



SOUTENANCE

ANALYSE DE SENTIMENT DE TWEETS

Cédric Dietzi

05/03/21

Sommaire

- Contexte
- Jeu de données
- Performances de référence
- Architectures
- Résultats
- Démonstration

Sentiment Analysis



My experience
so far has been
fantastic!

POSITIVE



The product is
ok I guess

NEUTRAL



Your support team is
useless

NEGATIVE

Contexte

La société **Air Paradis** souhaite anticiper le bad buzz sur les réseaux sociaux.

Prototyper un produit IA permettant de prédire le sentiment associé à un Tweet

Sentiment Analysis



My experience
so far has been
fantastic!

POSITIVE

800 000



The product is
ok I guess

NEUTRAL



Your support team is
useless

NEGATIVE

800 000

Jeu de données

1 600 000 tweets

1. **target:** négatif ou positif
2. **ids:** une clé d'identification du tweet
3. **date:** date de création
4. **flag:** un flag
5. **user:** utilisateur
6. **text:** le texte du tweet

NB: pas d'émojis dans les textes

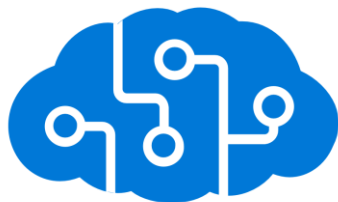
Go, A., Bhayani, R. and Huang, L., 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford, 1(2009)*



Vision



Language



Microsoft Azure
Cognitive Services

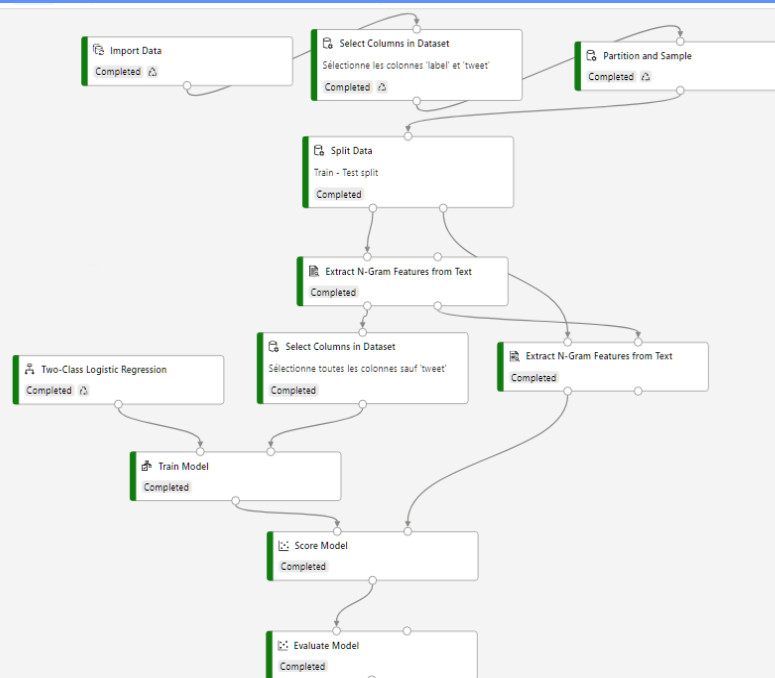


Speech



Knowledge

Azure Designer



Voir en annexe pour la version agrandie

Performances de référence - Baselines

Performances de référence (Accuracy)

Azure Cognitive Services	73,1%
Azure ML Designer : 1- Gram + Régression logistique	74,9%

Good, I should say Great Experiences with this great Company !!

