Détectez les Bad Buzz grâce au Deep Learning

Création d'un prototype d'un produit IA pour

prédire le sentiment associé à un tweet

Utilisation de données

Open Source

Azure

Environnement de travail

Librairies python spécialisées importées :

- Scikit-learn
- Keras
- TensorFlow

Transformation du jeu de données

"@sandydemandy =O I wish I could just afford the verizon version flat out when it drops.....but I can't...and don't have an upgrade... "

Remplacement des liens internet
Remplacement des mentions (@...]
Remplacement des hashtags
Remplacement des émoticônes
Remplacement des abréviations et contractions
Suppression des caractères uniques
Remplacement des caractères spéciaux

+

Tokenisation des phrases
Tokenisation des mots
Suppression de la ponctuation
Étiquetages morpho-syntaxiques
Lemmatisation
Elimination des stop-words

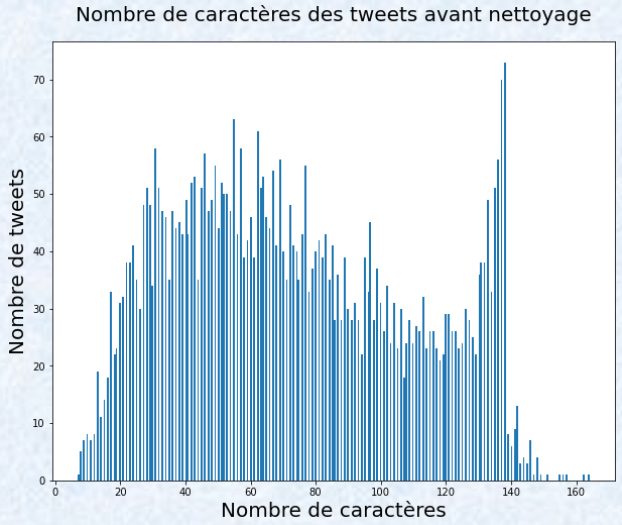
OU

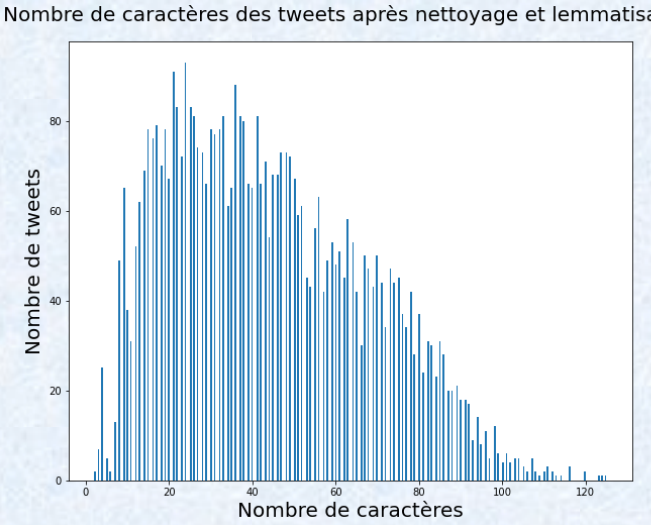
Tokenisation des mots
Suppression de la ponctuation
Racinisation
Elimination des stop-words

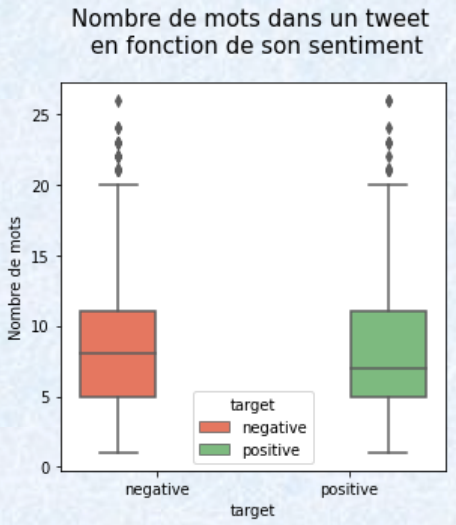
' **user emoticon**_surprise wish could just afford the verizon version flat out when it drops.....but **can not**...and **do not** have an upgrade... '

