

Suivi des performances et mises à jour en production du modèle LUIS

par Emmanuel Letremble

Table des matières

1. Critères d'évaluation.....	2
1.1 Évaluation du modèle à l'entraînement.....	2
1.2 Évaluation du modèle en production.....	3
2. Mécanismes d'évaluation.....	5
2.1 Évaluation de la <i>Precision</i> , du <i>Recall</i> et du <i>F1-score</i>	5
2.2 Évaluation de la <i>Satisfaction client</i> en production.....	7
2.3 Autres outils pour évaluer et améliorer notre modèle.....	9
3. Méthodologie de mise à jour en production.....	11
3.1 Utilisation des métriques d'évaluation pour les mises à jour.....	11
3.2 Modalités des mises à jour.....	12

1. Critères d'évaluation

Un modèle **LUIS** est un **modèle de classification**, mais il y a plusieurs niveau de classification à considérer ; d'un coté on demande à notre modèle d'identifier / **classifier l'intention** de la phrase qui lui est fournie, et d'un autre coté on lui demande également d'identifier / **classifier les entités** (*ville de départ ou d'arrivée, les dates, le budget...*) présentes dans cette même phrase.

1.1 Évaluation du modèle à l'entraînement

Pour évaluer un tel modèle, nous pouvons faire appel aux métriques habituellement utilisées pour les tâches de classification ; *Accuracy, Precision, Recall, Specificity, F1-score, ROC AUC, PR AUC*, puis les calculer indépendamment pour chaque intention ou entité dont on veut faire le suivi.

Et d'ailleurs c'est très exactement ce que propose le service LUIS qui via son portail ou via son API permet de calculer la **Precision**, le **Recall** et le **F1-score** de chaque intentions et entités du modèle à partir d'un jeu de données de test.

Nous allons donc faire comme l'équipe de LUIS et utiliser ces 3 métriques.

Pour rappeler ce qu'elles sont exactement, considérons la matrice de confusion ci dessous, où un **TRUE POSITIVES** est un exemple auquel on a donnée la bonne Intention ou la bonne Entité etc.

		Actual class	
		1	0
Predicted class	1	TRUE POSITIVE	FALSE POSITIVE
	0	FALSE NEGATIVE	TRUE NEGATIVE

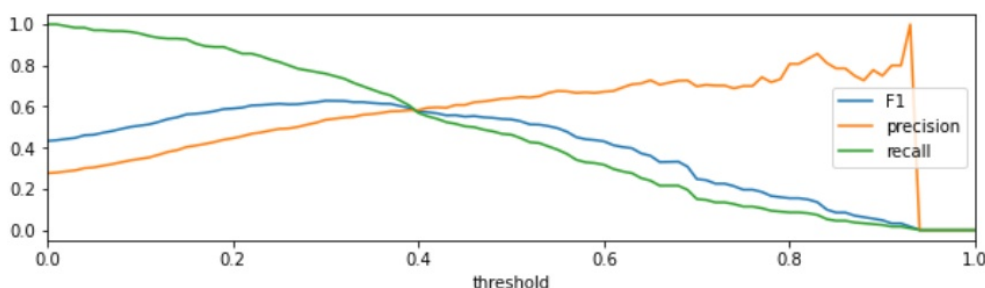
PRECISION : la fraction de **POSITIVES** prédits qui correspond effectivement à des **TRUE POSITIVES**. Autrement dit, quelle part des intentions ou entités prédites sont les bonnes ?

$$\frac{\text{TRUE POSITIVES}}{\text{TRUE POSITIVES} + \text{FALSE POSITIVES}}$$

RECALL : la fraction de **TRUE POSITIVES PRÉDITS** sur l'ensemble des **TRUE POSITIVES ATTENDUS**. Autrement dit, quelle part des véritables intentions ou entités ont été effectivement prédites ?

$$\frac{\text{TRUE POSITIVES}}{\text{TRUE POSITIVES} + \text{FALSE NEGATIVES}}$$

F1-score : la moyenne harmonique entre la **PRECISION** et le **RECALL**, et qui est donc plus facile à surveiller car elle cherche le meilleur équilibre pour optimiser ses deux métriques de références.



Inconvénients : pour être calculées, ces métriques nécessitent plusieurs conditions difficiles à réunir sur un modèle de chat-bot en production;

- ce sont des fractions, donc elles n'ont de sens que si elles se basent sur un ensemble de prédictions. *[mais notre modèle reçoit les énoncés/utterances non pas par lot, mais une à une...]*
- il nous faut connaître les véritables intentions et entités (*ground TP*) de chaque phrase collectée. *[la façon la plus sûr est de faire appel à des humains, mais ça exclut le temps réel...]*
- il nous faut théoriquement l'accord des utilisateurs pour collecter les énoncés/utterances.

Nous pouvons donc collecter les dialogues, puis les annoter manuellement avant d'en calculer les scores à intervalle régulier, mais c'est un processus coûteux en ressources que l'on ne peut pas faire en continu.

Ces métriques sont donc très bien pour évaluer nos modèles avant leur mise en production, mais il nous faut un autre moyen de surveiller les performances en production.

1.2 Évaluation du modèle en production

Comme nous l'avons vu, il est difficile d'évaluer en temps réel la qualité des classifications proposées par une chat-bot mis en production. Mais nous pouvons malgré tout essayer d'évaluer la qualité du dialogue au sens large en utilisant les retours clients.

