



Tweets **Sentiment Analysis**

Ahmed Rachid Hazourli
Anes Mekki

Table des matieres

Data Exploration

Exploration des
données

Model Training

Choix du modèle

Data Preprocessing

Différentes étapes du
prétraitement des
données

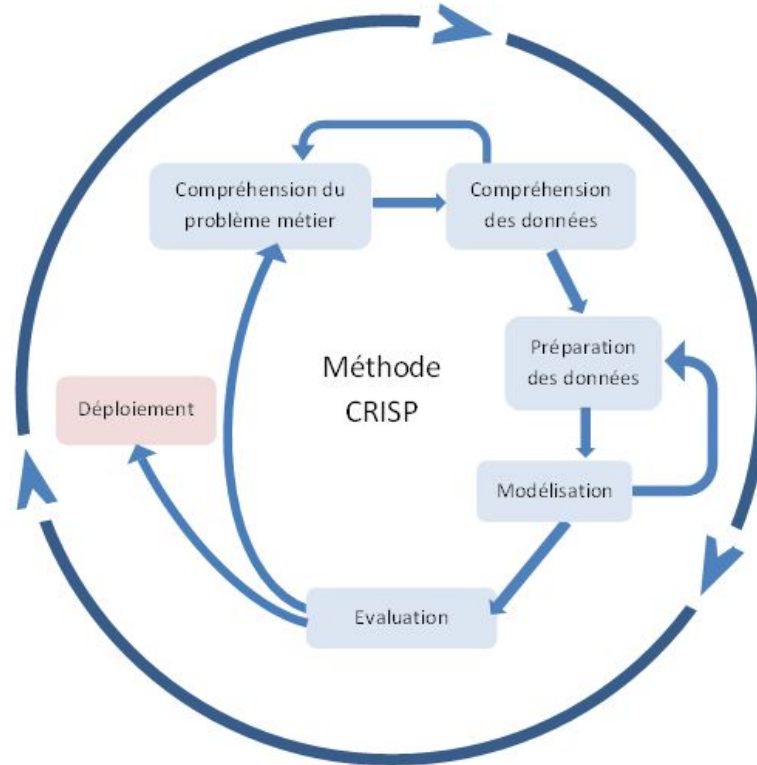
Résultats

Evaluation des
modèles

Étapes de la réalisation du projet



Démarche agile





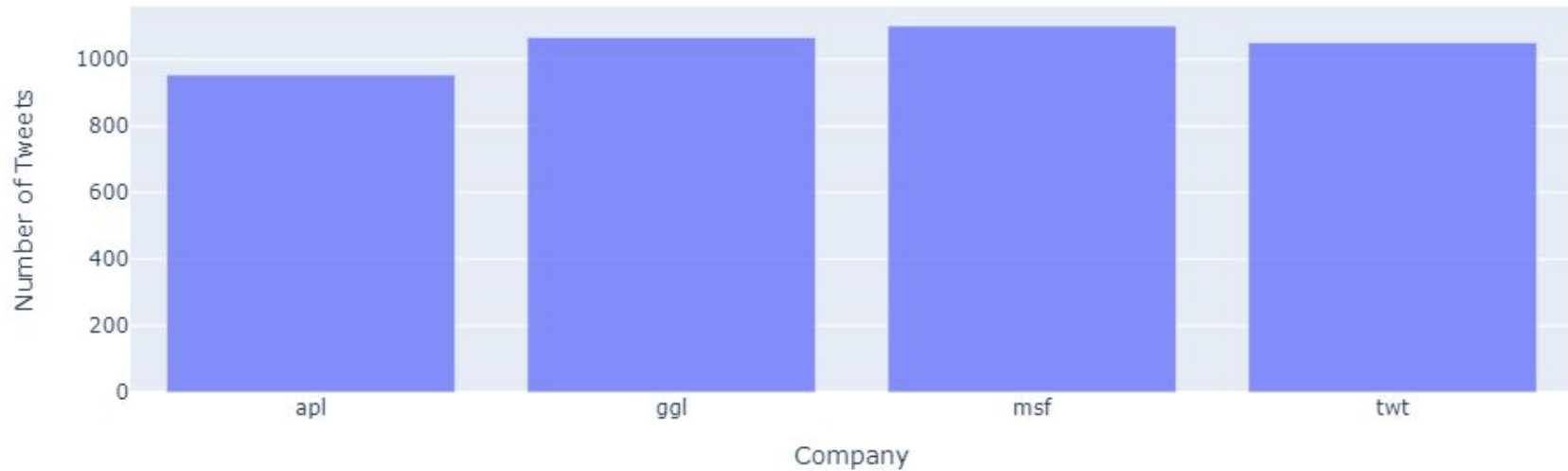
01

Data Exploration

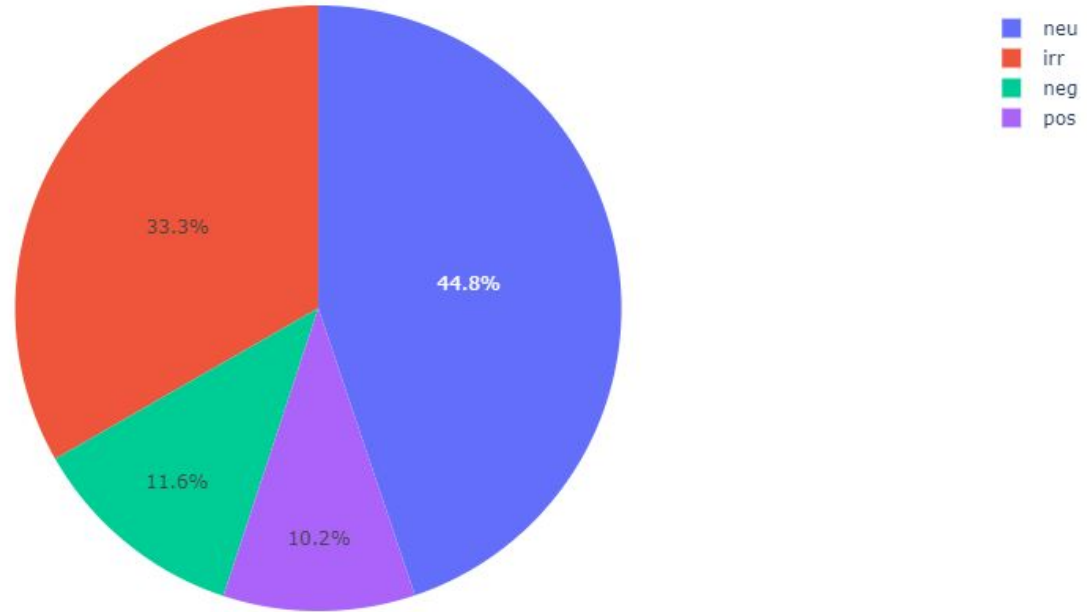
Data Exploration

- Notre dataset contient **4173 tweets**.
- Il existe 4 différents types des textes:
 - Positive, Negative, Neutral et Irrelevant
- Notre corpus contient des tweets sur 4 entreprises:
 - Microsoft, Google, Apple et Twitter