Preprocessing

Lemmatisation ou Stemming

good-i-should-say-great-experience-with-this-great-company

Codage

Word2Vect ou Glove

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

1 5 6 7 2 1 5 4 8 5 4 7 9 6 1 0 1 5 6 9 0 5 2 4 3 2 1 5 3 2 1 5 1 5 6 9 4 3 9 9

Génération de variables

Convolutional network

Identifie la présence de **motifs**
au niveau local

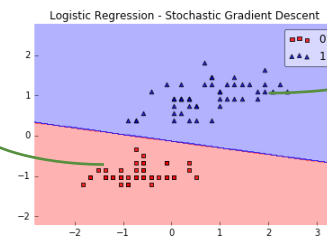
ou

Recurrent network

Identifie la présence de **motifs**
au niveau global

Décision

Négatif



Positif

Architectures

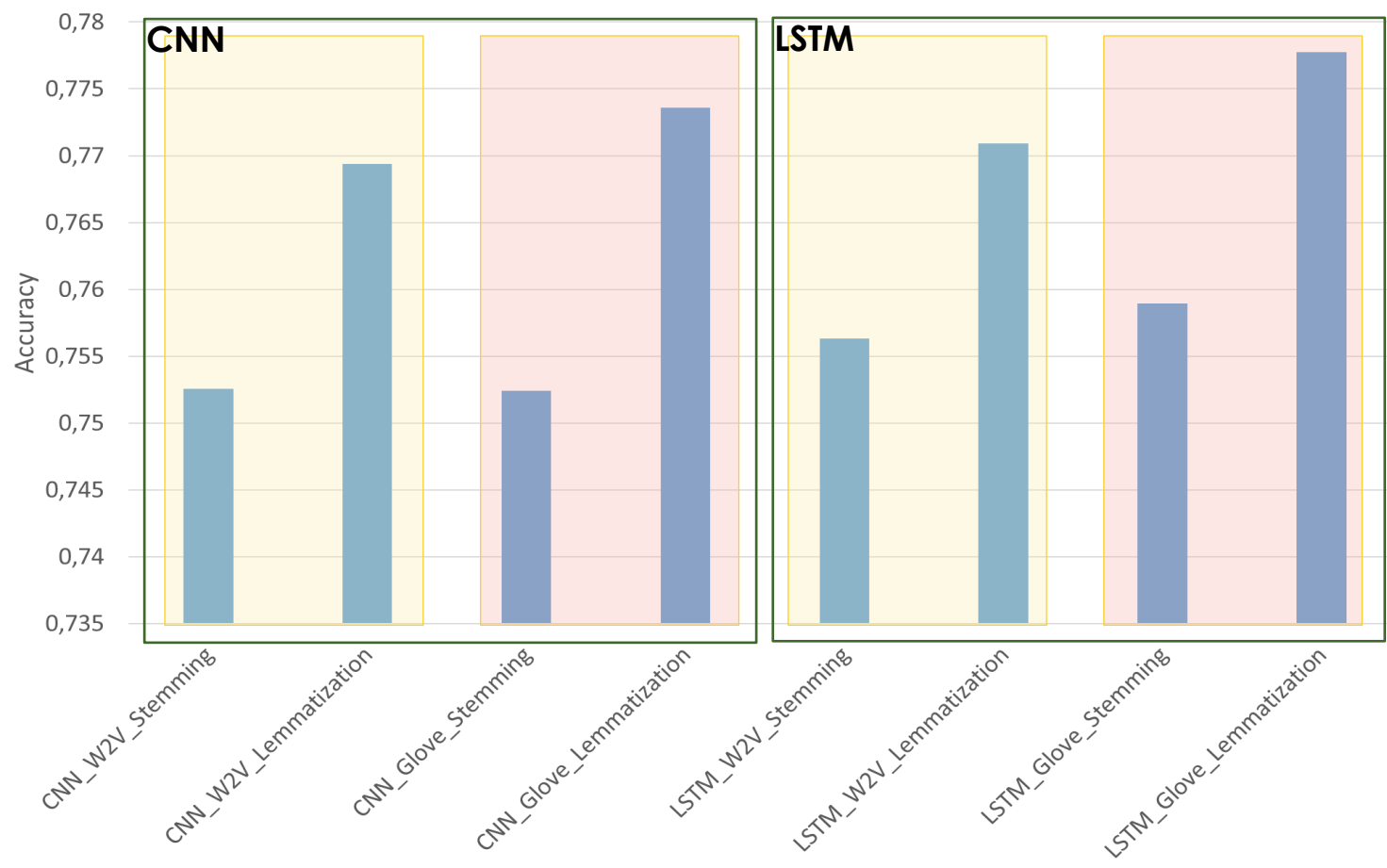
Preprocessing

Codage des mots (embedding)

Génération de variables
(feature engineering – RNN ou CNN)

Décision (dense layer)