Il existe de nombreuses façon de collecter directement ou indirectement les avis des utilisateurs qui interagissent avec le chat-bot :

- on peut tout simplement demander l'avis des utilisateurs en fin de dialogue.
- on peut aussi regarder le nombre d'utilisateurs qui reviennent pour un second achat / dialogue.
- on peut regarder la progression du nombre de dialogues initiés (*impact du bouche à oreille*)
- on peut regarder le rapport entre le nombre de dialogues initiés et le nombre de réservations.
- on peut programmer notre chat-bot pour lever des alertes dans certaines conditions.

Sur ce projet, nous avons choisi de mettre en place le dernier choix uniquement pour le moment.

Booking not confirmed : une alerte est levée par le chat-bot et envoyée sur Azure Insights lorsqu'un utilisateur ne confirme pas le résumé de réservation d'un voyage en fin de dialogue.

Please confirm, I have you traveling

- from: **Paris** to: **Dubai** on: 2022-10-23
- then from: **Dubai** to: **Paris** on: 2022-10-24

with a budget of **555** Euros

Yes

No

Misunderstanding × 3 : une alerte est levée par le chat-bot et envoyée sur Azure Insights après 3 phrases non comprises par le chat-bot (*Sorry, I didn't get that. Please try asking in a different way*).

Sorry, I didn't get that. Please try asking in a different way

Lorsque l'un ou l'autre de ces événements se produit, on demande à l'utilisateur son autorisation pour partager le dialogue complet avec notre équipe. En cas de refus, l'alerte est tout de même envoyée sur Insighs, mais sans l'historique de la conversation.

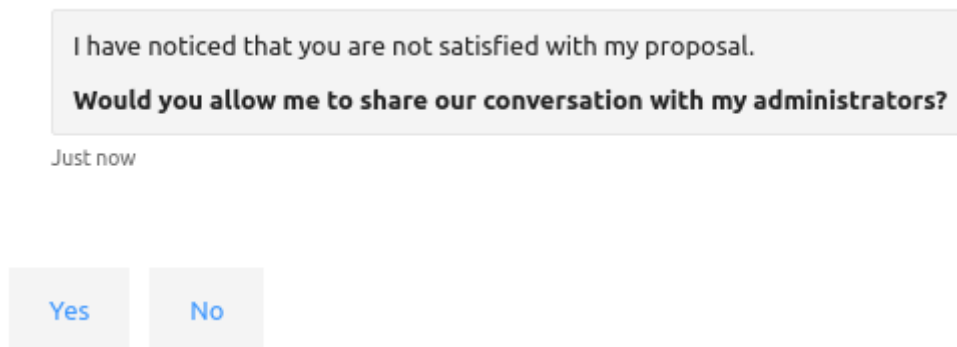


Fig-1: demande d'autorisation en cas de non confirmation d'un achat.