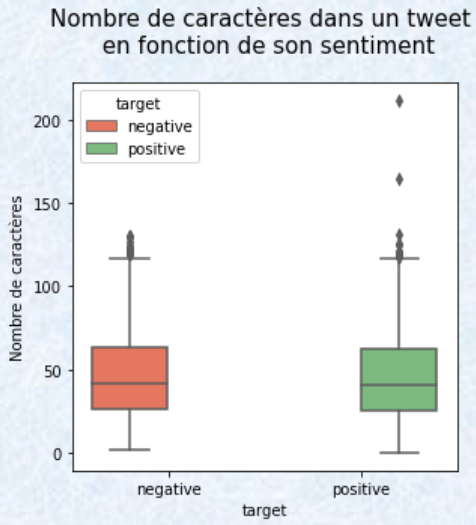
'user emoticon_**surpris** wish could afford verizon version flat out drop but not not **upgrad**'

'user emoticon **surprise** wish could afford verizon version flat out drop but not not **upgrade**'







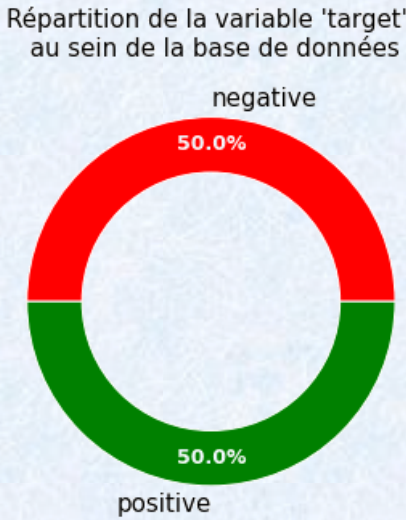


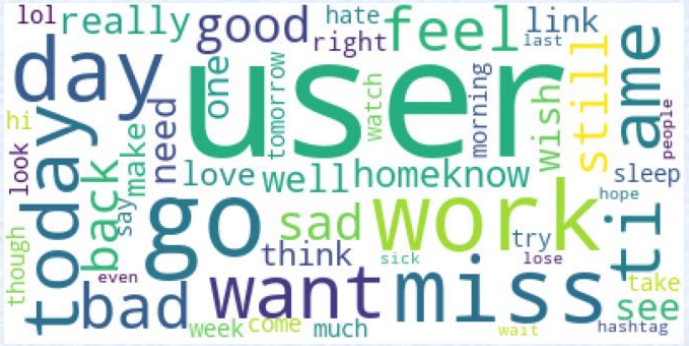
Transformation du jeu de données

positive

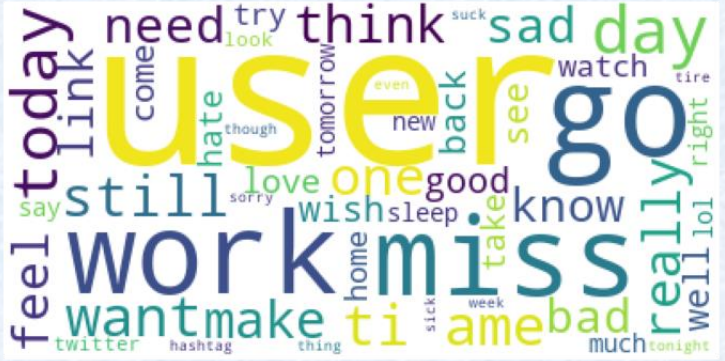
negative

Lemmatisation





Racinisation



Les modèles simples

Bag of Words

Occurrence des n-grammes

	« love, resort, fee, love, burger »	« love, view, love, sorry, credit, card, serious, fee»
Resort	1	0
Love	2	2
Credit	0	1

Seuil de fréquence minimum

Seuil de fréquence maximum

N-grammes

Term frequency-inverse document frequency

$$TFIDF_{t,d,D} = TF_{t,d} \times IDF_{t,D}$$

- $TFIDF_{t,d,D}$: Importance d'un terme t dans un document d
- $TF_{t,d}$: Fréquence d'un terme t dans un document d
- $IDF_{t,D}$: Importance du terme t dans l'ensemble des documents D

