployment Azure

Azure

Azure est une plateforme de cloud computing proposée par Microsoft. Elle offre une gamme de services, notamment le stockage de données, le calcul, l'intelligence artificielle, l'IoT et bien d'autres. Azure permet aux utilisateurs de déployer et de gérer des applications sur un réseau mondial de centres de données gérés par Microsoft. Il offre également des outils de développement, de déploiement et de gestion d'applications, ainsi que des services de sécurité et de conformité.

Docker et Docker Hub:

Docker est une plateforme open source permettant de créer, de déployer et de gérer des applications dans des conteneurs. Les conteneurs Docker permettent d'emballer une application avec toutes ses dépendances dans une unité standardisée pour le développement logiciel. Docker Hub est un service cloud permettant aux développeurs de stocker et de partager des conteneurs Docker. Il fournit également des outils pour l'automatisation du pipeline de développement.

Azure Insights:

Azure Insights est un service de surveillance et de diagnostic pour les applications déployées sur Azure. Il collecte et analyse les données de télémétrie pour fournir des informations sur les performances et l'utilisation des applications. Azure Insights offre des fonctionnalités telles que la surveillance des performances, le suivi des logs, l'analyse des erreurs et des alertes personnalisées pour aider les développeurs à diagnostiquer et à résoudre les problèmes rapidement.

Les étapes suivies

Le déploiement de l'API se fait via le service offert par Microsoft Azure.

Tout d'abord, nous dockerisons l'API, ce qui permet de créer un environnement idéal dans lequel nous sommes certains que notre API fonctionnera correctement.

Ensuite, nous chargeons l'API sur DockerHub, une plateforme cloud qui permet aux développeurs de stocker et de partager des conteneurs Docker.

Enfin, nous appelons l'API DockerHub sur notre plateforme de déploiement dans le cloud, Azure. Là, nous pouvons utiliser une URL fournie par la plateforme pour accéder à l'API. Les résultats générés par l'API sont envoyés vers Azure Insights, un service de surveillance et de diagnostic qui collecte et analyse les données de télémétrie. Cela nous permet d'analyser les performances et l'utilisation de notre API en temps réel, ce qui est essentiel pour assurer son bon fonctionnement et son évolutivité.

L'ajout d'une regle qui permet de recevoir des allertes par mail tous le fois que les feedback des utilisateurs sont négatives, nous permet d'etre tres rapide dans la résolution des problematiques qui peuvent arriver.

Les etapes suivies

Le déploiement de l'API se fait via le service offert par Microsoft Azure.

Tout d'abord, nous dockerisons l'API, ce qui permet de créer un environnement idéal dans lequel nous sommes certains que notre API fonctionnera correctement.

Ensuite, nous chargeons l'API sur DockerHub, une plateforme cloud qui permet aux développeurs de stocker et de partager des conteneurs Docker.

Enfin, nous appelons l'API DockerHub sur notre plateforme de déploiement dans le cloud, Azure. Là, nous pouvons utiliser une URL fournie par la plateforme pour accéder à l'API. Les résultats générés par l'API sont envoyés vers Azure Insights, un service de surveillance et de diagnostic qui collecte et analyse les données de télémétrie. Cela nous permet d'analyser les performances et l'utilisation de notre API en temps réel, ce qui est essentiel pour assurer son bon fonctionnement et son évolutivité.

L'ajout d'une règle qui permet de recevoir des alertes par mail chaque fois que les feedbacks des utilisateurs sont négatifs nous permet d'être très réactifs dans la résolution des problématiques qui peuvent survenir.