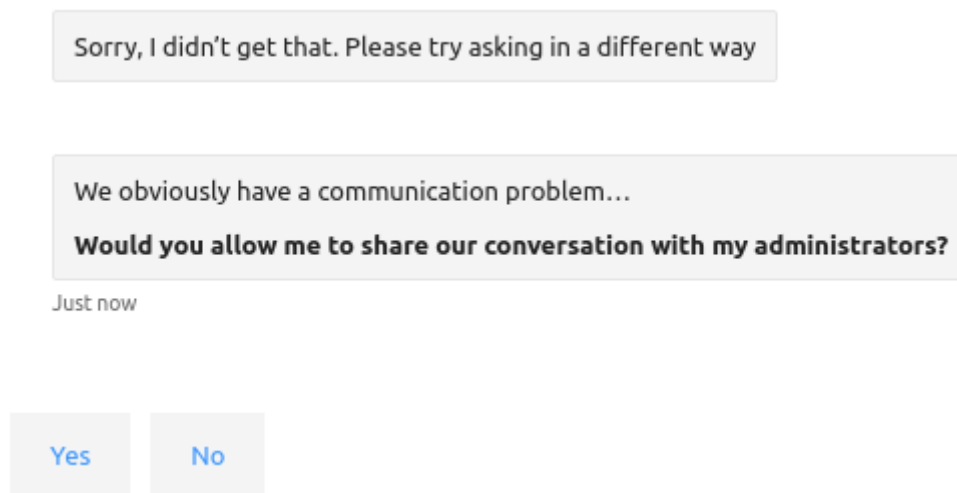


Fig-2: demande d'autorisation en cas d'incompréhensions répétées

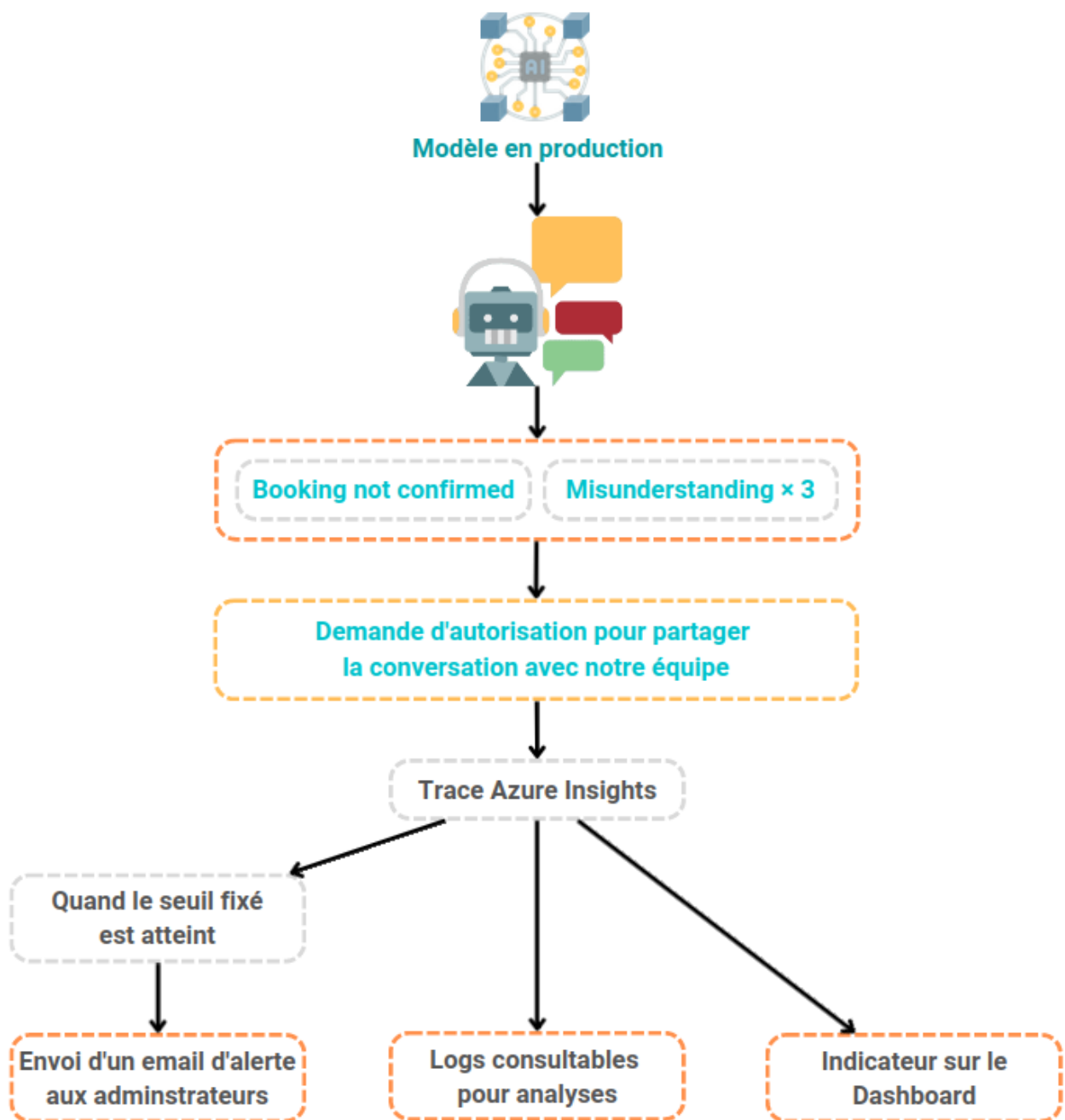
Nous avons donc pour le moment 4 types de traces sur Azure Insights :

- Booking not confirmed (with history)
- Booking not confirmed (without history)
- Misunderstanding × 3 (with history)
- Misunderstanding × 3 (without history)

Les traces contenant l'historique des échanges, nous permettent de reproduire le dialogue et de voir ce qui a mal été compris par le chat-bot ou l'utilisateur et donc de savoir comment améliorer notre prochaine version du modèle.

Les traces sans historiques, nous permettent d'avoir une idée de la satisfaction des utilisateurs dans le respect des [conseils de la CNIL pour respecter les droits des personnes](#).

Voici un schéma résumant le fonctionnement du système d'évaluation en production se basant sur les retours clients.



2. Mécanismes d'évaluation

2.1 Évaluation de la *Precision*, du *Recall* et du *F1-score*

Pour évaluer notre modèle LUIS sans passer par notre chat-bot, nous avons mis en place un **script Python** qui peut être appelé avec en paramètre un fichier JSON contenant des énoncés de test.

```
(venvLuis) >>> python evaluate.py --valid_path ../data/valid.json --app_id YOUR_APP_ID
```

Ce qui nous permet d'obtenir les *Precision*, *Recall* et *F1-score* des différentes **intentions**...

```
valid.json --app_id 84614f36-5e07-4725-a485-c255f41daac2
Push a batch of validation samples from ../data/valid.json
OK --> operationId: 54951a59-8e6a-4a2b-9a0e-3b8d91b67e28_638019936000000000
Evaluating .... done!
Fetch results
--> JSON:
{
  "intentModelsStats": [
    {
      "modelName": "BookFlight",
      "modelType": "Intent Classifier",
      "precision": 1.0,
      "recall": 1.0,
      "fScore": 1.0
    },
    {
      "modelName": "Greet",
      "modelType": "Intent Classifier",
      "precision": 1.0,
      "recall": 1.0,
      "fScore": 1.0
    },
    {
      "modelName": "None",
      "modelType": "Intent Classifier",
      "precision": "NaN",
      "recall": "NaN",
      "fScore": "NaN"
    },
    {
      "modelName": "Quit",
      "modelType": "Intent Classifier",
      "precision": 1.0,
      "recall": 1.0,
      "fScore": 1.0
    }
  ],
}
```

... mais également les scores pour les **entités** supportées par le modèle...