Accuracy des différentes architectures



Résultats

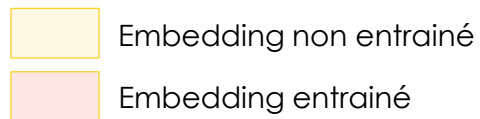
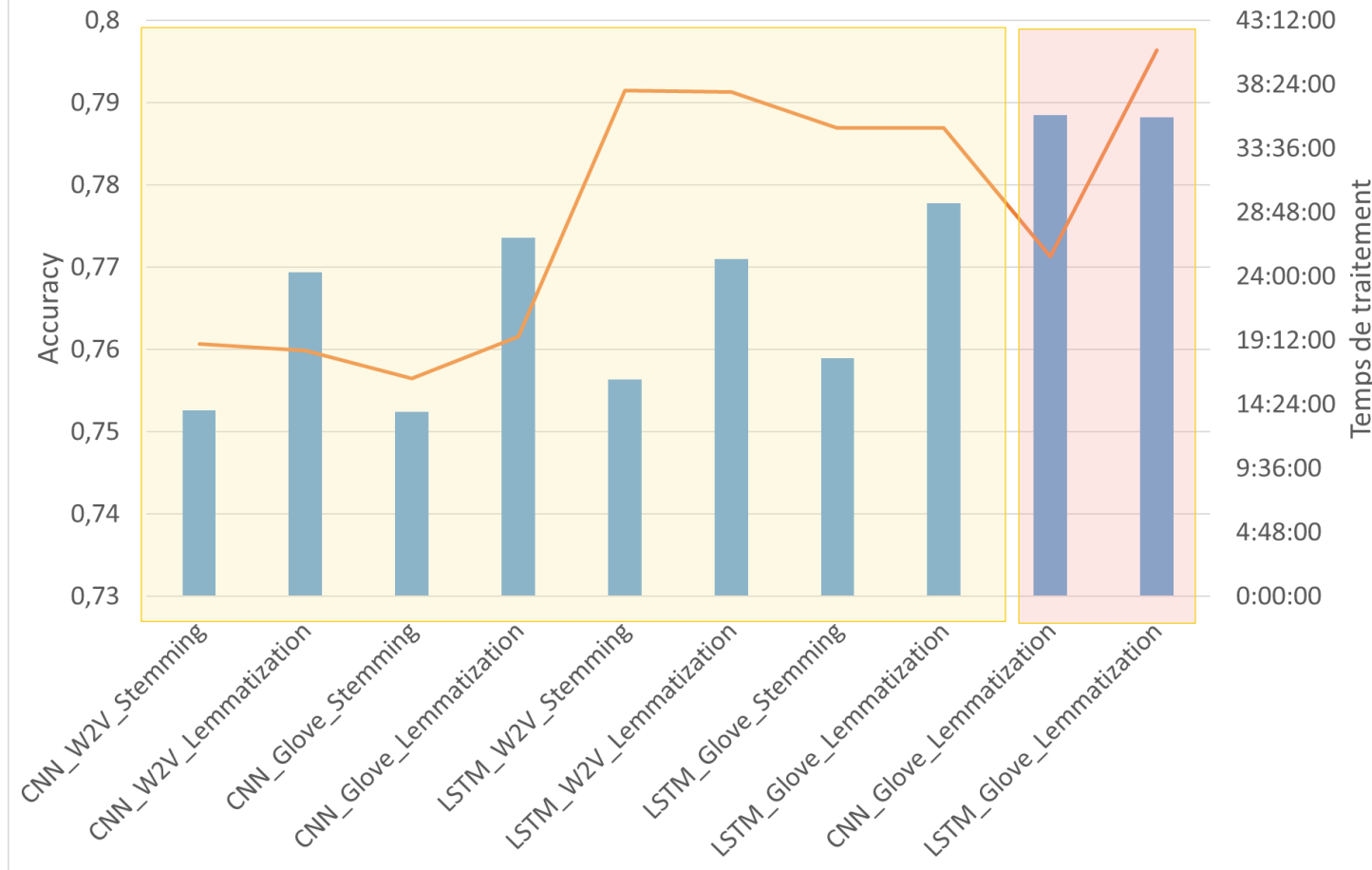
Entraînement sur 80 000 exemples. Embedding non entraîné.

