

## On a testé pour vous... Quel outil pour mettre en place un service d'analyse de sentiment sur vos reviews ?

Tout existe en la matière, et on peut rapidement se perdre dans la diversité d'options existant lorsqu'on veut simplement suivre sa réputation, et détecter l'insatisfaction client avant qu'elle ne grandisse outre-mesure. Aujourd'hui, on compare 3 possibilités allant du plus simple au plus adapté afin de vous aider à choisir l'option qui répondra le mieux à votre besoin !

Pour ce faire, nous avons ici eu recours aux outils proposés par la plate-forme Microsoft Azure, mais on trouve des propositions comparables chez tous les principaux fournisseurs de services basés dans le cloud.

Nous les avons comparés sur la base d'un jeu de données contenant 1.6 million de tweets, et autant d'étiquettes binaires reflétant le sentiment de chaque tweet. On a sélectionné au préalable un échantillon de 1600 tweets qui servira de base au comparatif des différentes méthodes.

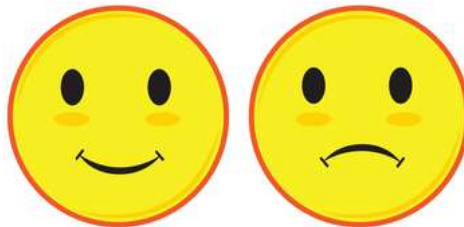


Fig 1 : on traite ici l'analyse binaire du sentiment, bien qu'il soit possible d'introduire une granularité. L'API Textanalytics offre cette possibilité en prédisant également un sentiment neutre.

### Niveau 1 : l'API Text Analytics

Microsoft Azure propose un service d'analyse de sentiment déjà intégré à sa plate-forme, consultable via une API de type REST. Ce service délivre directement des prédictions de probabilité correspondant à un sentiment positif, négatif ou neutre – ces prédictions sont probablement issues d'un modèle avancé d'analyse du langage. C'est on ne peut plus simple : on envoie ses textes, on récupère les prédictions. Le point faible étant que lorsqu'on utilise une boîte noire, on ne peut ni savoir ce qui se passe sous le capot, ni l'adapter à son propre besoin pour en améliorer les prédictions.

### Niveau 2 : le modèle sur mesure « no-code »

Sur le service ML Studio Classic, Microsoft propose un outil modulaire et ludique ne nécessitant aucune édition de code. On fournit son jeu de données, on câble des modules visuels que l'on choisit dans une liste de fonctions pré-établies, on règle les propriétés de chaque fonction, on exécute et on entraîne un modèle que l'on pourra déployer par la suite.

Les fonctions disponibles le sont évidemment en quantité limitée, mais elles permettent de construire un modèle adapté à ses propres données en un temps assez court et sans code, pour peu que l'on possède déjà quelques bases d'apprentissage machine.

Le modèle testé ici inclue donc le nettoyage du texte et la vectorisation/embedding par l'outil natif et l'entraînement d'un classifieur (modèle boosté à arbres de décision).