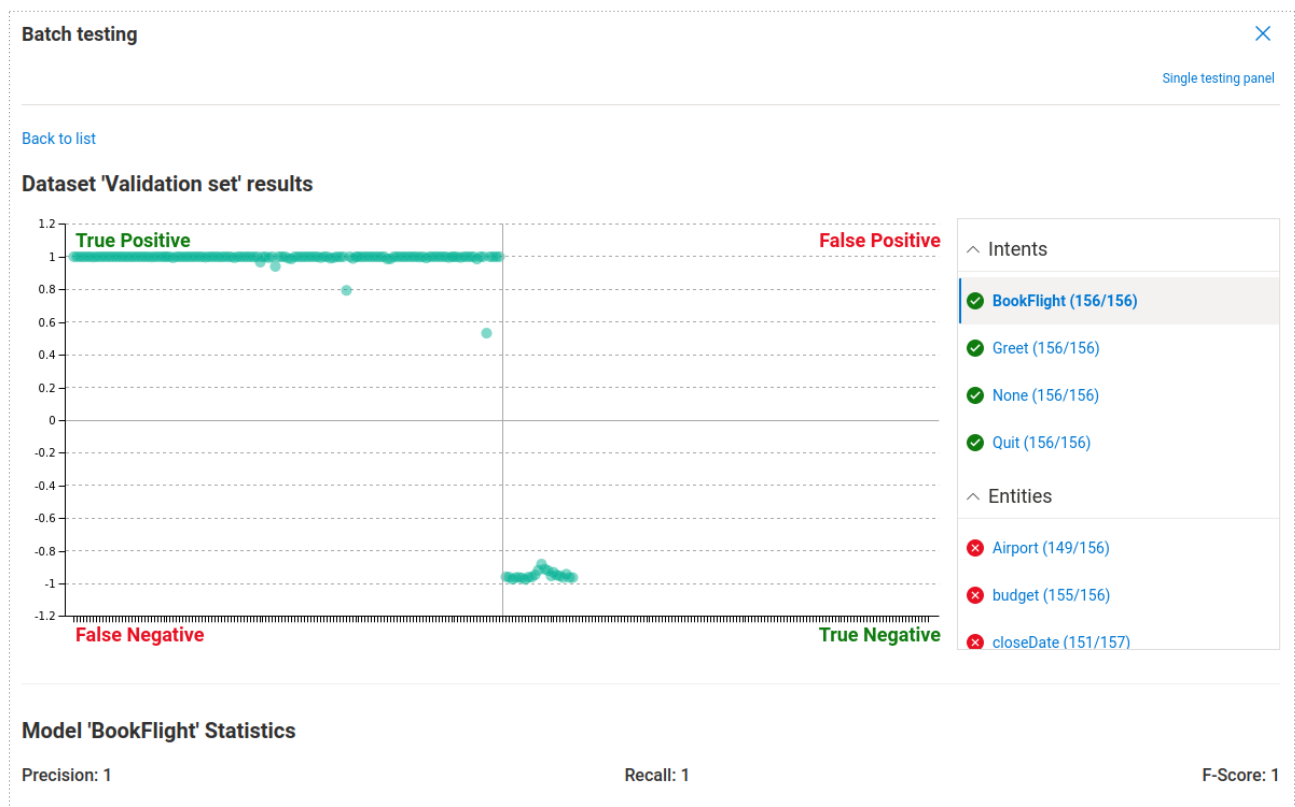
```
"entityModelsStats": [
  {
    "modelName": "budget",
    "modelType": "Entity Extractor",
    "precision": 0.95,
    "recall": 1.0,
    "fScore": 0.97
  },
  {
    "modelName": "closeDate",
    "modelType": "Entity Extractor",
    "precision": 0.71,
    "recall": 0.92,
    "fScore": 0.8
  },
  {
    "modelName": "openDate",

```

... et le détails par énoncés/utterances indiquant les faux positifs pour savoir ce qui ne va pas.

Similairement, le [portail LUIS.ai](#) nous permet d'évaluer notre modèle avec ce même jeu de données de test, mais avec un rendu plus graphique...

Ce qui nous permet là encore d'obtenir les *Precision*, *Recall* et *F1-score* des différentes **intentions**...



... mais également les scores pour les **entités** supportées par le modèle...




... et les détails par énoncés/utterances permettant de savoir ce qui ne va pas.

2.2 Évaluation de la Satisfaction client en production

Comme nous l'avons indiqué, le chat-bot a été configuré pour remonter des traces sur Azure Insights afin de [pouvoir réagir rapidement si le service pose problème](#). Et dans le cadre de cette première version, nous avons choisi un seuil d'alerte de 1 pour être prévenu très rapidement.

Ainsi, quand le chat-bot détecte un problème, une trace d'alerte est envoyée à Insights et dans un délai d'une à deux minutes, nous recevons [un email](#) nous enjoignant d'aller voir ce qui se passe.