	« love, resort, fee, love, burger »	« love, view, love, sorry, credit, card, serious, fee»
Resort	1/5 * log (2 / 1) = 0,06	0
Love	2/5 * log (2 / 2) = 0	0
Credit	0	0,03

Classifieur Naive Bayes (nltk)

	Base de données	temps	accuracy
0	Petit - Brut	3.5	0.56
1	Petit - Nettoyé	3.1	0.55
2	Petit - Lemmatisation	2.5	0.57
3	Petit - Racinisation	2.5	0.57
4	Grand - Brut	16.7	0.57
5	Grand - Nettoyé	15.2	0.57
6	Grand - Lemmatisation	12.1	0.59
7	Grand - Racinisation	12.1	0.59

Classifieur Naive Bayes (Sklearn)

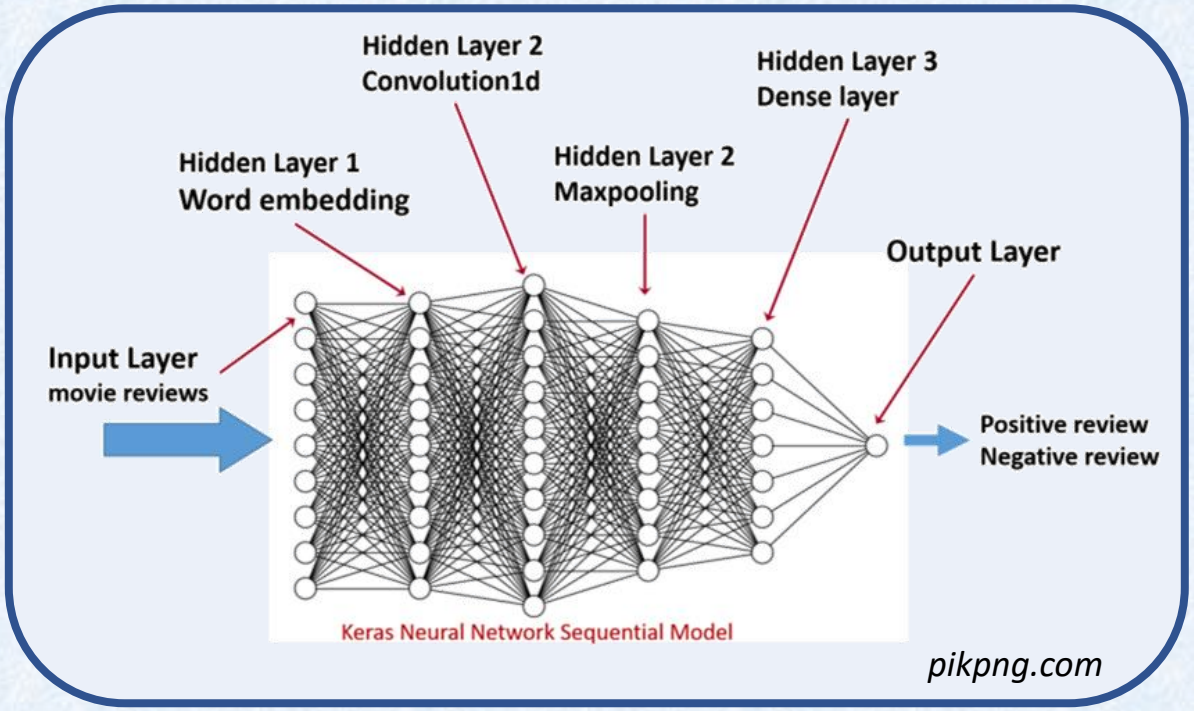
	Base de données	temps	accuracy	precision
0	Petit - Lemmatisation - BoW	0.1	0.71	0.71
1	Petit - Racinisation - BoW	0.1	0.71	0.71
2	Grand - Lemmatisation - BoW	0.2	0.73	0.72
3	Grand - Racinisation - BoW	0.2	0.73	0.72
4	Petit - Lemmatisation - TF-IDF	0.1	0.70	0.69
5	Petit - Racinisation - TF-IDF	0.1	0.70	0.69
6	Grand - Lemmatisation - TF-IDF	0.1	0.72	0.71
7	Grand - Racinisation - TF-IDF	0.1	0.73	0.71

