Projet 7 Openclassrooms

Réalisez une analyse de sentiments grâce au Deep Learning

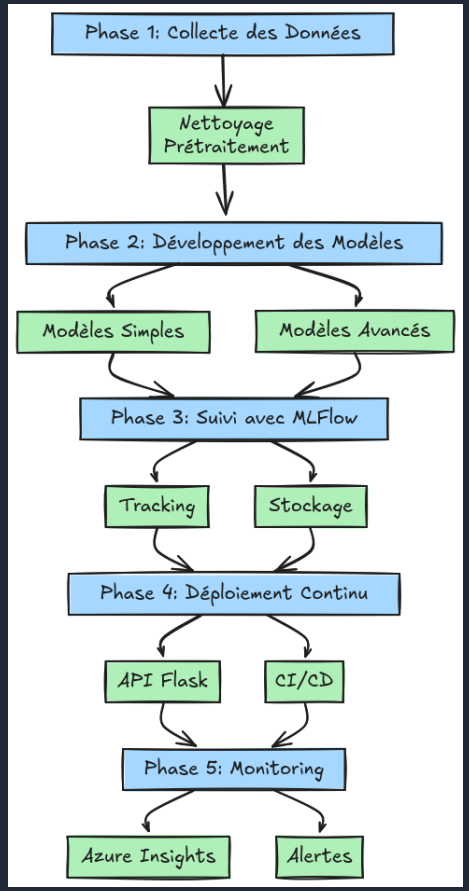
I. Présentation du contexte projet

Contexte Projet

- L’entreprise **Air Paradis** souhaite anticiper les bad buzz sur les réseaux sociaux
- **Problématique :** La compagnie a connu des retours négatifs sur les réseaux sociaux et souhaite pouvoir les détecter rapidement.

Notre mission :
Développer un prototype d'IA pour la détection automatique du sentiment des tweets permettant d'identifier en amont les retours négatifs potentiels.

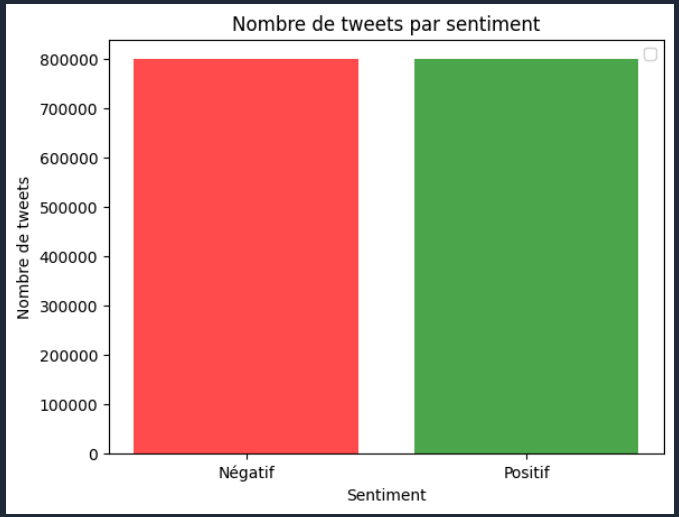
Présentation démarche Projet



II.Analyse et prétraitement du jeu de données

Présentation du jeu de donnée

- Jeu de donnée contenant **160000 tweets** contenant :
  - Le sentiment du tweet (négatif ou positif)
  - Le texte du tweet
  - L'utilisateur du tweet



Distribution des sentiments des tweets

Distribution équilibrée du nombre de tweet positif et négatif

Nettoyage et pré-traitement du texte

- Supression des textes vides
- Mise en forme du texte (minuscule)
- Supression de certains symbole :
  - Suppression des urls
  - Suppression des stopwords
  - Supression des caractères spéciaux

Export du CSV avec les textes nettoyés avant entrainement

III.Présentation des approches d'embbeding et classification pour l'analyse de sentiment

Qu’est-ce qu’un Modèle d’Embedding ?