 Microsoft Azure

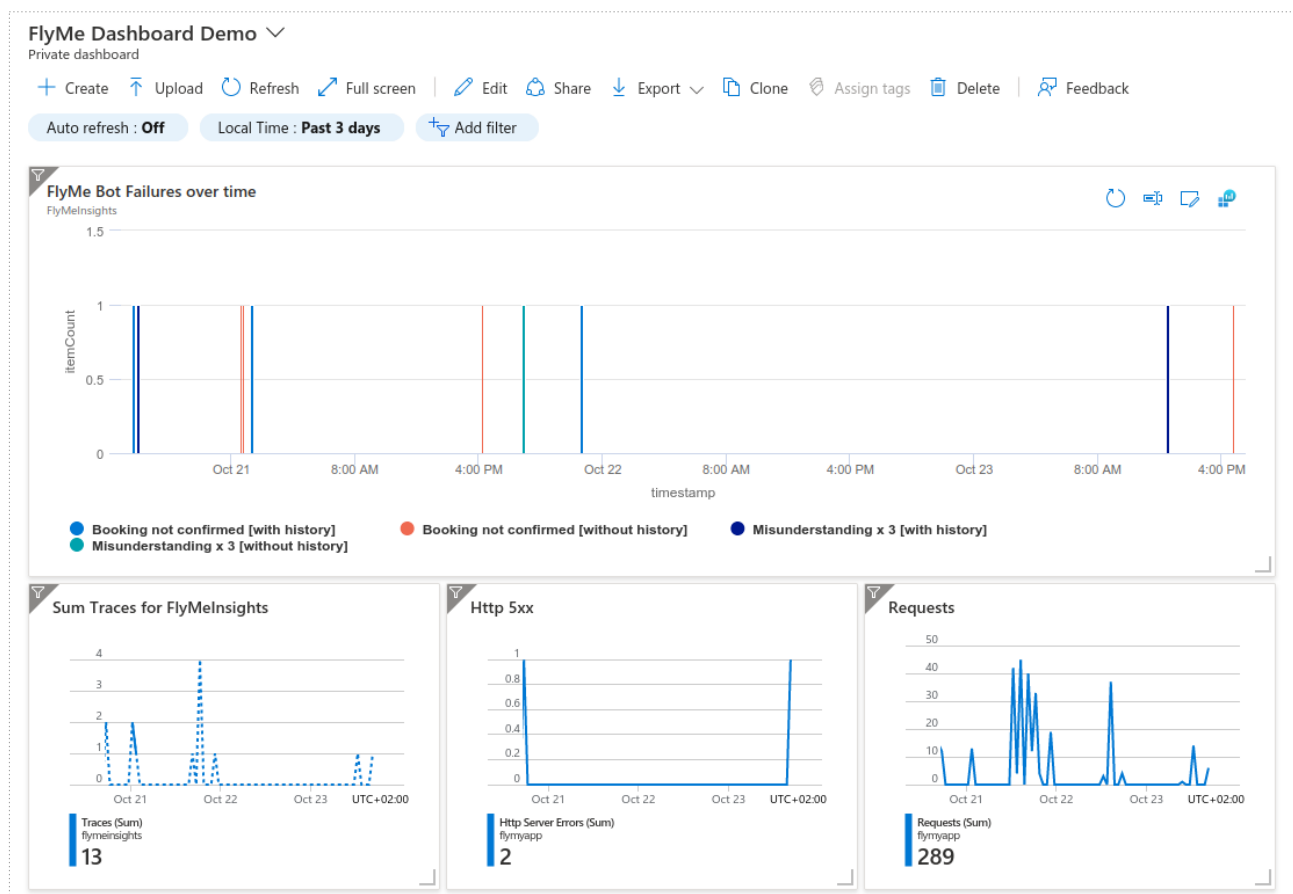
Fired:Sev2 Azure Monitor Alert Unsatisfied client on flymeinsights (microsoft.insights/components) at 10/23/2022 10:34:09 AM

[View the alert in Azure Monitor >](#)

Summary

Alert name	Unsatisfied client
Severity	Sev2
Monitor condition	Fired
Affected resource	flymeinsights
Resource type	microsoft.insights/components
Resource group	flyme

Simultanément, un indicateur apparaît sur [le dashboard](#) que nous avons configuré *(qui peut être partagé avec qui de droit, et qui propose d'autres indicateurs)*.



Les alertes peuvent ensuite être explorées selon divers critères...

Alerts

Create

Alert rules

Action groups

Alert processing rules

Columns

Refresh

Export to CSV

Open query

Search

Time range : Past 3 days

Subscription : Pay-As-You-Go

Add filter

More (4)

Total alerts

Critical

Error

Warning

Informational

Verbose

No grouping

Name	Severity	Alert condition	User response	Fire time
Unsatisfied client	2 - Warning	Fired	New	10/23/2022, 12:34 PM
Unsatisfied client	2 - Warning	Fired	New	10/21/2022, 10:39 PM
Unsatisfied client	2 - Warning	Fired	New	10/21/2022, 6:54 PM
Unsatisfied client	2 - Warning	Fired	New	10/21/2022, 6:54 PM
Unsatisfied client	2 - Warning	Fired	New	10/21/2022, 6:54 PM
Unsatisfied client	2 - Warning	Fired	New	10/21/2022, 6:54 PM
Unsatisfied client	2 - Warning	Fired	New	10/21/2022, 4:19 PM
Unsatisfied client	2 - Warning	Fired	New	10/21/2022, 1:24 AM
Unsatisfied client	2 - Warning	Fired	New	10/21/2022, 12:54 AM
Unsatisfied client	2 - Warning	Fired	New	10/21/2022, 12:44 AM
Unsatisfied client	2 - Warning	Fired	New	10/20/2022, 5:59 PM

... et pour chacune, les détails et éventuellement l'historique des échanges (si le consentement a été donné par l'utilisateur) peuvent être étudiés pour essayer de comprendre le problème.