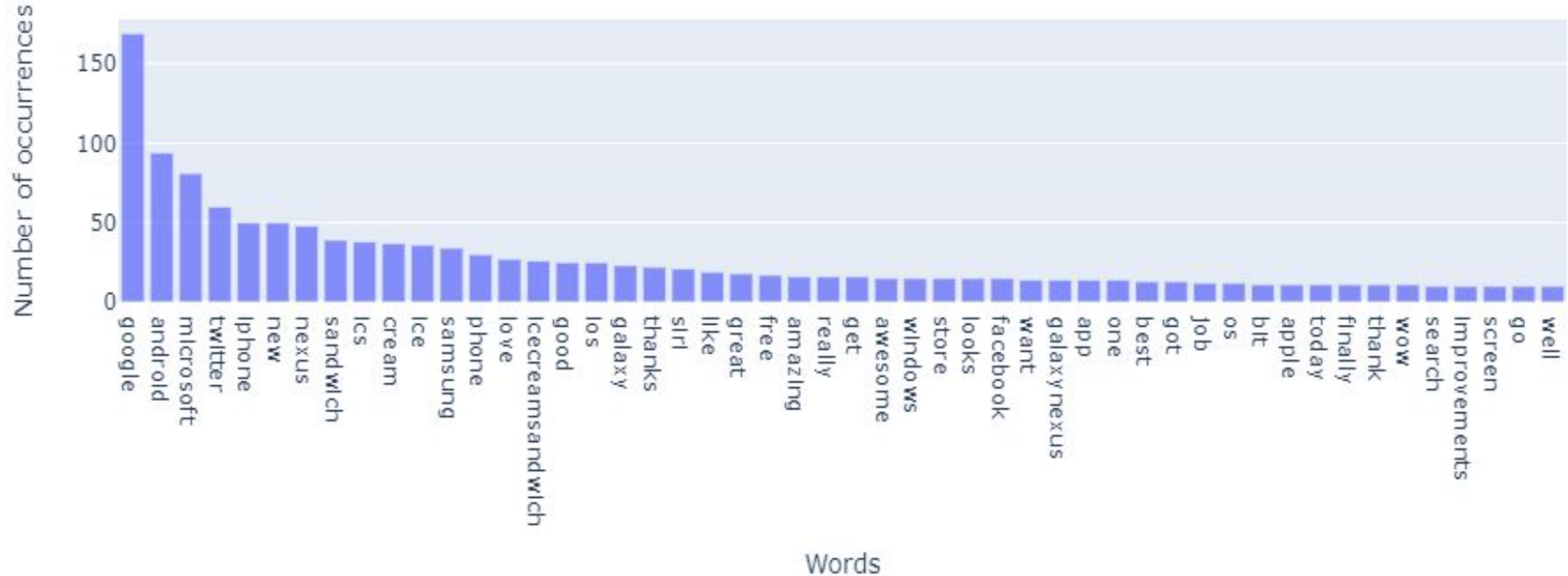
Nombre de Tweets par Entreprise



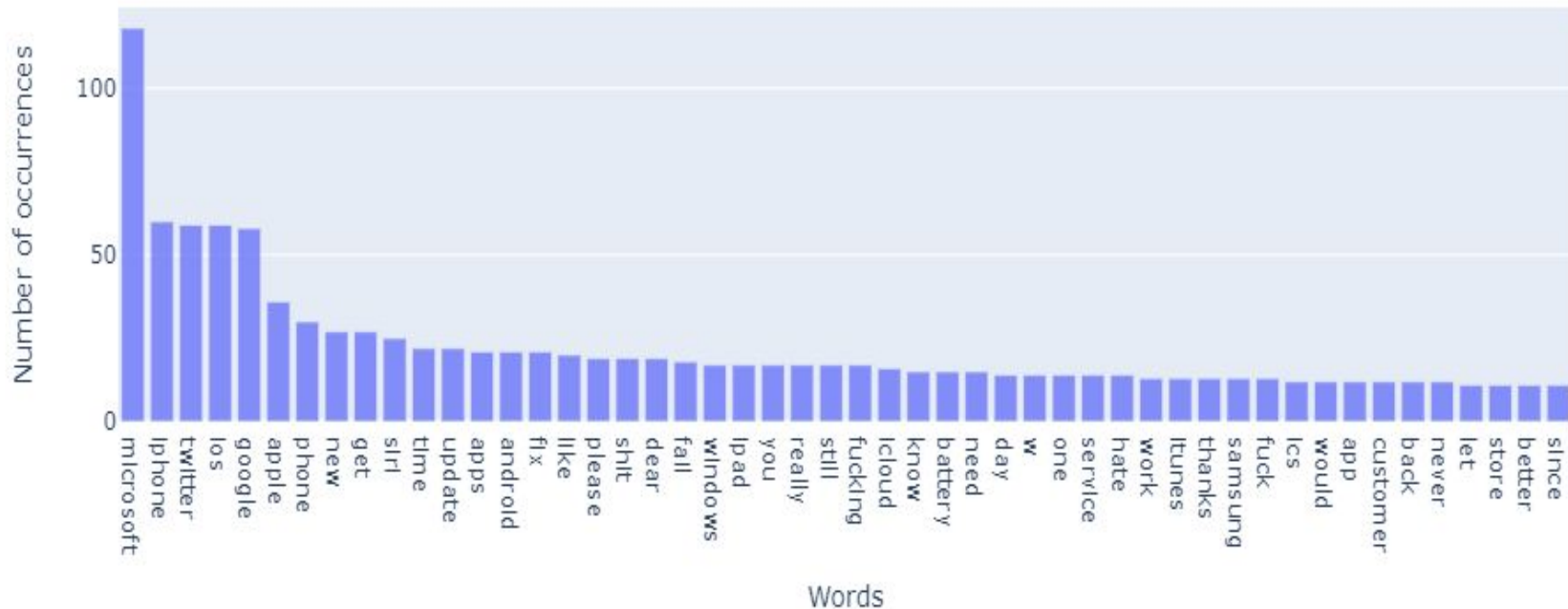
Pourcentage de Tweets par classe de Sentiment



