

Anticiper les bad buzz avec l'IA – Développement d'un prototype pour l'analyse de sentiments

Introduction

Les réseaux sociaux sont devenus une place publique où les utilisateurs partagent leurs opinions, bonnes ou mauvaises, souvent en temps réel. Si cela représente une opportunité pour les entreprises de se connecter à leur clientèle, cela constitue également une menace : un commentaire négatif peut rapidement se transformer en "bad buzz" et nuire à leur réputation.

C'est dans ce contexte qu'"Air Paradis", une compagnie aérienne, a souhaité mettre en place un système prédictif pour détecter les tweets exprimant des sentiments négatifs. Notre mission consistait à développer un prototype utilisant l'intelligence artificielle pour relever ce défi. Cet article explique en détail la démarche adoptée, les résultats obtenus et les outils intégrés pour garantir la fiabilité et la maintenabilité de ce prototype en production.

1. Comparaison des approches : entre simplicité et complexité

Pour répondre à la problématique, nous avons exploré trois approches distinctes : un modèle classique, des modèles avancés basés sur des réseaux LSTM, et un modèle de pointe utilisant BERT. Chacune de ces approches a été évaluée pour sa performance, son coût, et sa pertinence dans un cadre de production.

1.1. Modèle classique : une baseline solide

Le modèle classique, basé sur une régression logistique, visait à établir une ligne de base rapide et robuste. Avant de construire le modèle, les données textuelles ont été prétraitées pour garantir une représentation uniforme. Les étapes comprenaient la suppression des stopwords, la lemmatisation, et la vectorisation via deux techniques : **TF-IDF** et **Count Vectorizer**.

TF-IDF a attribué un poids à chaque mot en fonction de sa fréquence relative dans les tweets, tandis que Count Vectorizer s'est concentré sur le simple comptage des occurrences. Ces deux techniques ont permis de transformer les tweets en vecteurs numériques exploitables par la régression logistique. Bien que relativement simples, ces modèles ont produit des résultats satisfaisants, avec une précision atteignant 77,5 % et un roc_auc dépassant 0,85 pour TF-IDF.

Baseline Model Comparison: TF-IDF Vs Count Vectorizer			
	Metric	TF-IDF	Count Vectorizer
1	Accuracy	0.775	0.774
2	F1-Score	0.774	0.774
3	Precision	0.775	0.776
4	Recall	0.775	0.774
5	ROC_AUC	0.855	0.849
6	Training Time (seconds)	2.61	4.19

1.2. Modèles avancés : une exploration plus poussée

Pour aller plus loin, nous avons implémenté des modèles basés sur des réseaux de neurones LSTM. Ces modèles se distinguent par leur capacité à capturer les relations séquentielles dans les données textuelles, ce qui est essentiel pour comprendre le contexte dans un tweet.

Deux embeddings ont été testés : **Word2Vec** et **FastText**. Word2Vec génère des vecteurs en fonction des relations entre les mots, tandis que FastText intègre également des informations sur les sous-mots, ce qui le rend plus performant dans certains cas. Pour optimiser les résultats, plusieurs configurations d'unités LSTM ont été testées (64, 128, 256). Par exemple, avec FastText et 128 unités LSTM, nous avons obtenu une précision de 80,3 % et un roc_auc de 0,885.

Corrected Advanced Model Comparison: LSTM With Word2Vec And FastText						
	Embedding	LSTM Units	ROC_AUC	Validation Accuracy	Validation Loss	Training Time (second)
1	FastText	64	0.867	0.793	0.442	997.38
2	FastText	128	0.885	0.803	0.426	4469.36
3	FastText	256	0.869	0.794	0.439	6538.14
4	Word2Vec	64	0.876	0.793	0.442	2313.91
5	Word2Vec	128	0.868	0.791	0.444	1632.69
6	Word2Vec	256	0.868	0.792	0.443	5458.26

1.3. Modèle avancé : BERT, l'état de l'art

Enfin, nous avons exploré **BERT**, un modèle pré-entraîné révolutionnaire conçu pour comprendre le contexte complexe des phrases. BERT, grâce à son architecture de transformers, excelle dans l'analyse de texte. Cependant, sa mise en œuvre s'est révélée coûteuse en termes de temps et de ressources. Les résultats attendus incluent un roc_auc impressionnant de 0,933, mais le temps d'entraînement dépasse les 16 heures, rendant son déploiement difficile pour un usage quotidien.

