Ingénierie IA

MIC
IA Consulting

Kindi BALDE

Détectez les Bad Buzz grâce au Deep Learning

Présentation des données

Problématique métier

Avis Paradis est une compagnie aérienne. Elle souhaite créer un produit IA permettant de détecter les bad buzz.

Pour un tweets sur les réseaux sociaux la concernant, elle veut déterminer rapidement le sentiment du tweet pour pouvoir de son côté ajuster sa politique marketing et e-réputation

Mission

Notre mission est de faire mettre en place un produit IA permettant de détecter le sentiment d'un tweet

MIC
IA Consulting

sommaire

1. Garbage-in garbage-out
2. Techniques de traitement de texte
3. Modèle API sur étagère
4. Modèle sur mesure simple
5. Modèle avancé
6. Démo fonctionnelle avec Gradio

1. Garbage-in Garbage-out

Application de traitements spécifiques pour nettoyer les données textuels

- Suppression des hashtags, @ , …
- Suppression des URLs
- Correction de l’orthographe
- Suppression des ponctuations
- Suppression des emojis
- Suppression des mots redondants (les “stopwords”)

2. Techniques de pré traitement de texte

Techniques pré traitement de texte

1. Tokenization ou le fait de découper les phrases ou paragraphes d'un texte en liste de mots. Cette technique est souvent implémentée pour avoir la main ensuite pour faire des traitement spécifique sur chaque mots. elle est d'ailleurs faite avant le Lemmatisation ou le Stemming.
2. Séquençage, ou le fait représenter chaque mot d'un corpus en une séquence de chiffres. Un passage obligé pour toute analyse de mot ou groupe de mots par une machine.
3. Stemming ou le fait de prendre une liste "tokenizé" et d'enlever le suffixe des mots qui en ont. Cette technique de normalisation, permet de rendre le modèle plus généralisable, notamment sur les nouveaux mots.
4. Lemmatisation comme le stemming mais va plus loin en ramenant chaque mot à sa racine lexicale. il supprime le suffixe et le préfixe des mots en prenant en compte leurs racines dans le dictionnaire (verbe, nom, adjectif, etc...)

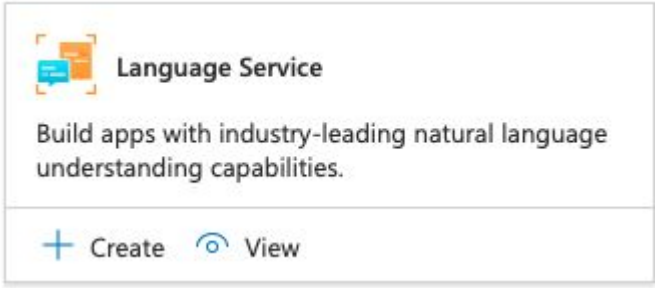
2. Techniques de pré traitement de texte

Embedding, une représentation vectorielle des mots dans une matrice à dimension plus petite où chaque mot est représenté par un vecteur. Dans cette dimension, chaque mot a une signification car il prend en compte sa relation (proximité ou non) avec les autres mots du corpus.

L'embedding ou le processus de plongement des mots se fait comme suit :

1. télécharger un word embedding déjà pré entraîné (ici le Wor2vec et le Glove)
2. commencer par une phase de tokenization et de séquençage des mots du vocabulaire (de notre dataset) en retournant un dictionnaire de word index (dont l'indexation dépend de la taille du vocabulaire, ici 20000 mots)
3. faire la projection en extrayant pour chaque mot indexé de notre vocabulaire, un vecteur de représentation dans la nouvelle dimension.
4. Nous avons ainsi une matrice embedding à utiliser dans notre modèle.