Régression logistique (Sklearn)

	Base de données	temps	accuracy	precision
0	Petit - Lemmatisation - BoW	0.3	0.72	0.70
1	Petit - Racinisation - BoW	0.3	0.72	0.70
2	Grand - Lemmatisation - BoW	1.3	0.73	0.70
3	Grand - Racinisation - BoW	1.6	0.73	0.71
4	Petit - Lemmatisation - TF-IDF	0.2	0.71	0.70
5	Petit - Racinisation - TF-IDF	0.2	0.71	0.71
6	Grand - Lemmatisation - TF-IDF	1.0	0.73	0.72
7	Grand - Racinisation - TF-IDF	1.0	0.74	0.72

Les modèles d'apprentissage profond

Le principe du Deep Learning



Choix des couches + paramétrisation des couches

Embedding

- Taille du vocabulaire
- Dimension de l'entrée
- Dimension de la sortie

Dropout

- Taux de neurones ignorés

Flatten

Dense

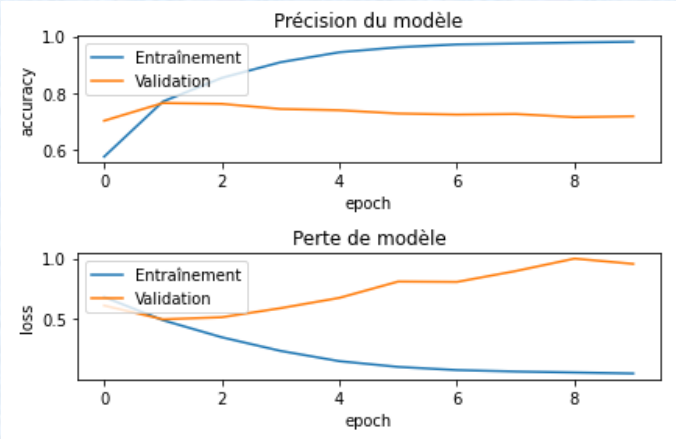
- Nombre de neurones

Optimisation :

- Choix du learning rate
- Choix du type d'optimisation (learning rate)
- Choix de la fonction de perte (binary_crossentropy)
- Choix de la métrique à optimiser (accuracy)

Optimisation :