Tableau des résultats de l'entraînement BERT :						
Époque	Training Loss	Training Accuracy (%)	Validation Loss	Validation Accuracy (%)	ROC-AUC	Training Time (s)
1	0.3839	82.57	0.3412	85.47	-	19535
2	0.2834	88.12	0.3437	85.79	-	19275
3	0.1948	92.34	0.3948	85.63	93.33	19847
Total	-	-	-	-	-	58657

2. Adoption des principes MLOps pour une IA durable

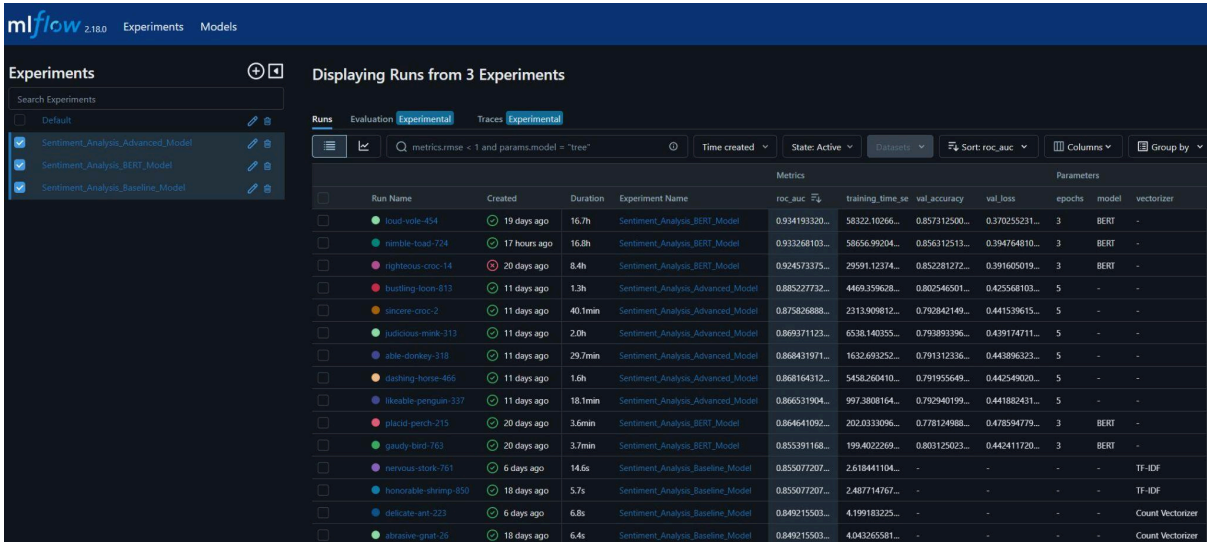
Une fois les modèles construits, la question de leur mise en production s'est posée. C'est ici que les principes **MLOps** (Machine Learning Operations) sont entrés en jeu. Ces principes combinent les pratiques de DevOps et les spécificités des projets d'apprentissage machine, pour garantir des déploiements fiables et une maintenance continue.

Un suivi rigoureux avec MLFlow

L'utilisation de **MLFlow** a permis de structurer le projet autour d'une gestion efficace des expérimentations. Chaque entraînement de modèle, qu'il s'agisse des baselines ou des modèles avancés, a été enregistré comme une "run". Cela inclut les paramètres d'entrée

(par exemple, type de vectorisation, nombre d'unités LSTM) ainsi que les métriques de sortie (accuracy, roc_auc, etc.).

Grâce à MLFlow, nous avons pu visualiser et comparer facilement les performances des modèles. L'interface graphique a révélé des tendances claires, comme la supériorité de FastText avec 128 unités LSTM dans les modèles avancés. De plus, les modèles ont été sauvegardés avec leur signature, garantissant leur compatibilité avec les données d'entrée.