Run

Time range : Last 24 hours

Save

Share

New alert rule

Export

Pin to

Format query

1 traces

Results

Chart

timestamp [UTC]	message	severityLevel	itemType	customDimensions
10/23/2022, 10:28:52.710 AM	Misunderstanding x 3 [with history]	2	trace	{'0':"Bot,What can I help you with today?";'1':"User,from Pa...
timestamp [UTC]	2022-10-23T10:28:52.71057Z			
message	Misunderstanding x 3 [with history]			
severityLevel	2			
itemType	trace			
customDimensions	{'0':"Bot,What can I help you with today?";'1':"User,from Paris to London";'2':"Bot,When will you start your travel?";'3':"Bot,What ca			
0	Bot,What can I help you with today?			
1	User,from Paris to London			
2	Bot,When will you start your travel?			
3	Bot,What can I help you with today?			
4	User,from Paris to London with a budget of 500£			
5	Bot,When will you start your travel?			
6	Bot,What can I help you with today?			
7	User,Hi			
8	Bot,Well, hello there! 😊			
	How can I help you?			
9	User,Can I go from paris to dubai with 555€			
10	Bot,When will you start your travel?			

Ensuite, on peut tenter de reproduire l'échange avec LUIS.ai pour mieux comprendre comment il classifie les intentions et entités... et même comparer l'impacte des éventuelles modifications apportées au modèle par rapport à celui qui est en production.

Test

[Start over](#) [Batch testing panel](#)

Type a test utterance ...

a flight from berlin to tokyo please. my travel window is from sep 1st 2022 to oct 1st 2022, with a budget of 5000€

BookFlight (1.000) [Inspect](#)

can you i go to paris tomorrow ?

BookFlight (0.997) [Inspect](#)

where can i go starting from milan ?

BookFlight (0.997) [Inspect](#)

can i go to paris with 250€ ?

BookFlight (0.998) [Inspect](#)

book me a trip from paris to london with a budget of 500€

BookFlight (1.000) [Inspect](#)

Version: 0.1

[Start over](#) [Compare with published](#)

User input

book me a trip from paris to london with a budget of 500€

Top-scoring intent

BookFlight (1.000) [Assign to a new intent](#)

ML entities

☐ Debug required features ⓘ

From

paris

To

london

budget

500 €

Composite entities

No predictions

Other entities

Airport

paris

geographyV2

paris

Airport

london

geographyV2

london

money

500 €

Published (Staging)

[Additional Settings](#) [Show JSON view](#)

User input

book me a trip from paris to london with a budget of 500€

Top-scoring intent

BookFlight (1.000)