3. API sur étagère



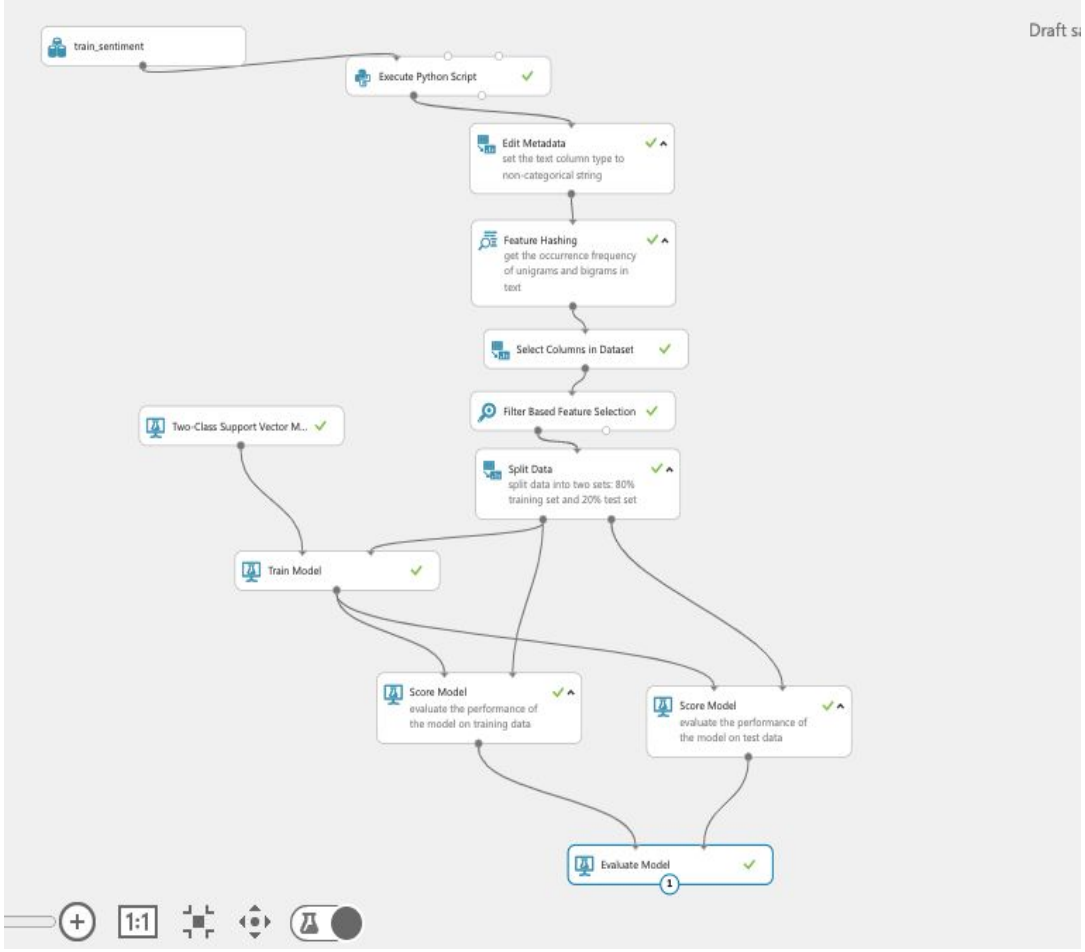
Utilisation Language service de MSFT Azure
Un service parmi les services cognitifs d'Azure.
Il permet de déployer un endpoint (un point de terminaison) qui appelé, permet de rapidement avoir une réponse. Dans notre cas le sentiment d'un tweet.

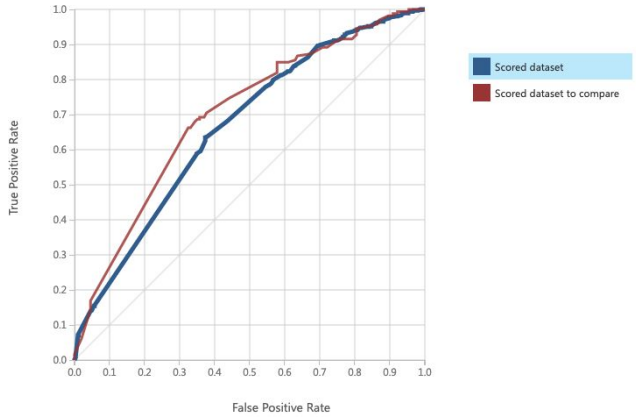
Avantages :
- accessibilité et rapidité d'implémentation
- intégration rapide à une solution
- utilisation comme première approche avant d'aller vers la mise en place d'un modèle plus complexe (Proof Of Concept)

inconvénients :
- service non personnalisable, car non accessibilité des paramètres du modèle.
- Non transparence sur les données d'entrainement du modèle (corpus de données pourrait être biaisés - race - sexe)

Neutral sentiments :
- filtrer les tweets à sentiments neutres
- les cas où le sentiment est neutre prendre le max score des deux autres sentiments (positive et négative)

4. Modèle sur mesure simple





True Positive	False Negative	Accuracy	Precision	Threshold	AUC
526	140	0.621	0.604	0.5	0.666

False Positive	True Negative	Recall	F1 Score
345	269	0.790	0.684

Positive Label	Negative Label
1	0

MIC
IA Consulting

4. Modèle sur mesure simple avec Azure ML Studio

Azure ML studio offre la possibilité de mettre en place des modèle de machine learning fonctionnelle et pratique. Il n'est pas nécessaire de coder.