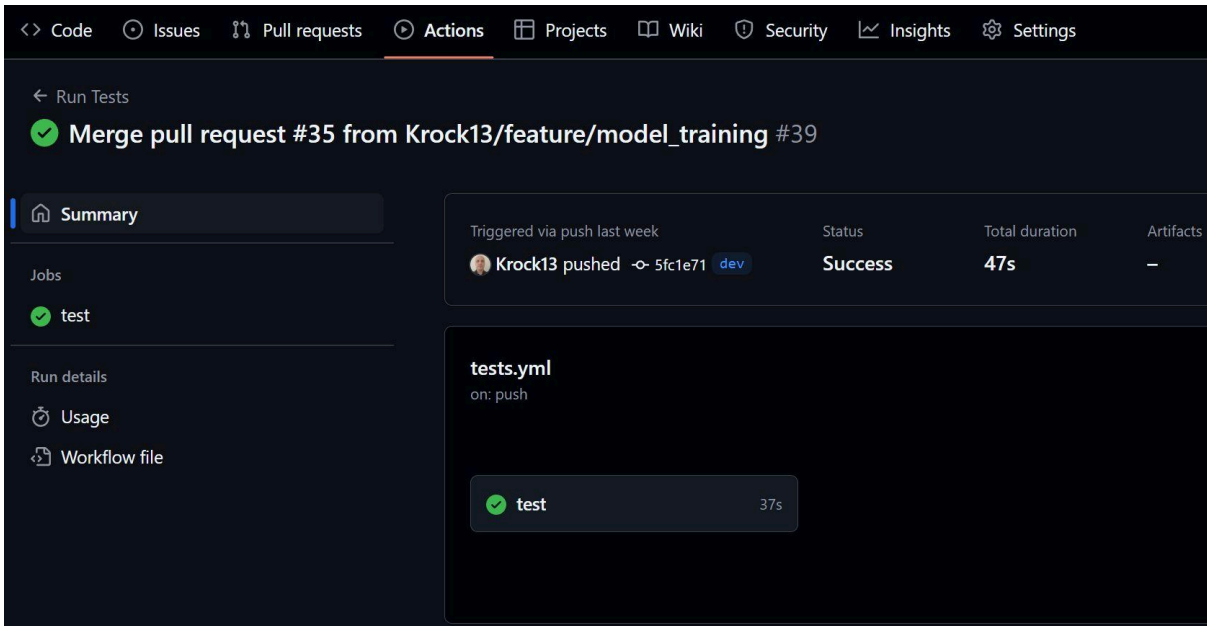


The screenshot shows the MLFlow Experiments interface. On the left, there's a sidebar with 'Experiments' and a search bar. Below it, three experiments are listed: 'Sentiment_Analysis_Advanced_Model', 'Sentiment_Analysis_BERT_Model', and 'Sentiment_Analysis_Baseline_Model'. The main area is titled 'Displaying Runs from 3 Experiments'. It features a table of runs with columns for Run Name, Created, Duration, Experiment Name, and various metrics (roc_auc, training_time, val_accuracy, val_loss, epochs, model, vectorizer). The runs are sorted by roc_auc in descending order. The first run is 'loud-vole-454' with a roc_auc of 0.934193320...