La lemmatisation et l'embedding GloVe donnent de meilleurs accuracies que le stemming et Word2Vec.

sur CNN : 77,36%

sur LSTM : 77,77%

Accuracy des différentes architectures



Résultats

L'entraînement de l'embedding améliore de +1% les résultats.

Les deux architectures CNN ou LSTM obtiennent des performances comparables mais LSTM est 60% plus lent.

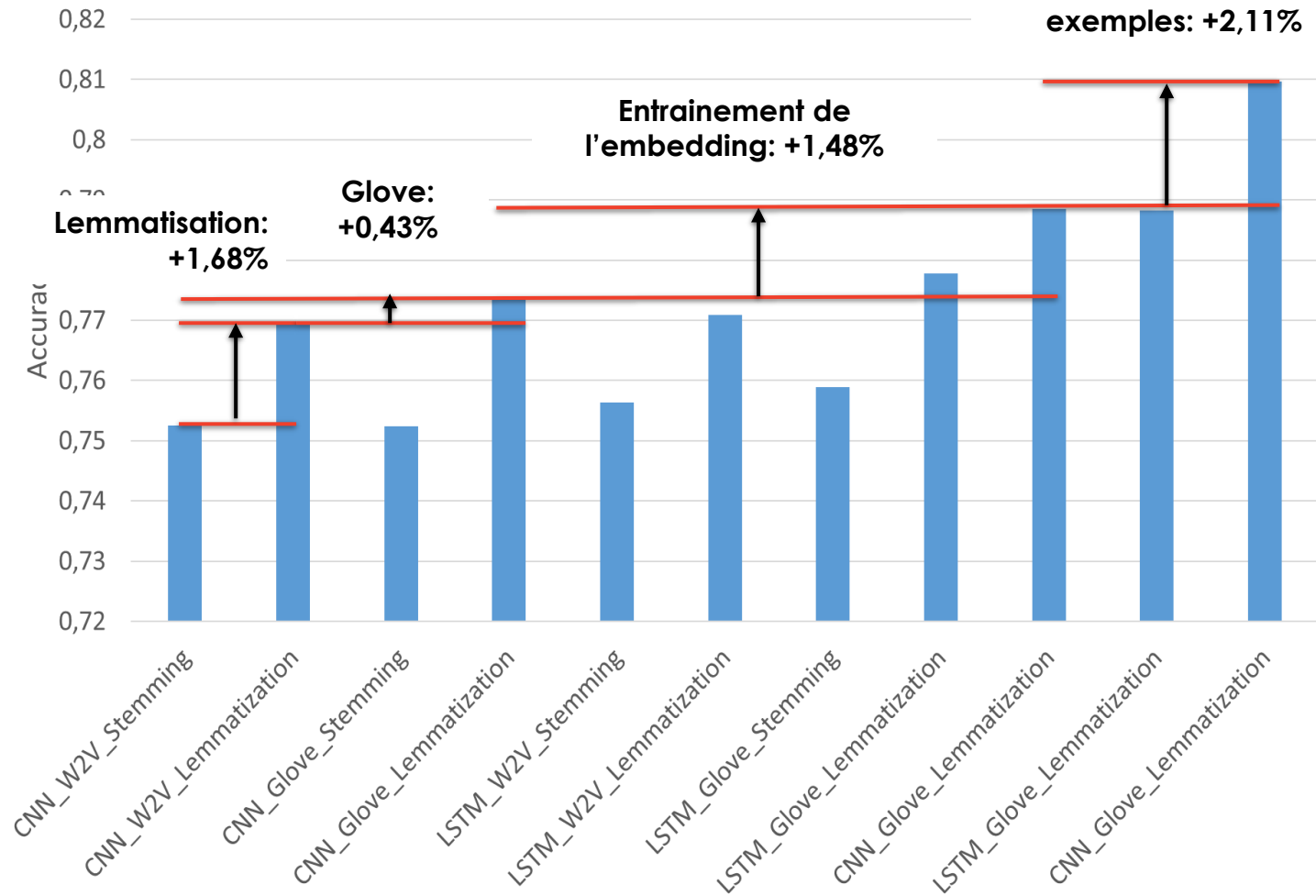
CNN = 78.84%

LSTM = 78.82%

Architecture retenue:
Lemmatisation + Glove +
CNN

Accuracy des différentes architectures


Entraînement sur > 1,5M
exemples: +2,11%



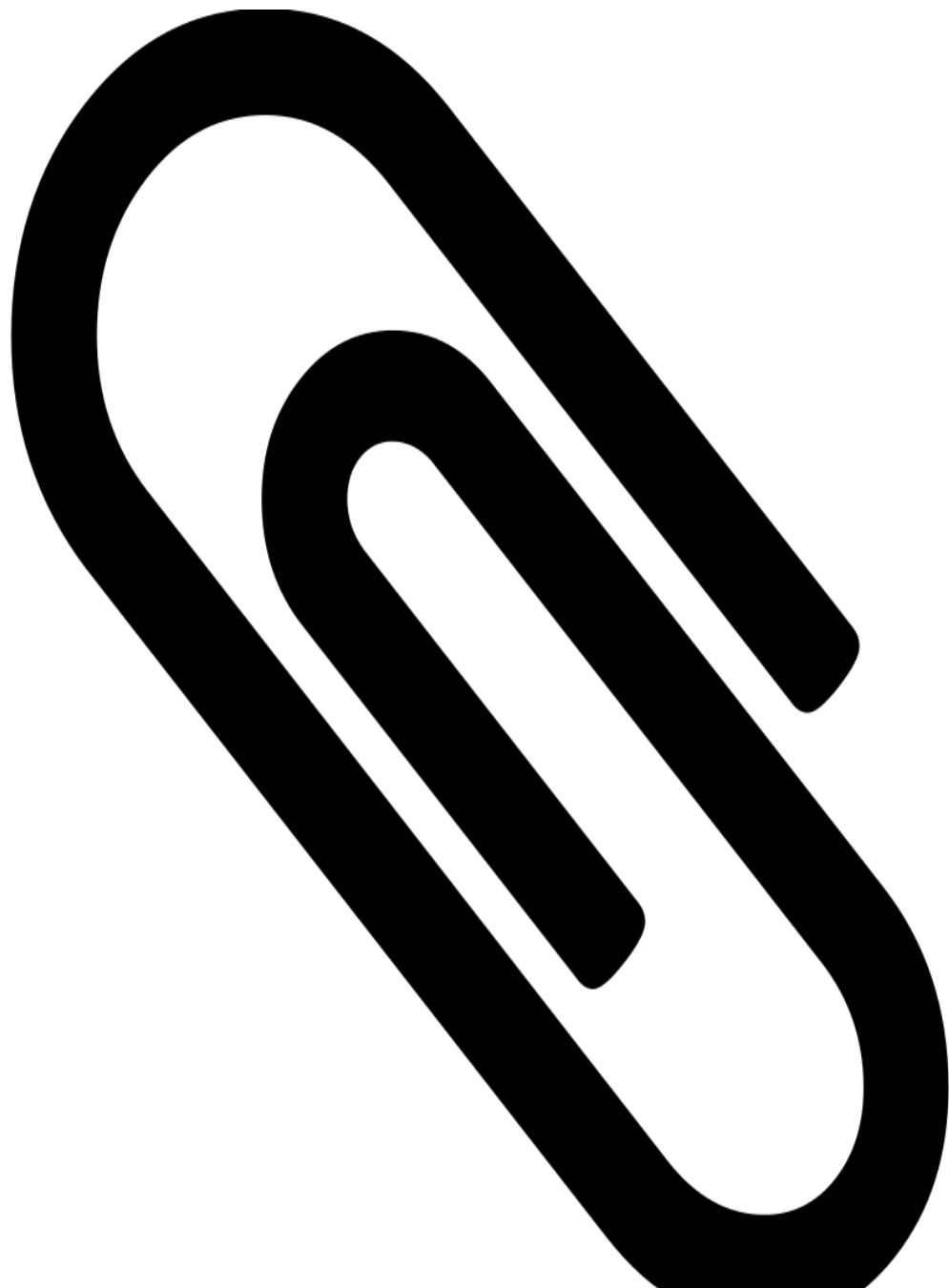
Résultat final

Entraînement sur 1,52 M
d'exemples.