Top words dans les Tweets Positifs



Top words dans les Tweets Negatifs

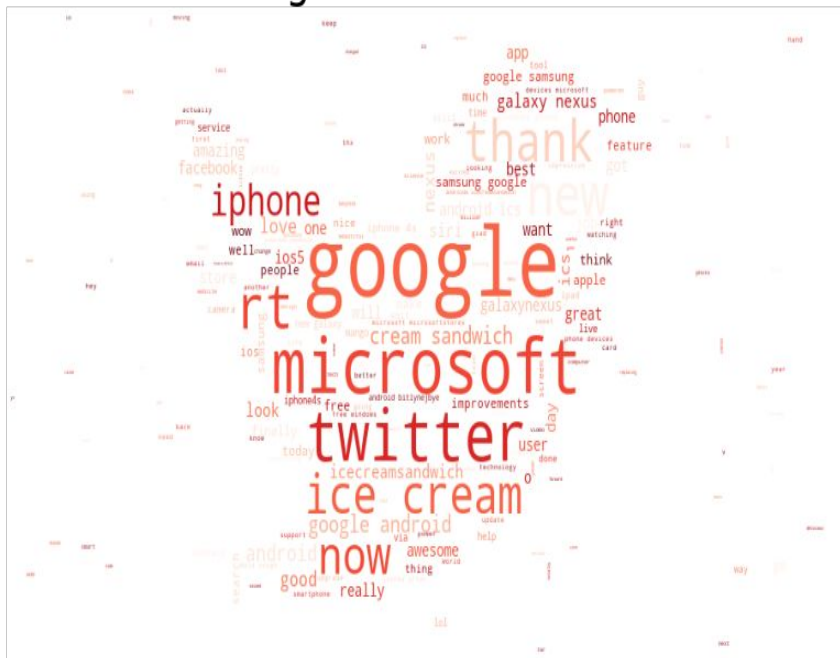


- 
- A decorative graphic on the left side of the slide consisting of two overlapping squares. The top square is a medium blue, and the bottom square is a darker blue, extending further down and to the left.
- Plusieurs mots montrent explicitement le sentiment exprimé dans le tweet comme :
 - Sentiment positif: love, good, like...
 - Sentiment négatif: shit, fucking, fuck, hate...

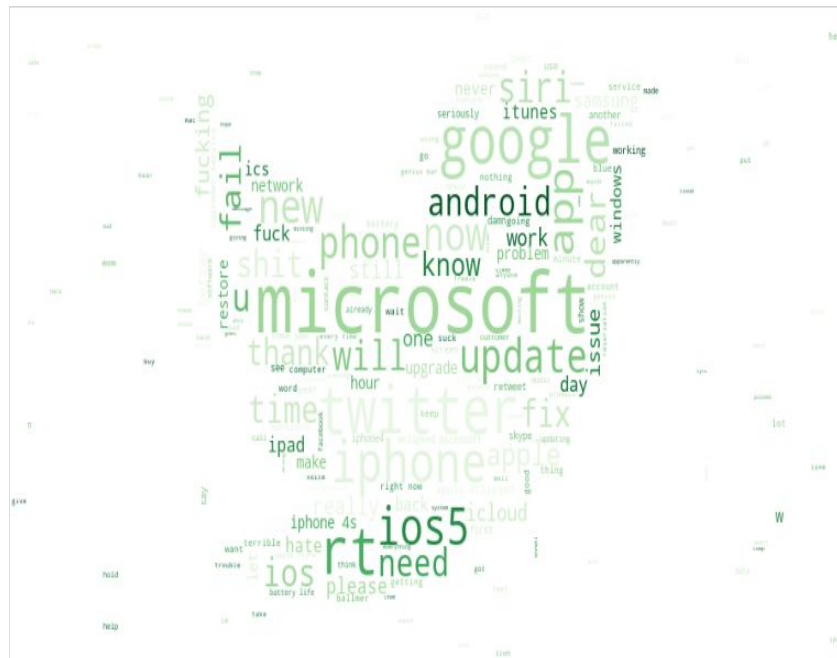
WordCloud

Voici un nuage de mots dans lequel on peut apercevoir quels sont les mots les plus utilisés dans les tweets de ses utilisateurs (plus la taille de la police d'un mot est grande, plus le nombre d'occurrences de celui-ci est important)