Run Name	Created	Duration	Experiment Name	roc_auc	training_time	val_accuracy	val_loss	epochs	model	vectorizer
loud-vole-454	19 days ago	16.7h	Sentiment_Analysis_BERT_Model	0.934193320...	58322.10266...	0.857312500...	0.370255231...	3	BERT	-
nimble-toad-724	17 hours ago	16.8h	Sentiment_Analysis_BERT_Model	0.933268103...	58656.99204...	0.856312513...	0.394764810...	3	BERT	-
righteous-croc-14	20 days ago	8.4h	Sentiment_Analysis_BERT_Model	0.924573375...	29591.12374...	0.852281272...	0.391605019...	3	BERT	-
hustling-loon-813	11 days ago	1.3h	Sentiment_Analysis_Advanced_Model	0.885227732...	4469.359628...	0.802546501...	0.425568103...	5	-	-
sincere-croc-2	11 days ago	40.1min	Sentiment_Analysis_Advanced_Model	0.875826888...	2313.909812...	0.792842149...	0.441539615...	5	-	-
judicious-mink-313	11 days ago	2.0h	Sentiment_Analysis_Advanced_Model	0.869371123...	6538.140355...	0.793893396...	0.439174711...	5	-	-
able-donkey-318	11 days ago	29.7min	Sentiment_Analysis_Advanced_Model	0.868431971...	1632.693252...	0.791312336...	0.443896323...	5	-	-
dashing-horse-465	11 days ago	1.6h	Sentiment_Analysis_Advanced_Model	0.868164312...	5458.260410...	0.791955649...	0.442549020...	5	-	-
likeable-penguin-337	11 days ago	18.1min	Sentiment_Analysis_Advanced_Model	0.866531904...	997.3808164...	0.792940199...	0.441882431...	5	-	-
placid-parrot-215	20 days ago	3.6min	Sentiment_Analysis_BERT_Model	0.864641092...	202.0333096...	0.778124988...	0.478594779...	3	BERT	-
gaudy-bird-763	20 days ago	3.7min	Sentiment_Analysis_BERT_Model	0.853391168...	199.4022269...	0.803125023...	0.442411720...	3	BERT	-
nervous-stork-761	6 days ago	14.6s	Sentiment_Analysis_Baseline_Model	0.853077207...	2.618441104...	-	-	-	-	TF-IDF
honorable-shrimp-850	18 days ago	5.7s	Sentiment_Analysis_Baseline_Model	0.853077207...	2.487714767...	-	-	-	-	TF-IDF
delicate-ant-223	6 days ago	6.8s	Sentiment_Analysis_Baseline_Model	0.849215503...	4.199183225...	-	-	-	-	Count Vectorizer
abrasive-grail-26	18 days ago	6.4s	Sentiment_Analysis_Baseline_Model	0.849215503...	4.043265581...	-	-	-	-	Count Vectorizer

Un pipeline automatisé

Deux pipelines GitHub Actions ont été configurés pour automatiser le déploiement. Le premier vérifie les fonctionnalités via des tests unitaires, tandis que le second génère un conteneur Docker et déploie l'API sur Azure Web App. Cette automatisation réduit les risques d'erreur humaine et accélère la livraison.