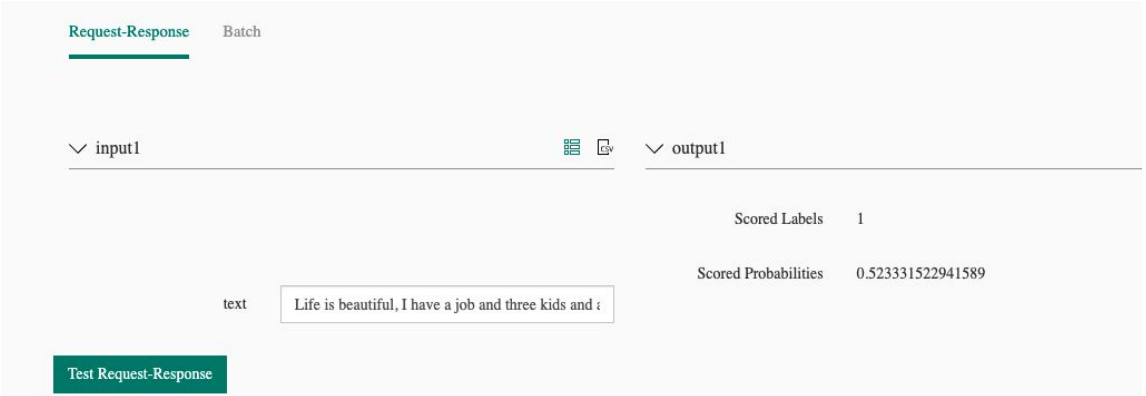
Avantages

les avantages sont multiples :

- facilité de création d'un modèle de Machine Learning spécifique.
- facilité de testing
- facilité de déploiement en web service

Inconvénients

- fonctionnalité ML pour le Deep learning inexistantes



5. Modèle avancé

5.1 Elaboration du modèle

Nous avons implémenté plusieurs modèles de deep learning pour la classification de texte.

Bidirectional LSTM avec own embedding et sans lemmatization, ni stemming (modèle de base)

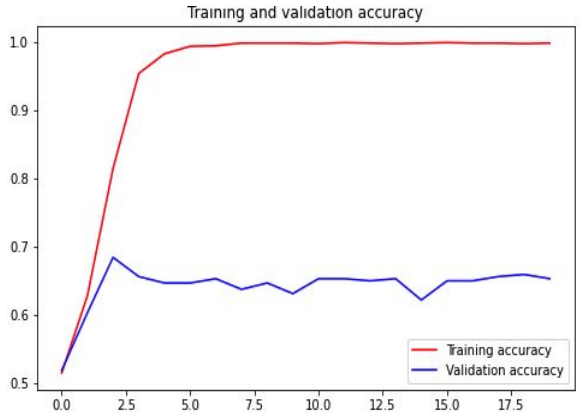
Bidirection LSTM avec utilisation d'une matrice de plongement extraite de Glove et un prétraitement du texte par le Lemmatizing et le stemming

Bidirection LSTM avec utilisation d'une matrice de plongement extraite de Word2Vec et un prétraitement du texte par le Lemmatizing et le stemming

Le modèle BERT pour la classification de texte "bert uncased" avec un prétraitement du texte par le Lemmatizing et le stemming

Hugging Face

	MPODEL_ACCURACY
TEXT_ANALYTICS_NEUTRAL_FILTERED	0.681706
TEXT_ANALYTICS_NEUTRAL_TRANSFORMED	0.729375
BID_LSTM	0.652500
BERT_LEMMATIZING	0.498125
BERT_STEMMING	0.498750
BID_LSTM_GLOVE_LEMMATIZING	0.476250
BID_LSTM_GLOVE_STEMMING	0.476250
BID_LSTM_W2V_LEMMATIZING	0.476250
BID_LSTM_W2V_STEMMING	0.476250



5. Modèle avancé

5.2 Entraînement sur Azure

Il existe plusieurs façon d'entraîner un modèle sur Azure.
Nous entraîner notre modèle suivant les étapes suivantes :

1. Création d'un workspace : un espace de travail sur Azure. La porte d'entrée pour toute création de projet sur Azure.
2. préparation du modèle à entraîner dans un script PY et d'un fichier YML pour les librairies nécessaires pour le modèle.
3. création des ressources de calcul : nous avons créé 4 noeuds (instances) en mode cluster pour exécuter nos jobs sur Azure.
4. création d'une expériment : porte d'entrée pour exécuter un modèle un job sur Azure.
5. configuration d'une image de containerisation de notre job
6. Le job ML sera ainsi soumis en provisionnant un container avec les scripts du modèle et ses configurations.

5. Modèle avancé

5.2 Mise en production sur Azure

Une fois l'entraînement terminé et les poids du modèle récupérés en local, nous pouvons déployer le modèle en "production" sur Azure.

- Création d'un script PY d'inférence du modèle (score.py)
- Enregistrement ou création du modèle sur Azure ML (experiment)
- préparation de la configuration d'un container de déploiement du modèle avec les services ACI d'Azure
- déploiement du modèle à travers un service qui prend en paramètre :
    - le fichier PY d'inférence
    - l'environnement PY ou les informations sur les librairies nécessaires au modèle
    - les information de création d'un container de mise en service du modèle
    - le modèle enregistré dans Experiment

Nous récupérons le endpoint du modèle déployé prêt pour utilisation.

8. Une démo fonctionnelle avec l’outil Gradio