- Batch_size
- Epoch

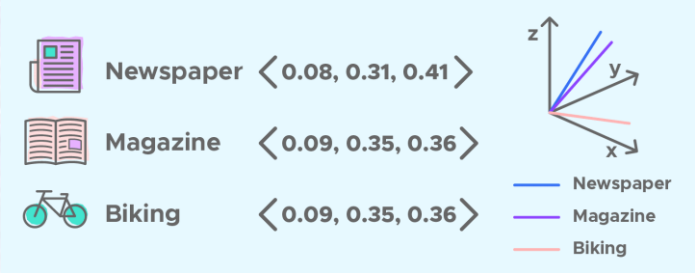


Les modèles d'apprentissage profond

Les types de plongements de mots

Embedding
=
Représentation vectorielle du vocabulaire d'un corpus

Mots qui ont une signification proche
→ vecteurs représentatifs + corrélés

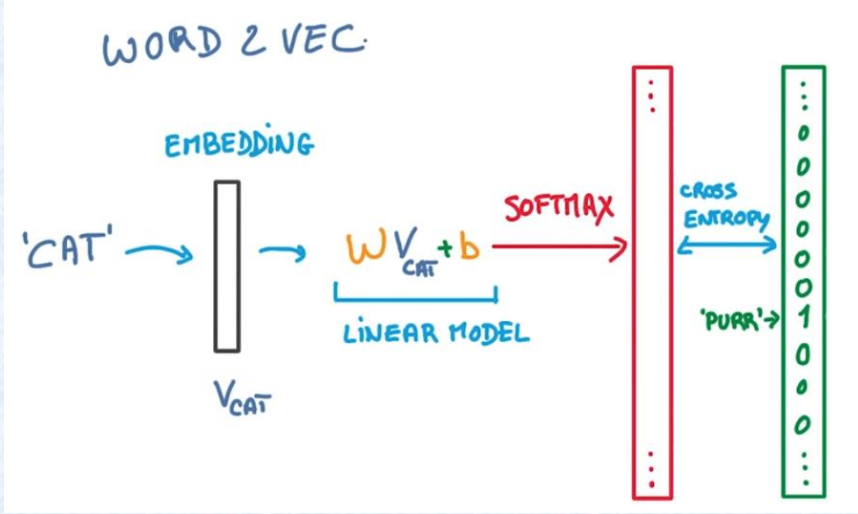


Etablir le contexte d'un corpus et étudier la sémantique (les similarités de langage) des différents mots des textes

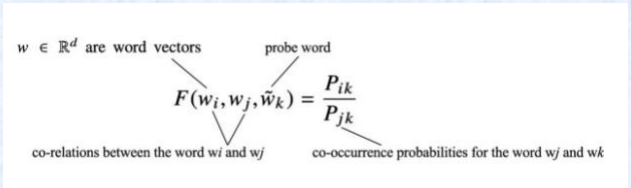
Grand texte : beaucoup de vecteurs !

Word2vec

CBOW → prédit la probabilité d'un mot dans le contexte
Skip-gram → prédit le contexte d'un mot



Glove

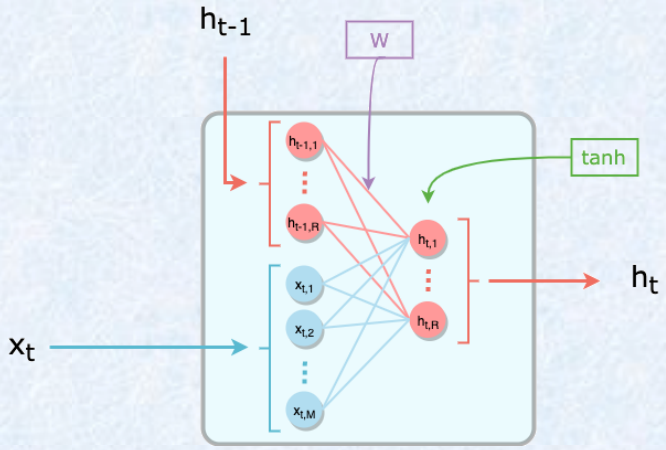


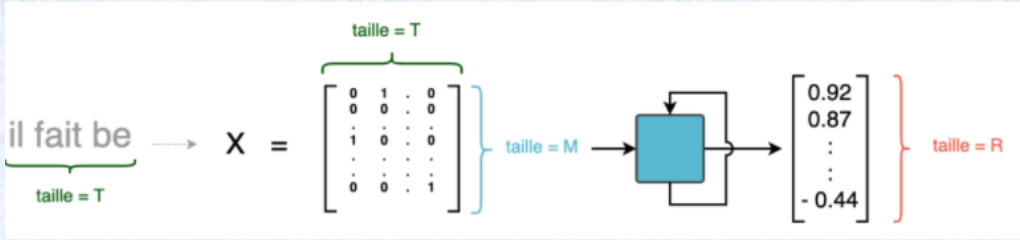
Mesure de la similarité de facteurs entre des mots pour prédire leur co-occurrence