- **Définition** : Un modèle d’embedding transforme des mots ou des phrases en vecteurs numériques, capturant les relations sémantiques entre les termes exploitable par des algorithmes de machine learning.
- **Modèles utilisés** :
  - TFIDF : Méthode statistique basée sur la fréquence d’apparition des mots, sans prise en compte du contexte sémantique.
  - Word2Vec : Transforme les mots en vecteurs en regardant leur contexte proche, avec les méthodes Skip-gram et CBOW…
  - GloVe : Crée des vecteurs en analysant combien de fois les mots apparaissent ensemble dans un texte.
  - DistilBERT : Version simplifiée de BERT, qui comprend le sens des mots en utilisant leur contexte dans une phrase.

Qu’est-ce qu’un Modèle de Classification ?

- **Définition** : Un modèle de classification attribue une étiquette (positif, négatif) à un texte en se basant sur ses caractéristiques.
- **Modèles utilisés** :
  - Régression Logistique : Modèle linéaire de classification binaire qui estime la probabilité qu’une observation appartienne à une classe donnée.
  - Random Forest : Combine plusieurs arbres de décision pour améliorer la précision et éviter le surapprentissage.
  - LightGBM : Algorithme qui crée des arbres de décision rapidement. Il est efficace pour les grandes bases de données

Modèle embedding --> transformation des tweets en représentation numérique --> **Modèle classification** entrainement prédiction du sentiment associé à chaque tweet

Résultats des Modèles - Analyse de Sentiment

Nom de l'exécution	Durée	Précision
TFIDF_LogisticRegression	5.0s	77.06%
Word2Vec_LightGBM	19.7s	72.86%
Word2Vec_LogisticRegression	8.3s	72.52%
TFIDF_LightGBM	23.7s	72.11%
DistilBERT_LogisticRegression	9.0s	71.66%
DistilBERT_LightGBM	19.6s	71.39%
GloVe_LightGBM	19.9s	71.00%
TFIDF_RandomForest	38.7s	70.76%
Word2Vec_RandomForest	20.8min	70.71%
GloVe_LogisticRegression	7.5s	70.54%
DistilBERT_RandomForest	20.4min	69.15%
GloVe_RandomForest	20.8min	68.88%

Choix du modèle Distilbert+Regression Logistique

IV. Du développement au déploiement

Démarche MLOps : Du Développement au Déploiement

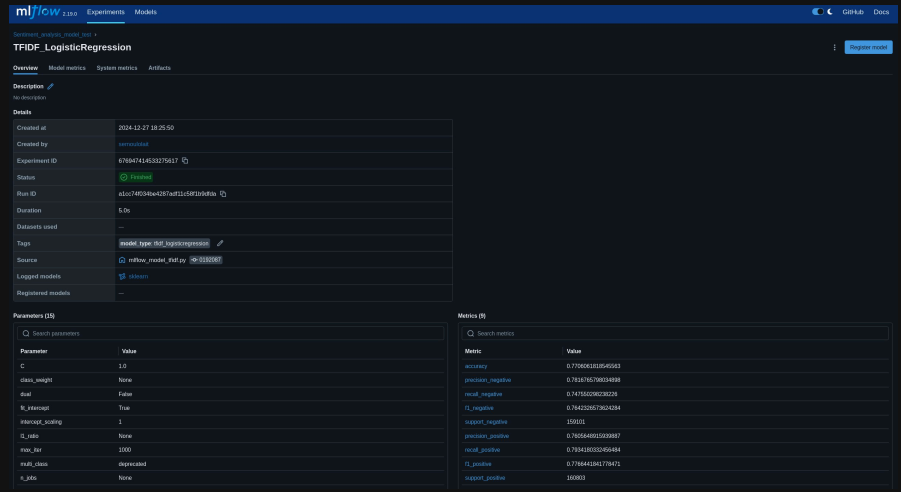
Principe du MLOps

- **Définition** : Le MLOps (Machine Learning Operations) est un ensemble de pratiques qui vise à déployer et maintenir des modèles de machine learning en production de manière efficace et automatisée.
- **Objectifs** :
  - Automatiser le cycle de vie des modèles ML.
  - Assurer la reproductibilité des expériences.
  - Surveiller et améliorer les performances des modèles en production.