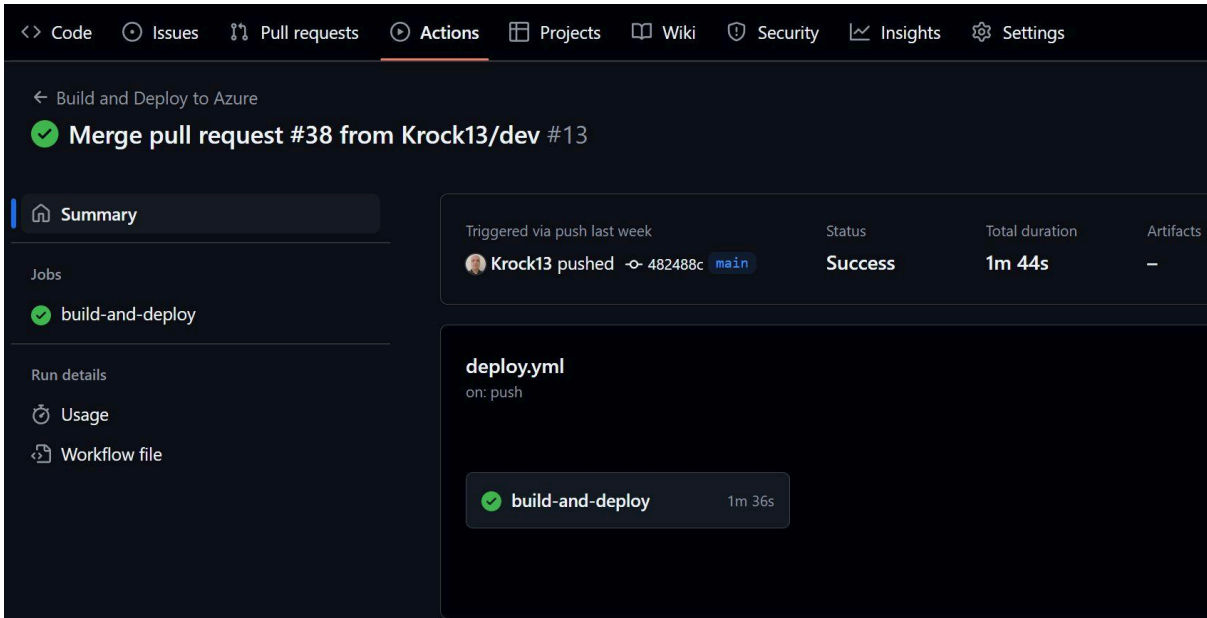


The screenshot shows the GitHub Actions interface. The top navigation bar includes 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The main area is titled 'Run Tests' and shows a workflow run for 'Merge pull request #35 from Krock13/feature/model_training #39'. The workflow is triggered via push last week and has a status of 'Success'. The total duration is 47s. The workflow file is 'tests.yml' and it contains a job named 'test' which has a duration of 37s.

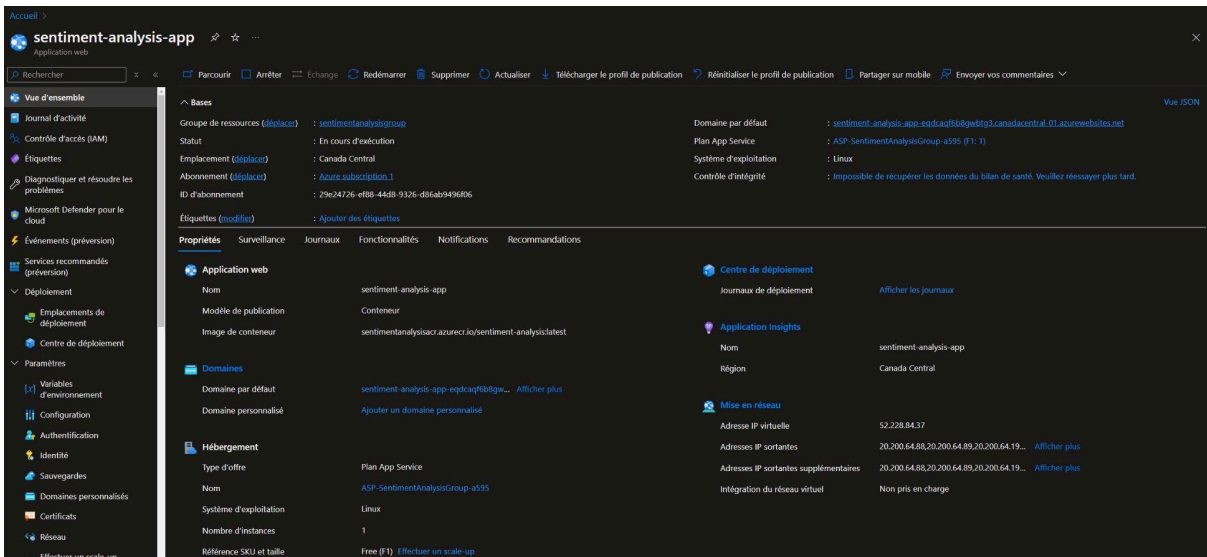
Triggered via	Status	Total duration	Artifacts
Krock13 pushed 5fc1e71 dev	Success	47s	-

Job	Duration
test	37s

La capture ci-dessus illustre l'exécution des tests unitaires via GitHub Actions. À chaque **push**, le pipeline vérifie automatiquement que les fonctionnalités de l'application sont conformes aux attentes. Ce processus garantit une base de code fiable avant toute étape supplémentaire, comme le déploiement.



Cette capture montre le second pipeline GitHub Actions qui prend en charge la construction et le déploiement. Une fois les tests réussis, un conteneur Docker est généré automatiquement et déployé sur Azure Web App. Cette étape permet une livraison continue du modèle en production, minimisant les délais et les risques liés à des déploiements manuels.