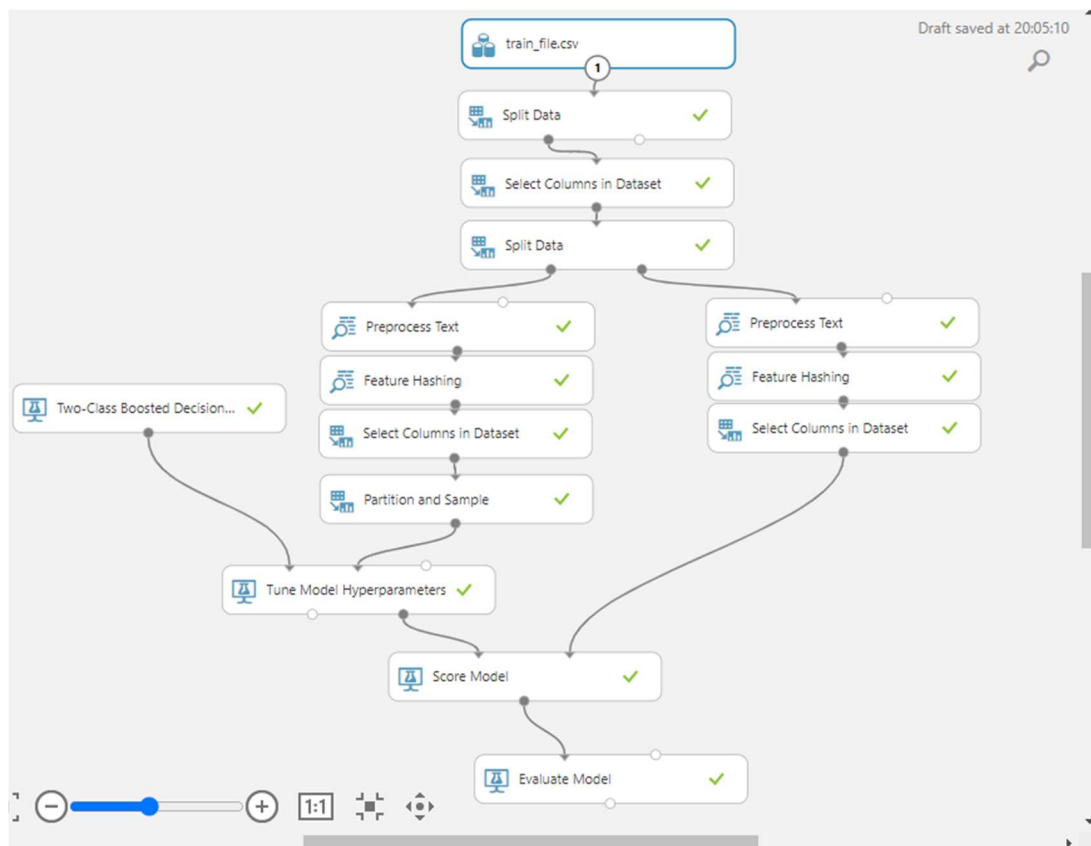


Fig 2 : traitements et entraînement d'un modèle de machine learning via interface graphique

### Niveau 3 : le modèle sur mesure avancé

On va ici recourir au code en Python et à la librairie Keras, autant pour définir les propriétés de notre modèle que pour l'entraîner et le mettre en œuvre dans le cloud Microsoft. L'avantage étant qu'il n'y a aucune limite à la variété et à la sophistication des traitements et des modèles utilisés, l'inconvénient étant que cela implique un investissement initial plus important, tant en temps humain qu'en ressources de calcul. La première étape consiste à développer un modèle prédictif qui offre un rapport complexité/performances en cohérence avec son besoin. La seconde implique de créer, entraîner et déployer son modèle dans un container du cloud Microsoft.

La variété d'outils et d'options peut être dérangement tant l'offre est pléthorique, mais elle permet à chacun de construire une application répondant exactement à son besoin, et offre une visualisation directe sur les métriques de performances choisies au long de l'apprentissage d'un modèle.

### Comparaison de performance des 3 modèles :

Afin de réaliser la comparaison des prédictions fournies par nos 3 modèles, nous avons établi les bases suivantes :

- Le modèle « API sur étagère » prédisant sur 3 classes, nous établissons un seuil à 50% sur la probabilité de sentiment négatif. On considère que toute probabilité supérieure ou égale à ce seuil équivaut à une prédiction de sentiment négatif.
- Les 2 modèles entraînés l'ont été en optimisant l'exactitude (Accuracy) des prédictions.
- Les prédictions de chaque modèle pour notre échantillon de test ont été récupérées – sur cette base, on génère les matrices de confusion et on compare l'aire sous la courbe ROC.

Les résultats d'évaluation des modèles sont donnés dans le tableau ci-dessous :

Modèle	API	ML Studio	Sur Mesure
Accuracy	0.714	0.732	0.821
AUC	0.709	0.741	0.831

### Conclusion :

Les 3 possibilités testées mettent en jeu des coûts de mise en place et de maintenance variables, et les performances des 3 modèles comparés sont évidemment proportionnelles à ces coûts.

Si votre besoin est ponctuel ou que l'erreur de prédiction n'est pas critique pour votre projet, nul doute que les modèles simples et pré-construits fournis sous forme d'API ou de blocs visuels sans code sauront répondre rapidement à ce besoin avec une pertinence raisonnable.

Si en revanche vous avez besoin de minimiser votre erreur ou de maintenir un modèle fonctionnel à plus long terme, alors il sera sans doute préférable d'investir et de développer un modèle sur mesure, qui une fois adapté à votre problème vous permettra de bénéficier d'une pertinence et d'une exactitude de prédiction accrues, et dont vous garderez la maîtrise au long cours.

En tous les cas, le traitement réservé aux données texte fera une différence majeure.