Source : jonathan-hui.medium.com

Les modèles d'apprentissage profond

Les réseaux de neurones récurrents

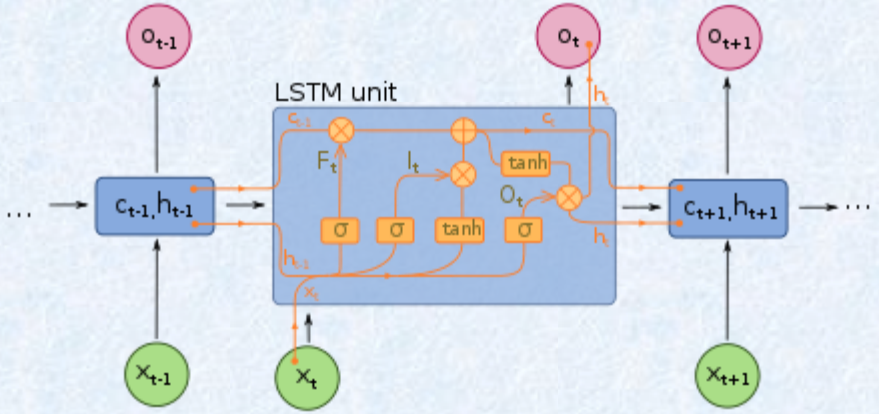




simpleRNN

- Taille de sortie
- Taille séquence (T)
- Nombre de variables (M)

Le modèle LSTM - Long Short-Term Memory



1. Détecter les informations pertinentes venant du passé (via forget gate)

2. Choisir celles qui seront pertinentes à long terme (via l'input gate, cell state → mémoire longue)

3. Piocher dans cell state les informations importantes à court terme (hidden state suivant à travers l'output gate)

LSTM

- Activation
- Taille séquence (T)
- Nombre de variables (M)

Les modèles d'apprentissage profond

Modélisations locales

Keras simple **embedding simple**

	Base de données	temps	accuracy	precision
0	Grand - Lemmatisation	25.6	0.75	0.78
1	Grand - Racinisation	26.6	0.76	0.76
2	Grand - Tweets nettoyés	24.6	0.75	0.80
3	Grand - Tweets bruts	27.0	0.75	0.70

Keras **RNN** embedding simple

	Base de données	temps	accuracy	precision
0	Grand - Lemmatisation	54.2	0.74	0.74
1	Grand - Racinisation	52.5	0.75	0.76
2	Grand - Tweets nettoyés	53.6	0.75	0.76
3	Grand - Tweets bruts	58.9	0.75	0.74

Keras **LSTM** embedding simple

	Base de données	temps	accuracy	precision
0	Grand - Lemmatisation	343.7	0.75	0.74
1	Grand - Racinisation	24.7	0.76	0.76
2	Grand - Tweets nettoyés	25.5	0.76	0.78
3	Grand - Tweets bruts	27.6	0.76	0.76

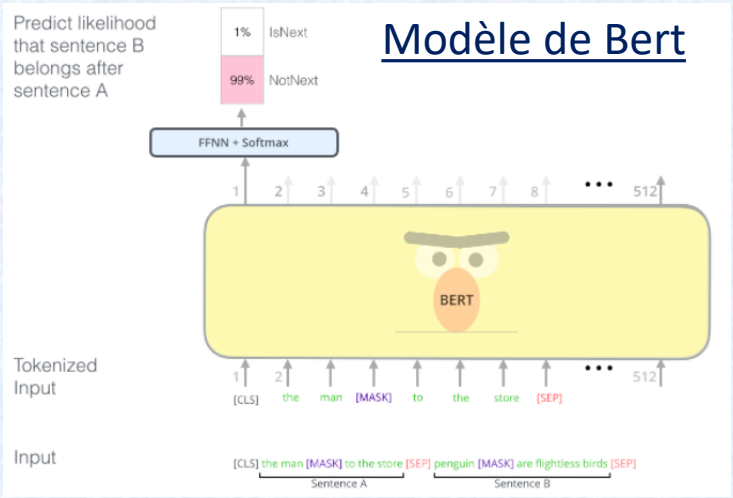
Keras LSTM embedding **Word2vec**

	Base de données	temps	accuracy	precision
0	Grand - Lemmatisation	461.1	0.75	0.78
1	Grand - Racinisation	516.0	0.76	0.76
2	Grand - Tweets nettoyés	524.0	0.76	0.77
3	Grand - Tweets bruts	544.5	0.75	0.73

Keras LSTM embedding **Glove**

	Base de données	temps	accuracy	precision
0	Grand - Lemmatisation	567.7	0.76	0.77
1	Grand - Racinisation	662.9	0.75	0.74
2	Grand - Tweets nettoyés	773.6	0.76	0.79
3	Grand - Tweets bruts	843.8	0.75	0.73