Intégration de MLflow dans une démarche MLOps

1. **Suivi des expériences** : Enregistrement systématique des paramètres, des métriques et des artefacts pour chaque expérience, assurant une traçabilité complète.
2. **Gestion des versions** : MLflow permet de versionner les modèles, les données et le code.
3. **Surveillance et gestion** : MLflow Registry offre des outils pour surveiller les performances.

- Suivi des expériences et suivi des performance des modèles
- Enregistrement du modèle réintégrer pour intégration à l'application Flask



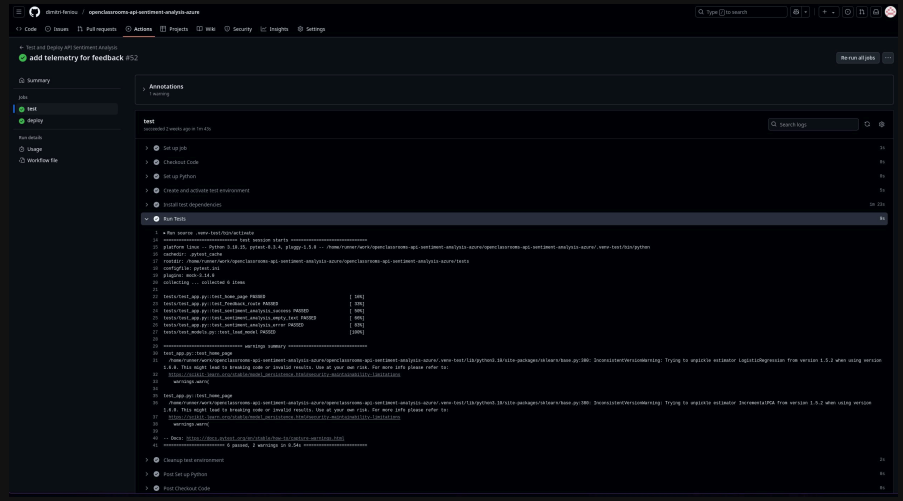
	Run Name	Created	Duration	Source	Models	Metrics accuracy
	TFIDF_LogisticRegression	5 days ago	5.0s	mlflow_model_tfidf.py	sklearn	0.7706061818545...
	Word2Vec_LightGBM	5 days ago	19.7s	mlflow_model_word2vec.py	sklearn	0.7285998299489...
	Word2Vec_LogisticRegression	5 days ago	8.3s	mlflow_model_word2vec.py	sklearn	0.7251956837051...
	TFIDF_LightGBM	5 days ago	23.7s	mlflow_model_tfidf.py	sklearn	0.7211382164649...
	DistilBERT_LogisticRegression	3 days ago	9.0s	mlflow_distilbert.py	sklearn	0.7166274882464...
	DistilBERT_LightGBM	3 days ago	19.6s	mlflow_distilbert.py	sklearn	0.7139173001900...
	GloVe_LightGBM	5 days ago	19.9s	mlflow_model_glove.py	sklearn	0.7100942782834...
	TFIDF_RandomForest	5 days ago	38.7s	mlflow_model_tfidf.py	sklearn	0.7076654246273...
	Word2Vec_RandomForest	5 days ago	20.8min	mlflow_model_word2vec.py	sklearn	0.7071308892667 8
	GloVe_LogisticRegression	5 days ago	7.5s	mlflow_model_glove.py	sklearn	0.7054960238071...
	DistilBERT_RandomForest	3 days ago	20.4min	mlflow_distilbert.py	sklearn	0.6915887266179...
	GloVe_RandomForest	5 days ago	20.8min	mlflow_model_glove.py	sklearn	0.6887878863659...