ML entities

From

paris

To

london

budget

500€

Composite entities

No predictions

Other entities

Airport

paris

geographyV2

paris

Airport

london

geographyV2

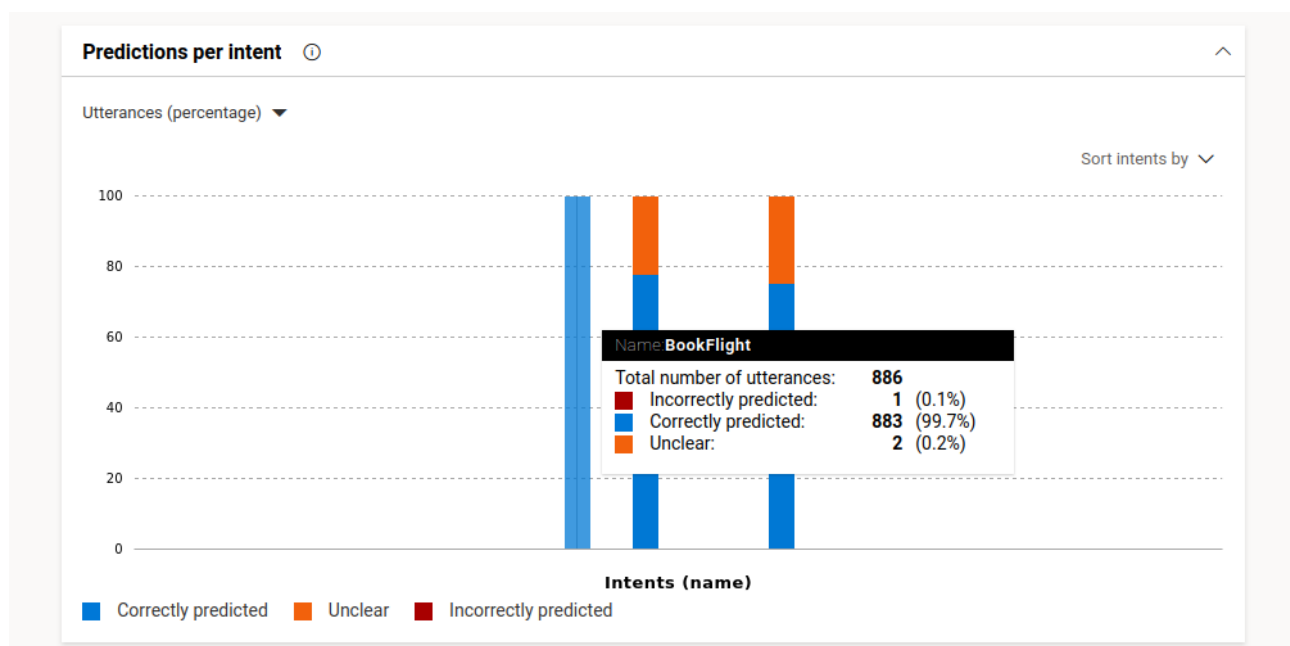
london

money

500€

2.3 Autres outils pour évaluer et améliorer notre modèle

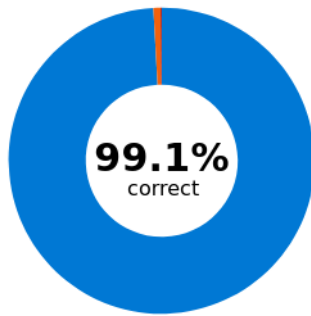
Notons que le [dashboard de LUIS.ai](#) propose plusieurs indicateurs basés le [training dataset](#), qui nous permettent d'identifier les sources de problèmes de façon très précise et qui ont par ailleurs l'avantage de [donner de nombreux conseils](#) pour améliorer le modèle.



Training evaluation ⓘ

Active version: 0.1 – trained Oct 21, 2022 12:29:11 AM

Overall predictions ⓘ



■ 99.1%	Correct predictions ⓘ
■ 0.1%	Incorrect predictions ⓘ
■ 0.8%	Unclear predictions ⓘ
4	Intents
11	Entities
908	Utterances

PROBLEMS AND SUGGESTIONS

Data Imbalance

These intents had many fewer utterances than other intents in your app, which can weigh predictions away from this intent. Consider adding more utterances to those intents.

[None](#)
[Quit](#)

Incorrect predictions ⬇

These intents had the highest percentage of incorrect predictions. Consider revising the incorrectly predicted utterances in these intents.

[BookFlight](#)

Unclear predictions ⬆

These intents had the highest percentage of unclear predictions. Consider revising the unclear utterances in these intents.

[Quit](#)
[Greet](#)
[BookFlight](#)

Intents with prediction errors ⓘ

Filters: [most problematic intents](#) ▾ — %

[BookFlight](#)
[Greet](#)
[None](#)
[Quit](#)

HOW TO IMPROVE INTENTS

This intent contains fewer example utterances relative to other intents.

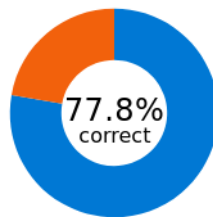
- Add more utterances to this intent.

This intent and its nearest intent(s) use similar words.

- Add more utterances to this intent using those words. This will reduce the weight of those words towards other intents.
- If two intents need to use similar words, but in different order, consider adding a pattern.

Greet 77.8% correctly predicted

ACCURACY



Total utterances	18
■ Correctly predicted	14 77.78%
■ Incorrectly predicted	0 0.00%
■ Unclear predicted	4 22.22%