La capture ci-dessus montre le conteneur déployé sur Azure Web App. L'application **sentiment-analysis-app** est en cours d'exécution et accessible via l'URL générée par Azure. Cette capture confirme que le pipeline de déploiement automatisé a abouti à un modèle fonctionnel en production.

3. Perspectives et améliorations

Bien que ce prototype réponde aux besoins exprimés par "Air Paradis", il existe plusieurs pistes pour en améliorer les performances, la fiabilité et l'adaptabilité à long terme. Ces perspectives sont cruciales pour garantir que le système continue d'apporter de la valeur à mesure que les comportements sur les réseaux sociaux évoluent.

Analyse des Sentiments des Tweets

Prédisez le sentiment d'un tweet avec votre modèle.

Entrez un tweet à analyser :

I love this project !

Analyser le tweet

Sentiment prédit : **Positive** avec une confiance de 0.90

Prédiction correcte

Prédiction incorrecte

Instructions :

1. Saisissez un tweet dans le champ ci-dessus.
2. Cliquez sur **Analyser le tweet** pour voir le sentiment prédit.
3. Utilisez les boutons pour valider ou signaler une erreur dans la prédiction.

La capture ci-dessus présente l'interface utilisateur de l'application, développée avec **Streamlit**. L'utilisateur peut saisir un tweet dans le champ dédié et obtenir une prédiction du sentiment associé, accompagnée d'une mesure de confiance. Les boutons "Prédiction correcte" et "Prédiction incorrecte" permettent de valider ou de signaler une erreur, ce qui alimente le système de suivi en production pour une éventuelle amélioration continue.

3.1. Incorporation d'un système d'apprentissage continu

L'un des défis majeurs des systèmes prédictifs est leur capacité à s'adapter à des données nouvelles ou évolutives. Les tweets, en particulier, sont souvent influencés par des tendances ou des événements temporaires. Par exemple, une campagne publicitaire ou un incident peut générer des termes ou expressions inédites. Un modèle d'apprentissage automatique classique, entraîné sur des données historiques, pourrait avoir des difficultés à interpréter ces nouvelles données.

Pour pallier ce problème, une solution consiste à intégrer un **système d'apprentissage continu**. Concrètement, les tweets signalés comme incorrects par les utilisateurs via l'interface Streamlit pourraient être collectés et ajoutés à un ensemble de données d'entraînement enrichi. Ces exemples seraient ensuite utilisés pour réentraîner périodiquement le modèle.

L'automatisation de ce processus, via des outils comme MLFlow et GitHub Actions, permettrait de déployer automatiquement le modèle mis à jour en production. Ce cycle d'apprentissage continu renforcerait l'efficacité du système tout en réduisant le besoin d'interventions manuelles.

3.2. Adaptation des métriques en production

Les métriques suivies pendant le développement, comme la précision ou le roc_auc, sont importantes pour évaluer les modèles. Cependant, en production, des indicateurs plus spécifiques à l'usage réel peuvent s'avérer pertinents. Par exemple :

- **Taux d'erreur utilisateur** : proportion de prédictions jugées incorrectes par les utilisateurs via l'interface Streamlit.
- **Temps de réponse** : rapidité avec laquelle l'API retourne une prédiction, un aspect crucial dans un environnement de production.
- **Engagement des utilisateurs** : fréquence et volume des retours d'utilisation, qui peuvent indiquer la pertinence et l'utilité du modèle.

En intégrant ces nouvelles métriques dans le pipeline de suivi, il serait possible de mieux comprendre les performances en conditions réelles et de prioriser les améliorations nécessaires.