Accuracy: 81 %

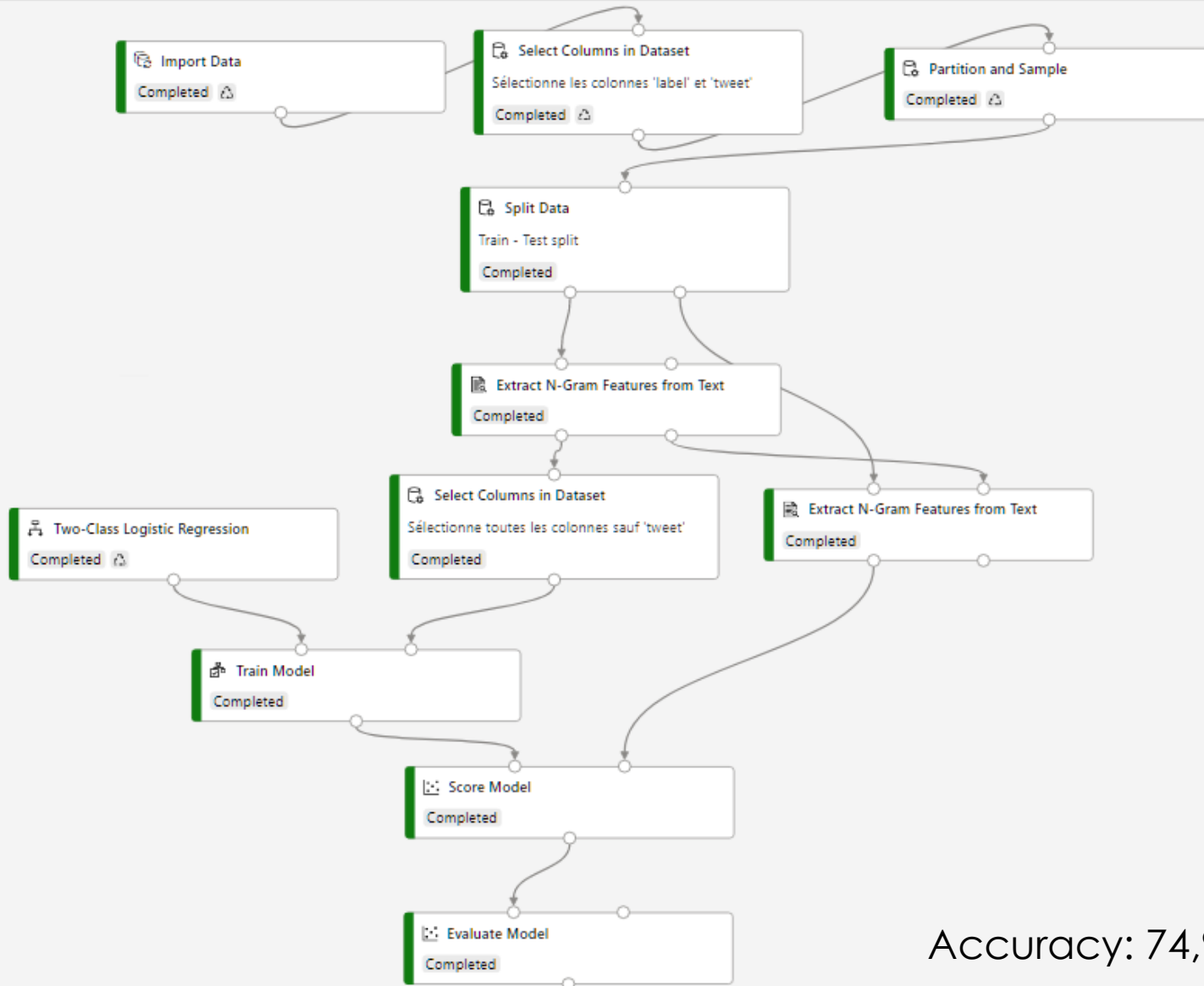


Démo !



Annexes

Azure Designer



Accuracy: 74,9%

Schéma du modèle de référence dans le Designer Azure ML

Performances de référence

Azure Cognitive Services	73,1%
Azure Cognitive Services	74,9%

Fichier de résultats



Feuille de
calcul Microsoft E