Unclear predictions

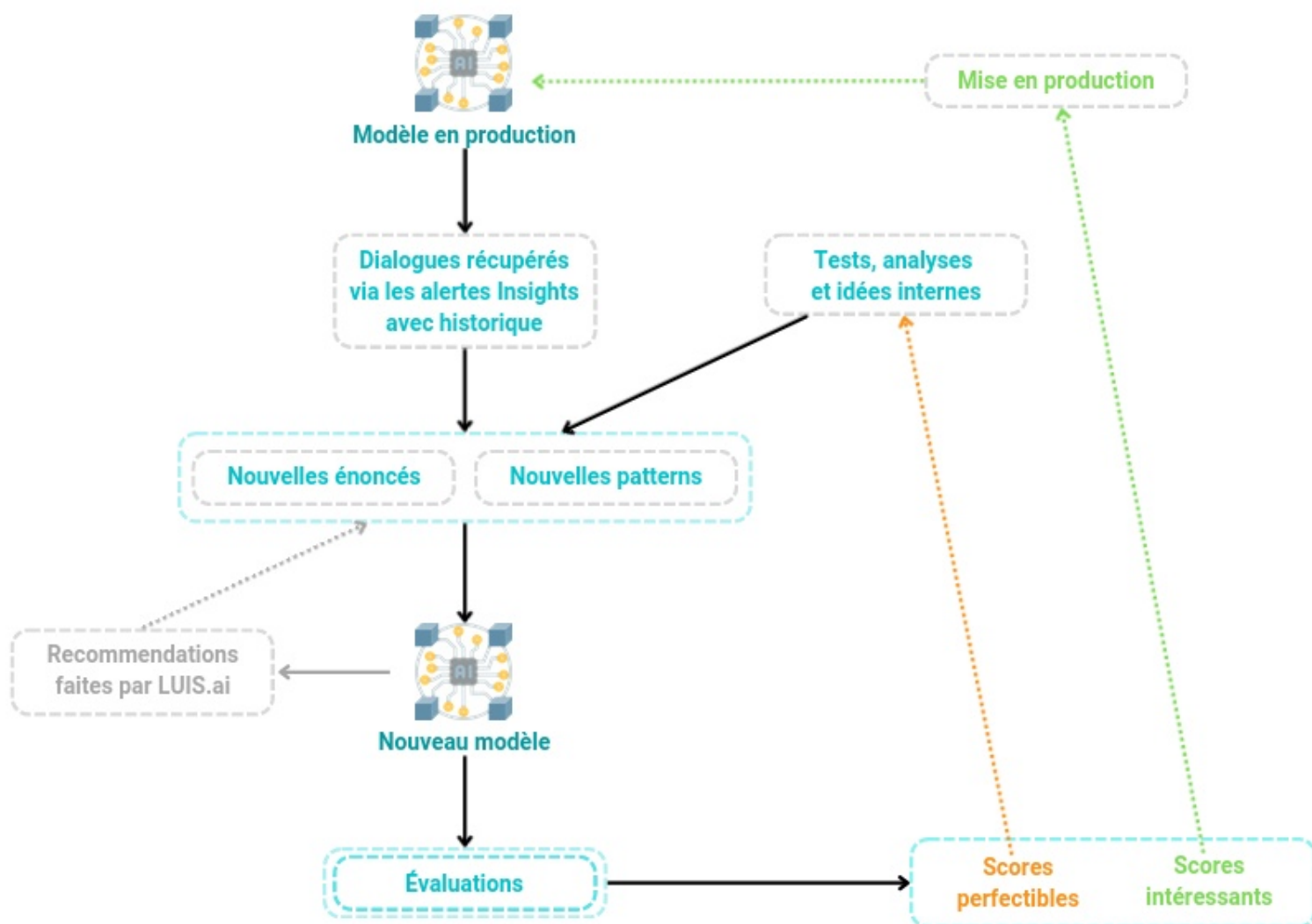
Many utterances in this intent were almost predicted as other intents. Consider making these utterances more distinct:

■ Almost predicted as Quit 4

3. Méthodologie de mise à jour en production

Pour éviter à notre chat-bot d'apprendre des réponses inappropriées il est préférable de **contrôler les nouveaux exemples** que l'on veut intégrer au jeu d'entraînement. (on a déjà vu des expériences de chat-bots apprenant en temps réel devenir des entités peu recommandables sous l'influence des utilisateurs qui prenaient le parti de lui enseigner n'importe quoi...).

Voici donc l'approche que nous pourrions utiliser pour mettre à jours notre modèle LUIS.ai



3.1 Utilisation des métriques d'évaluation pour les mises à jour

Les **évaluations du modèle en production** nous permettent de savoir si il est nécessaire d'envisager un nouveau modèle. Elles servent donc de déclencheur pour les mises à jours du modèle LUIS, et déterminent donc leur fréquence.

Les **évaluations avec les métriques de Precision, Recall et F1-score** nous permettent de nous assurer que le nouveau modèle que le nouveau modèle fonctionnent toujours bien sur le jeu de test (que l'on peut lui aussi diversifier de temps en temps). Il nous faut donc les calculer avant le déploiement de tout nouveau modèle.

3.2 Modalités des mises à jour

Les traces d'alertes envoyées par notre chat-bot à Azure Insights, ne sont pas forcément des indicateurs du fait que notre modèle ne fonctionne pas ; il est tout à fait naturel d'avoir des utilisateurs qui ne valident pas la transaction même si elle s'est parfaitement déroulée, ou encore des utilisateurs qui ne suivent pas correctement les instructions données par le bot (*réponses sans rapport avec la question, question sans rapport avec le business concerné ...*).

Nous ne devons donc pas forcément nous inquiéter de chaque signalement, mais il est malgré tout nécessaire de garder un œil sur le bon fonctionnement de notre projet.

Fréquence de calcul des performances : comme nous l'avons vu, il faut recalculer les scores avant chaque mise en production d'un nouveau modèle, et il ne sert à rien de faire des nouvelles mesures si l'on a pas produit de nouveau modèle ou au moins intégré de nouvelles énoncés au test set.

Plus le modèle sera abouti et moins nous aurons d'alertes remontées par le chat-bot. Il faut donc considérer que la fréquence des mises à jour du modèle LUIS devrait diminuer avec le temps.

Seuil de déclenchement des mises à jours : il est difficile à ce stade de déterminer un seuil de déclenchement approprié. Celui-ci va dépendre du nombre d'utilisateurs auxquels nous allons donner l'accès au chat-bot, et de la fréquence de son utilisation.

Plus vite nous aurons de nouvelles énoncés à analyser et intégrer au modèle LUIS et plus vite nous pourrons déployer un nouveau modèle. Mais si chaque nouvelle énoncés intégrée au modèle peut avoir son importance, il ne serait pas productif de déployer 20 modèles par jour...

On peut donc envisager un seuil correspondant à un nombre minimum de nouvelles énoncés collectées, ou encore un seuil se basant sur un nombre d'alertes par rapport au nombre total d'utilisateurs...