Negative Sentiment

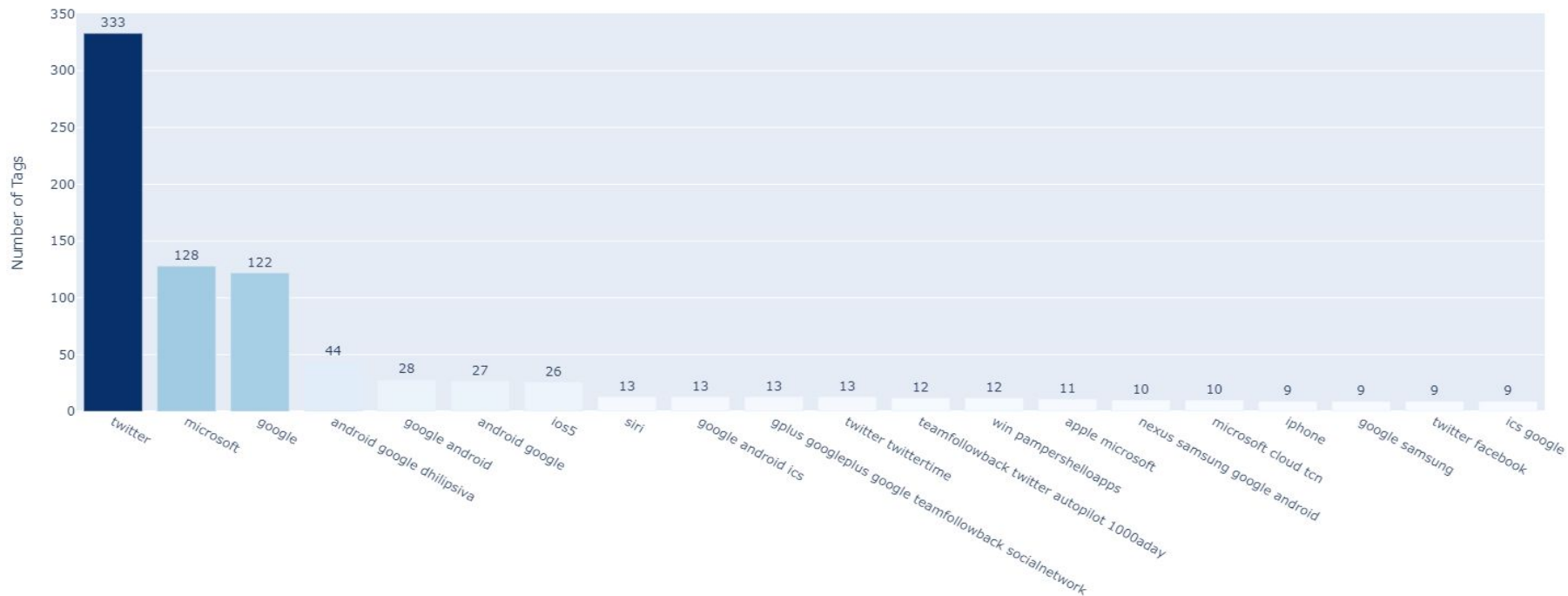


Positive Sentiment



Les hashtags les plus utilisés

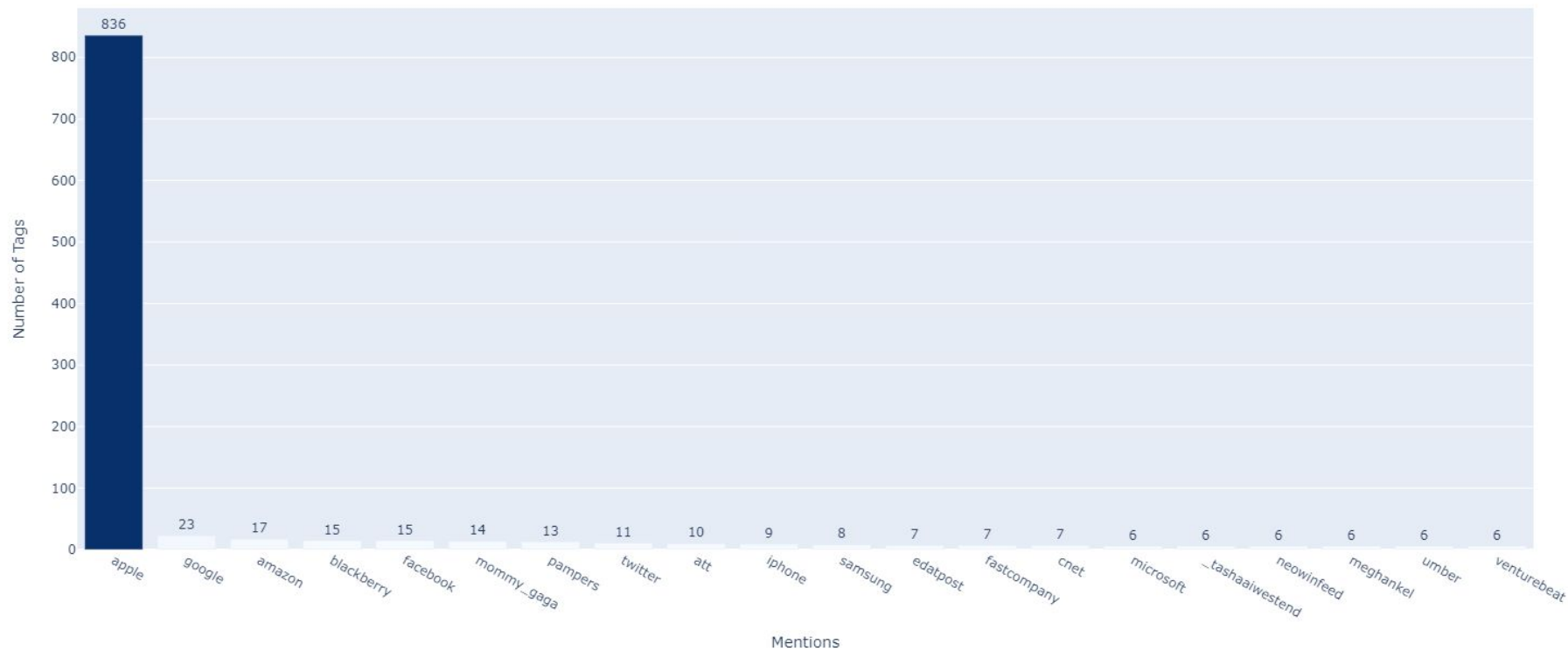
Top Trended Hastags



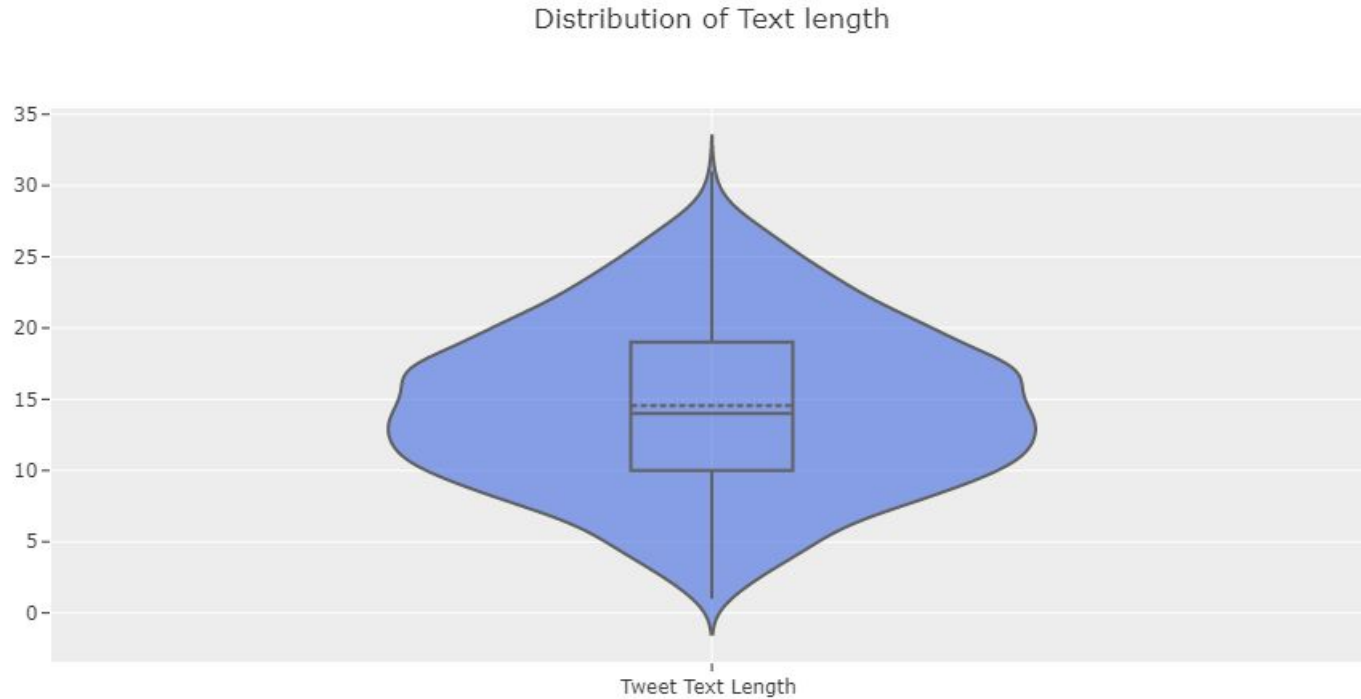
Hashtags

Les mentions les plus utilisées

Top Trended Mentions

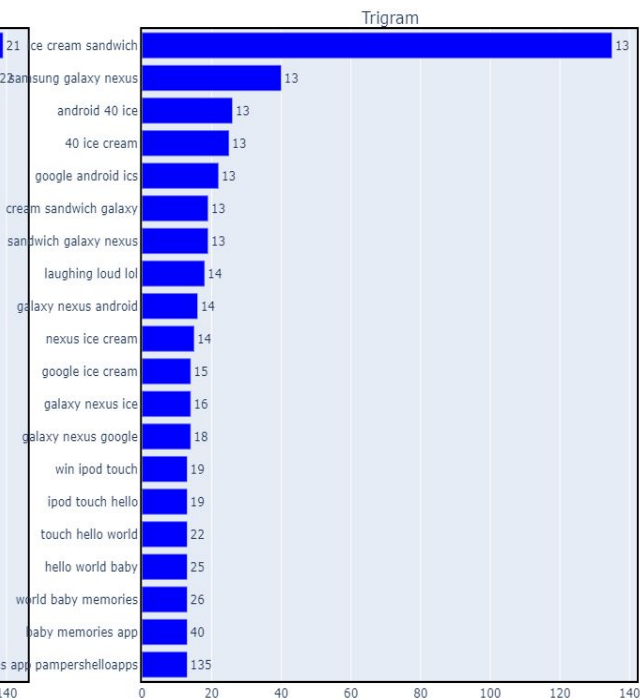
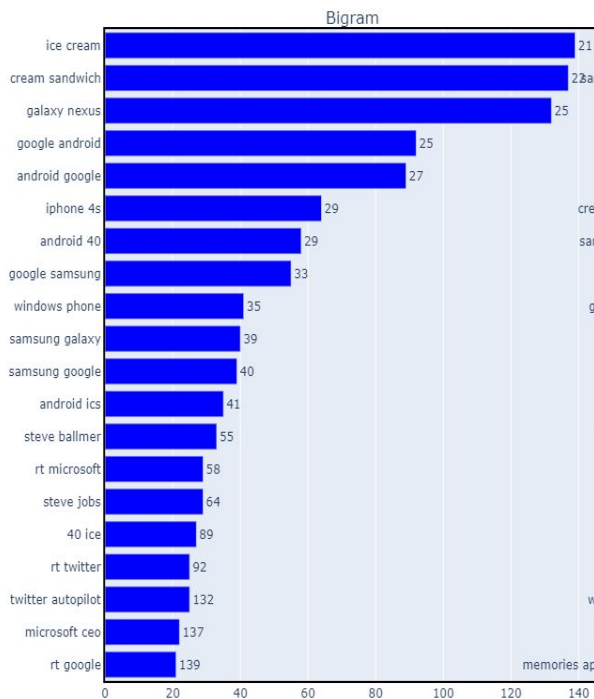
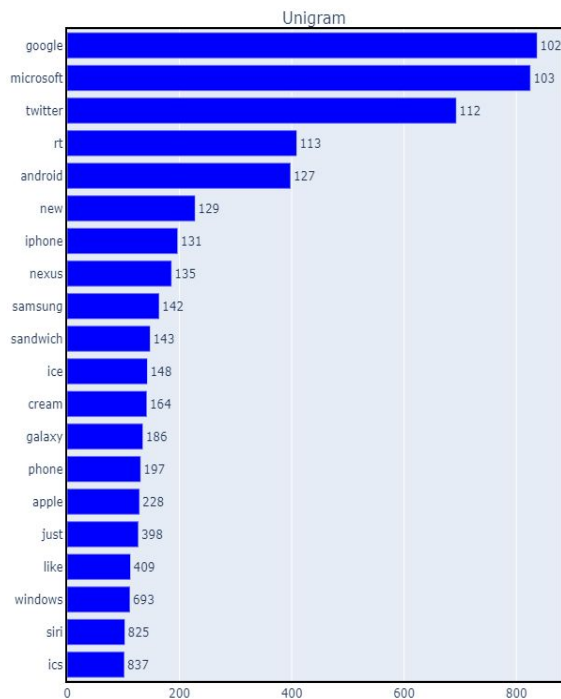


Distribution de la longueur des Tweets



Top N - grams

Top N Grams





02

Data Preprocessing

Préparation des données

Mise en
miniscule

Suppression des
stopwords

Suppression des
hashtags

Suppression des
mentions

Suppression des
URL

Lemmatization

Tokenization

Normalization

Formatage des
emojis

Suppression des
ponctuations

Les étapes du prétraitement:

La première étape à chaque fois que l'on fait du NLP est de construire une pipeline de nettoyage de nos données.