Modèle de Bert



	precision	recall	f1-score	support
0	0.81	0.80	0.81	524
1	0.78	0.80	0.79	476
accuracy			0.80	1000
macro avg	0.80	0.80	0.80	1000
weighted avg	0.80	0.80	0.80	1000

Les modèles d'apprentissage profond

Déploiement du modèle sur Azure

Keras **LSTM** embedding simple

- Importation du meilleure modèle
- Importation du meilleur tokenizer

Etapes du déploiement du modèle sur Azure :

1 – Relier l'environnement local au compte Azure
2 – Récupérer les variables d'environnements utiles pour l'identification
3 – Accéder au workspace
4 – Charger les modèles et les enregistrer sur l'espace de travail
5 – Créer le script avec les fonctions souhaitées pour notre API
6 – Définir l'environnement Python à envoyer à l'inférence
7 – Envoyer l'environnement et le script avec les fonctions

Name	Version	Created on ↓	Tags	Created by
best_tokenizer	3	Jan 18, 2022 ...	area : ···	Sandy Morais
best_model	3	Jan 18, 2022 ...	area : ···	Sandy Morais

Name	State	Type	Attached/Created	Location
p7-cluster-1	Succeeded	Kubernetes service	Created	francecentral

	Name ↑	Category	Available quota ⓘ
✓	Standard_A2_v2 2 cores, 4GB RAM, 20GB storage	General purpose	350 cores

```
1 service.test_cloud_1_tweet(['i love apples'])
```

```
Le temps d'acquisition des prédictions est de 1.2 sec.
La prédiction pour ce tweet est [0]
```

```
Le temps d'acquisition des prédictions est de 55.1 sec.
Le score accuracy est : 0.75
Le score precision est : 0.74
              precision    recall  f1-score   support

           0       0.75      0.75      0.75       508
           1       0.74      0.74      0.74       492

    accuracy                           0.75      1000
   macro avg       0.75      0.75      0.75      1000
weighted avg       0.75      0.75      0.75      1000
```

L'approche « API sur étagère »

Service Cognitif Azure - Microsoft

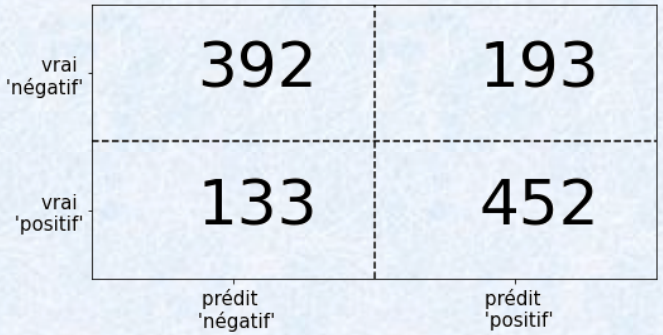
Identification + requête → réponse

Score positif
Score neutre
Score négatif

Azure

prediction_azure_tweet		prediction_tweet
[0.03, 0.96, 0.01]	→	1
[0.01, 0.0, 0.99]		0
[0.01, 0.04, 0.95]		0
[0.19, 0.77, 0.04]		1
[0.08, 0.91, 0.01]		1

Résultats



	precision	recall	f1-score	support
0	0.75	0.66	0.70	580
1	0.70	0.78	0.74	580
accuracy			0.72	1160
macro avg	0.72	0.72	0.72	1160
weighted avg	0.72	0.72	0.72	1160

```
Le score accuracy est : 0.72
Le score precision est : 0.7
```

Temps : entre 3 et 4 minutes

Niveaux/fonctionnalités	F0 Gratuit 5K Transactions sur 30 jours	S Standard 1000 Appels par minute
Prix	**0,00 € EUR par mois**	**à partir de 1,00 USD/1000 enregistrements texte**
Analyse des Sentiments	✓	✓
Extraction de phrases clés	✓	✓
Détection de langue	✓	✓
Extraction d'entité	✓	✓
Réponse aux questions	✓	✓ 1,26 € EUR