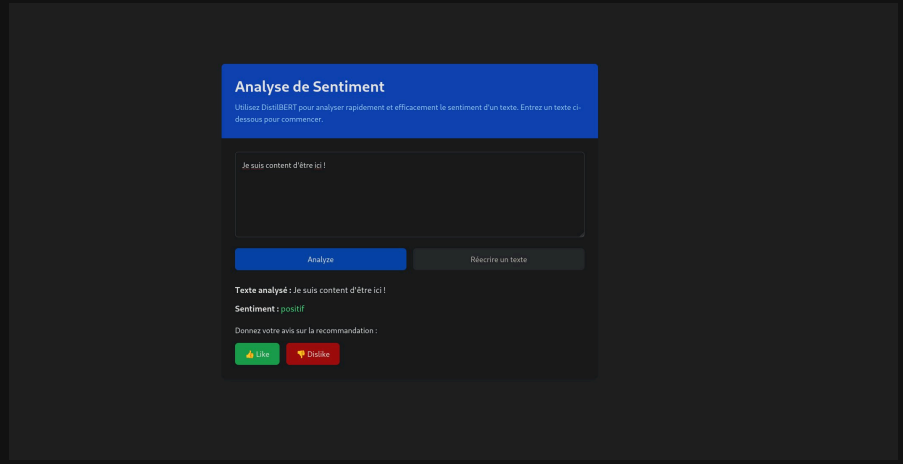
Déploiement en Production

Déploiement Automatisé

1. Déploiement automatique via GitHub Actions
2. Mise en place tests pour l'application
3. Hébergement de l'application Flask sur Azure App Service

Mise en place d'un pipeline CI/CD (Intégration Continue/Déploiement Continu)





Suivi de la Performance

Suivi en Production (Azure Application Insights)

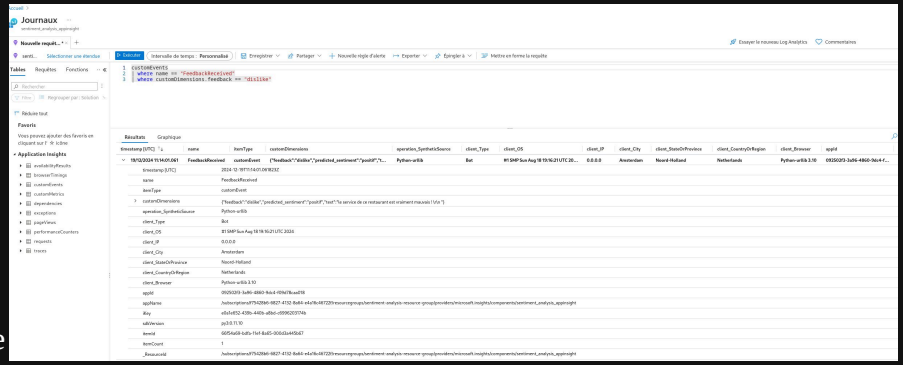
1. Collecte des prédictions incorrectes.
2. Utilisation de teletry azure app insight pour renvoyer les feedback
3. Alerte automatique (3 erreurs en 5 minutes).
4. Analyse et amélioration continue du modèle.

```python
# Enregistrement de la télémétrie avec Azure Application Insights
from applicationinsights import TelemetryClient

tc = TelemetryClient('e0a1e652-439b-440b-a8bd-c6996203174b')
tc.track_event(
    'FeedbackReceived',
    properties={
        'feedback': feedback,
        'text': text,
        'predicted_sentiment': predicted_sentiment
    }
)
tc.flush()

return redirect("/")
```

Amélioration Continue

Optimisation et Réentraînement possible

1. Identifier les erreurs grâce au feedback
2. Réentraînement du model en intégrant dans notre jeu donnée les feedbacks
3. Evaluation du modèle pour voir l'impact des feedback sur le modèle



V.Conclusion du Projet

Conclusion

- Choix du modèle DistilBERT pour la production
- Mise en oeuvre continue des principes MLOPS pour la mise en place d’une application d’analyse de sentiment sur le cloud Azure