The screenshot displays the Azure portal interface for an alert rule. The left sidebar contains navigation options like 'Accueil', 'Journal d'activité', 'Contrôle d'accès (IAM)', 'Étiquettes', 'Diagnostic et résoudre les problèmes', 'Historique', 'Paramètres', 'Verrous', 'Identité (préversion)', 'Configuration des règles d'alerte', 'Gérer les groupes d'actions', 'Automatisation', 'CLI/PS', 'Tâches', 'Exporter le modèle', 'Aide', 'Intégrité des ressources', and 'Support + Résolution des problèmes'. The main content area is titled 'Multiple incorrect predictions' and includes a search bar and action buttons (Modifier, Désactiver, Dupliquer, Supprimer, Actualiser). Below this, the 'Bases' section shows details for the resource group 'SentimentAnalysisGroup', location 'Canada Central', subscription 'Azure subscription 1', and ID '29c24726-e108-44d8-9326-d86ab9496006'. The 'Étendue' (Scope) table lists the resource 'sentiment-analysis-app' under the hierarchy 'Azure subscription 1 > sentimentanalysisgroup'. The 'Conditions' table shows a rule with the name 'Count > 3' and an estimated monthly cost of '\$1.5'. The 'Actions' table lists the action 'Application Insights Smart Detection' with the description 'Envoyer un e-mail aux rôles Azure Resource Manager, 1 E-mail'. The 'Gravité' (Severity) is set to '2 - Avertissement' (Warning), and the 'Description' states 'Plus de 3 prédictions incorrectes sont signalées dans une période de 5 minutes'.

La capture ci-dessus montre une règle d'alerte configurée dans **Azure Monitor** pour suivre les prédictions incorrectes. Si plus de trois erreurs sont détectées en cinq minutes, une alerte est envoyée par e-mail. Ce mécanisme garantit une surveillance proactive en production, permettant d'identifier rapidement des dérives ou anomalies dans les prédictions.

3.3. Extensibilité à d'autres cas d'usage

Bien que ce prototype ait été conçu pour détecter les sentiments dans les tweets concernant "Air Paradis", son architecture peut être adaptée à d'autres cas d'usage et industries. Par exemple :

- **Secteur hôtelier** : analyser les avis des clients pour identifier les points récurrents de satisfaction ou d'insatisfaction.
- **Commerce électronique** : détecter les plaintes ou commentaires négatifs sur des produits pour améliorer le service client.
- **Secteur public** : surveiller les tendances sur les réseaux sociaux pour mieux comprendre les préoccupations citoyennes.

Cette extensibilité offre une opportunité pour "Air Paradis" de monétiser la solution en tant que produit technologique destiné à d'autres entreprises.

3.4. Gestion proactive des coûts en production

Le choix d'un modèle léger pour la production a été motivé par des considérations budgétaires, mais des techniques supplémentaires pourraient encore réduire les coûts :

- **Optimisation du modèle** : convertir les modèles TensorFlow en versions TensorFlow Lite pour diminuer leur taille et accélérer leur exécution.
- **Dimensionnement dynamique des ressources** : utiliser des instances cloud ajustables en fonction de la charge pour éviter le surdimensionnement des serveurs.
- **Compression des embeddings** : réduire la taille des vecteurs produits par les embeddings pour économiser de l'espace et des ressources de calcul.

En combinant ces approches, il serait possible de maintenir une infrastructure efficace et économique.

3.5. Enrichissement des fonctionnalités utilisateur

Enfin, pour améliorer l'expérience utilisateur, des fonctionnalités additionnelles pourraient être intégrées à l'interface Streamlit, telles que :

- La possibilité d'ajouter des commentaires explicatifs sur une prédiction jugée incorrecte.
 - Une visualisation des tendances des sentiments au fil du temps, permettant à l'utilisateur d'identifier des périodes critiques ou des campagnes réussies.
 - Un export des données collectées pour une analyse plus poussée par les équipes marketing.
-

Conclusion

Avec ces perspectives et améliorations potentielles, ce prototype dépasse sa fonction initiale de simple analyse de sentiments. Il devient un système évolutif, adaptable et orienté vers l'utilisateur, capable d'accompagner "Air Paradis" dans une gestion proactive de sa réputation sur les réseaux sociaux. En se tournant vers des approches d'apprentissage continu et en optimisant les coûts en production, cette solution ouvre également des opportunités pour des cas d'usage diversifiés.

Ce projet illustre non seulement l'efficacité d'une démarche combinant IA et MLOps, mais aussi l'importance de maintenir une vision stratégique pour pérenniser les solutions déployées. Grâce à une infrastructure flexible et un modèle évolutif, "Air Paradis" est mieux armée pour anticiper les défis futurs et renforcer sa relation avec ses clients.