Le plus gros travail du data scientist ne réside pas dans la création de modèle. Le nettoyage du dataset représente une part énorme du processus, nous avons appliqué ces étapes:

- La mise en minuscule des tweets a été une étape importante dans notre démarche.
- **Suppression des stopwords:** les mots fréquents et courants sans valeur inhérente comme: the, and, while ...
- **Suppression des hashtags:** comme #microsoft, #google, #apple
- **Suppression des mentions:** comme @apple ...
- **Suppression des URLs:** comme http://t.co/opjnrXlb...
- **Suppression des ponctuations:** comme ?, !...
- **Formatage des emojis:** convertir :) en **smile**, :(en **sad**...
- **Normalization:** we'll => we will, ain't => are not
- **Tokenization:** séparation des mots et des caractères spéciaux.
- **Lemmatization:** retour à la racine des mots, comme: being -> be.



03

Models

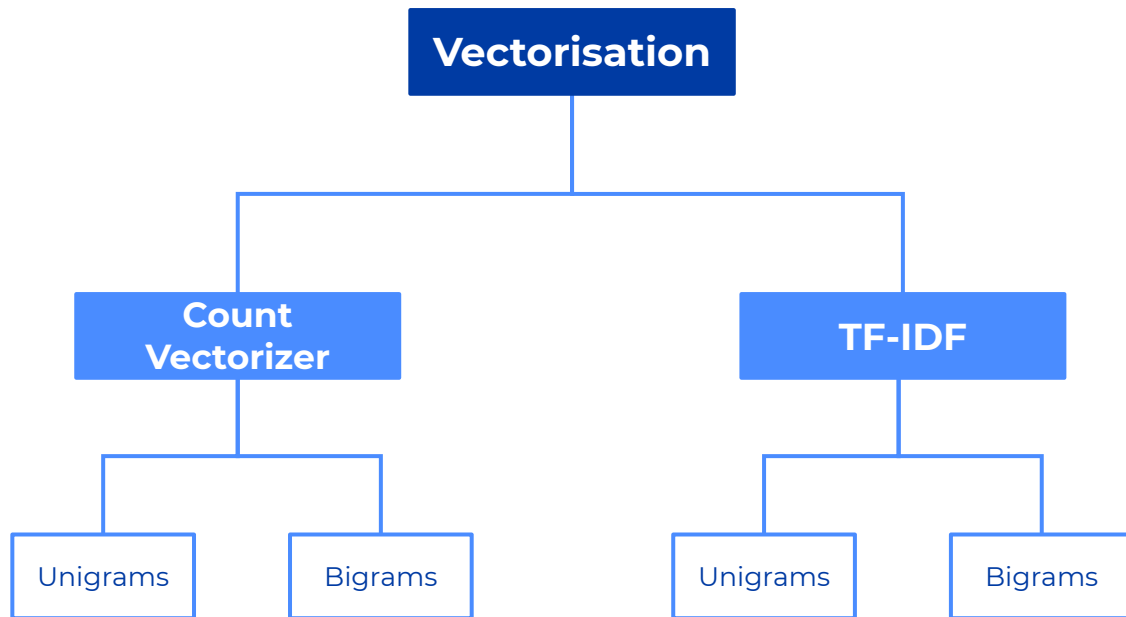
A decorative graphic on the left side of the slide consisting of two blue squares. One is a medium blue square positioned higher and to the right, and the other is a darker blue square positioned lower and to the left, partially overlapping the first one.

Word Embedding

Le word embedding est capable en réduisant la dimension de capturer le contexte, la similarité sémantique et syntaxique (genre, synonymes, ...) d'un mot.

Bag Of Words - Word Embeddings

Désormais, nous allons commencer notre travail de classification à l'aide des Word Embeddings.



Bag Of Words - Count Vectorizer

La manière la plus simple de représenter un document, on représente chaque tweet par un ensemble des mots qu'il contient.

Le document est alors représenté par un vecteur de fréquence d'apparition des différents mots utilisés.

Bag Of Words - TF-IDF

TF-IDF ou Term Frequency - Inverse Document Frequency est une méthode d'analyse utilisée dans une stratégie de référencement pour déterminer les mots clés et vectoriser nos documents.

Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection. Elle a pour but de transformer ces vecteurs de mots en une matrice numérique qui pondère les mots par importance. Par exemple un mot qu'on retrouve dans tous les documents n'a aucune pertinence.

From Text to Bag Of Words

Une représentation bag-of-words classique soit par des Unigrams, Bigrams ou Trigrams sera donc celle dans laquelle on représente chaque document par un vecteur de la taille du vocabulaire $|V|$ et on utilisera la matrice composée de l'ensemble de ces N documents qui forment le corpus comme entrée de nos algorithmes.

Après avoir construit nos vecteurs qui représentent les documents, nous allons entraîner les modèles Machine Learning / Deep Learning pour faire la prédiction / classification des Tweets.

Word2Vec

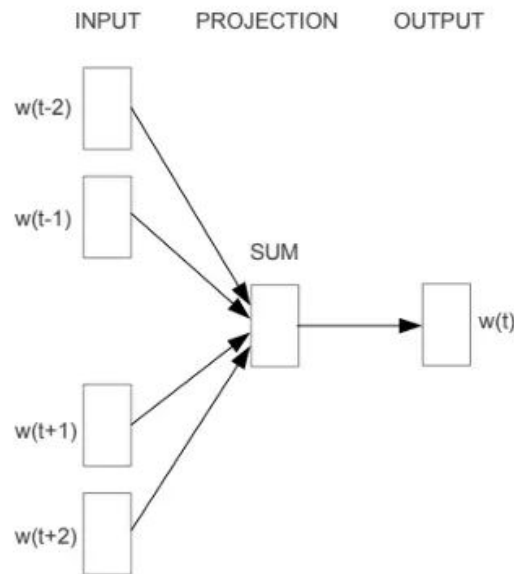
Le plongement de mots a pour objectif de transformer la représentation des mots en vecteurs de dimension définie N. Sous cette forme numérique, on peut ainsi appliquer les différentes opérations arithmétiques sur les vecteurs, l'exemple le plus connu étant :

$$\text{vec}(\text{Roi}) - \text{vec}(\text{Homme}) + \text{vec}(\text{Femme}) = \text{vec}(\text{Reine})$$

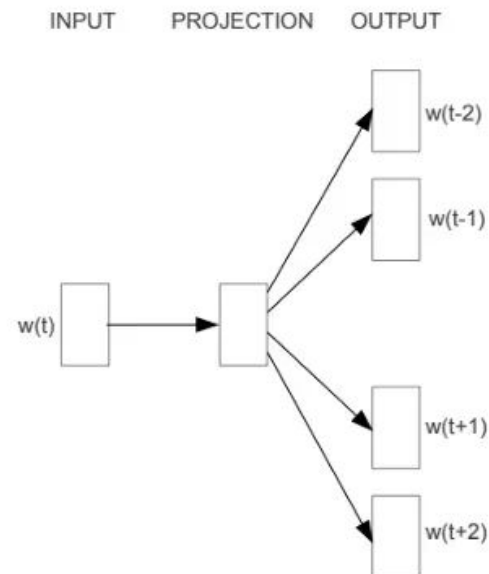
La transformation mot vers vecteur peut se faire à l'aide des réseaux de neurones. L'un des hyperparamètres est la dimensions N des vecteurs qui correspond à la taille de la couche cachée. Il y a deux types de modélisation possibles, en fonction de si l'on souhaite trouver un contexte à partir d'un mot ou trouver un mot à partir d'un contexte. Il existe deux méthodes de modélisation des vecteurs:

- **Skip-Gram** : prédire un contexte à partir d'un mot.
- **Continuous Bag of Words**: prédire le mot à partir d'un contexte

Word2Vec: CBoW vs SkipGram



CBoW



Skip-gram

Pourquoi le Word2Vec ?

Le Word2Vec embedding capture efficacement les propriétés sémantiques et arithmétiques d'un mot. Il permet également de réduire la dimension du problème et par conséquent la tâche d'apprentissage.

Nous pouvons nous imaginer utiliser l'algorithme Word2Vec pour pré-entraîner la matrice d'embedding du modèle de classification.

Par conséquent, notre modèle de classification aura une bien meilleure représentation des mots lors de la phase d'apprentissage des sentiments.



BERT est un modèle de représentation de textes écrit en langage naturel. La représentation faite par BERT à la particularité d'être contextuelle.

C'est-à-dire qu'un mot n'est pas représenté de façon statique comme dans un embedding classique mais en fonction du sens du mot dans le contexte du texte.

Par exemple, le mot "baguette" aura des représentations différentes dans "la baguette du magicien" et "la baguette du boulanger". En plus, le contexte de BERT est bi-directionnel, c'est-à-dire que la représentation d'un mot fait intervenir à la fois les mots qui le précèdent et les mots qui le suivent dans une phrase.

Pourquoi le BERT ?

Le principe d'utilisation de BERT est tout simple : Il est déjà pré-entraîné sur une grande quantité de données, on le modifie pour une tâche précise puis on le re-entraîne avec nos propres données.

La modification dont il est question ici consiste en général à rajouter un réseau de neurones à la sortie de BERT. Cela s'appelle : le **fine-tuning**.

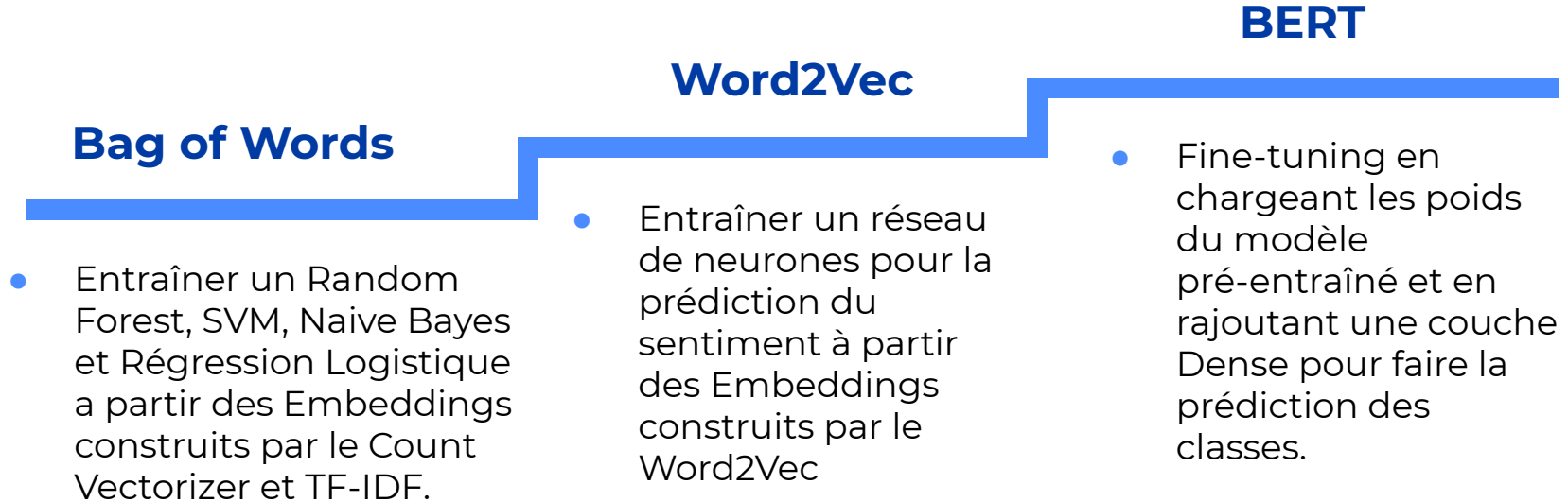
Nous avons utilisé le fine-tuning à l'aide du BERT pré-entraînée pour notre tâche d'Analyse du Sentiment des Tweets.



04

Model Training

Roadmap de l'entraînement des modèles



A decorative graphic on the left side of the slide consisting of two overlapping squares. The top square is a lighter blue, and the bottom square is a darker blue.

Data Splitting & Model Training

L'idée est comme d'habitude de diviser notre jeu de données en un échantillon d'entraînement (80%) dans lequel nous allons apprendre les paramètres du modèle et un échantillon test (20%) dans lequel nous allons les tester.

Nous entraînons nos Word Embeddings construits à partir des Bag of Words (Count Vectorizer et TF-IDF) par des modèles Machine Learning comme: Random Forest, SVM, Naive Bayes et la Régression Logistique.

A decorative graphic consisting of two blue squares. One is a medium blue square in the upper left, and the other is a darker blue square below it, extending further to the left.

Model Training

Par rapport aux embeddings construits par le Word2Vec nous entraînons un modèle Réseaux de Neurones avec une fonction de perte 'cross_entropy'.

Notre modèle sera composé des couches suivantes :

- La couche Embedding va transformer chaque mot du contexte en vecteur d'embedding. La matrice W de l'embedding sera apprise au fur et à mesure que le modèle s'entraîne.
- Ensuite, la couche GlobalAveragePooling1D permet de sommer les différents embedding pour avoir une dimension en sortie.
- Enfin, La couche Dense permet de prédire le mot cible.



05

Evaluation des modèles

Évaluer un modèle de classification

La façon la plus simple d'évaluer son modèle est de le tester sur des données dont on connaît déjà les labels et de comparer les résultats du modèle aux vraies valeurs des labels. Il existe plusieurs métriques comme:

- **Matrice de confusion:** plus la matrice de confusion tend à être diagonale, plus la classification est précise.
- **Accuracy:** est une métrique qui permet de mesurer le pourcentage de réussite de notre classifieur sur le jeu de données d'évaluation.
- **Précision:** est le pourcentage d'éléments correctement associés à un certain label par rapport à tous les éléments associés à ce label.
- **Rappel:** est le pourcentage de classifications correctes suivant un label donné. Soit la capacité du modèle à classifier correctement les éléments d'une catégorie donnée.
- **F1-score:** est une métrique qui prend en compte la précision et le rappel sur une catégorie donnée. Elle peut être vue comme une combinaison de ces deux métriques.

Evaluation des modèles

Après l'entraînement des différents modèles, nous pouvons mesurer leur performance en fonction des différentes métriques citées ci-dessus. Voici un tableau explicatif:

	Accuracy	Précision	Rappel	F1-score
Count-Vectorizer + Random Forest	0.68	0.71	0.68	0.63
Count-Vectorizer + SVM	0.67	0.67	0.68	0.66
Count-Vectorizer + Naive Bayes	0.61	0.63	0.62	0.63
Count-Vectorizer + Regression Logistique	0.69	0.70	0.70	0.68

Evaluation des modèles

	Accuracy	Précision	Rappel	F1-score
TF-IDF + Random Forest	0.61	0.62	0.62	0.62
TF-IDF + SVM	0.70	0.72	0.70	0.68
TF-IDF + Naive Bayes	0.66	0.69	0.66	0.60
TF-IDF + Regression Logistique	0.62	0.69	0.63	0.55

Evaluation des modèles

	Accuracy	Précision	Rappel	F1-score
Word2Vec +Neural Network	0.69	0.65	0.68	0.66
BERT	0.92	0.96	0.96	0.96

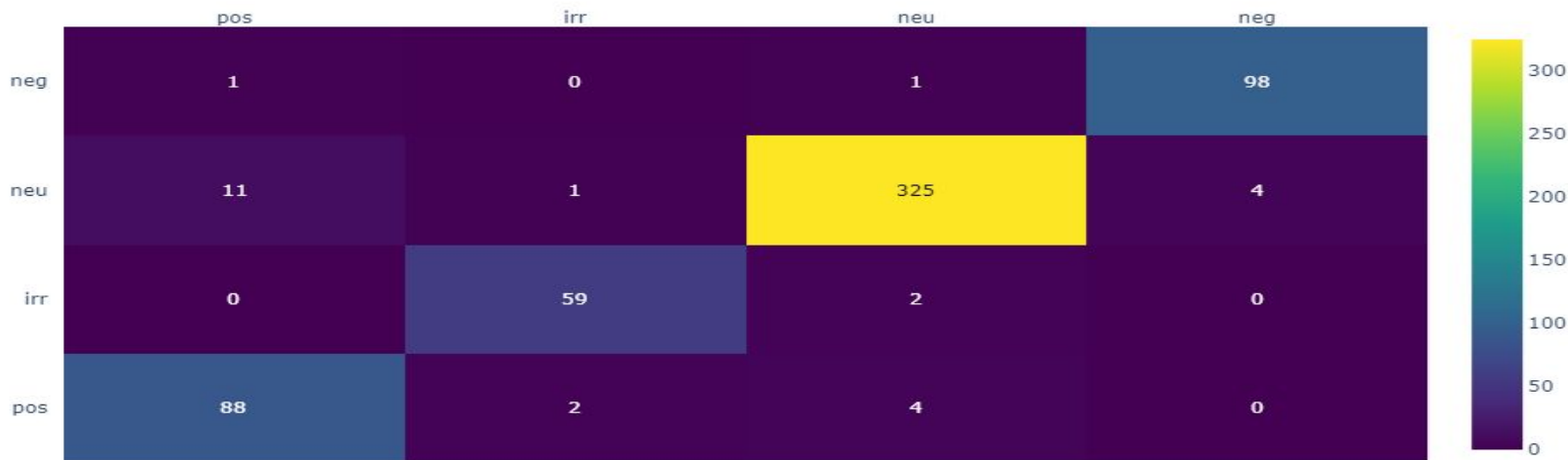
BERT is the Best 😊

	precision	recall	f1-score	support
0	0.88	0.94	0.91	94
1	0.95	0.97	0.96	61
2	0.98	0.95	0.97	341
3	0.96	0.98	0.97	100
accuracy			0.96	596
macro avg	0.94	0.96	0.95	596
weighted avg	0.96	0.96	0.96	596

BERT is the Best 😊

Le meilleur modèle est le BERT: nous avons obtenu une accuracy de 91% sur les données d'entraînement et 92% sur les données test. Nous pouvons alors visualiser la matrice de confusion pour prouver la performance de notre modèle sur les données test.


Confusion matrix on Test Data





06

Conclusion

A decorative graphic on the left side of the slide consisting of two overlapping squares. The bottom square is a dark blue, and the top square is a lighter, medium blue.

Les modèles doivent être capables de prendre en compte les liens entre les différents mots. Il se trouve que le passage de la sémantique des mots obtenue grâce aux modèles comme Word2vec et Bag of Words, à une compréhension syntaxique est difficile à surmonter pour un algorithme simple. Le BERT est la meilleure solution qui arrive à bien comprendre le sens du Tweets.