**Model performance Monitoring**

# Real-time Model Evaluation Criteria

The challenge of evaluating the performance metrics in production is that we **do not have access to the ground truth**. In this memo, I present the 3 types of metrics I have developed and logged to Azure Insights in order to provide real-time monitoring of the model performance.

For this project, 3 types of performance criteria in production have been identified:

* A user-provided metric
* Accuracy metrics
* Log metrics

## User-provided Metric

The simplest way to monitor the performance of our application is to ask the user for a score at the end of their interaction.

A **crucial limitation of this strategy is that users may decide to stop using the bot before reaching the end of a conversation,** thus not providing a score.

This can be evaluated by calculating the ratio between the number of dialogs (number\_dialogs) and the count of user scores (COUNT(user\_score)), both of these metrics are recorded in real time in Azure Insights.

The user score metric will thus provide an **optimistic estimate of the quality of the bot,** since users choosing to continue the dialog until the end and to provide a score are assessed to be more likely to be satisfied with the bot.

## Accuracy metrics

The term **accuracy** here is a bit of a misnomer because we cannot compute an actual accuracy since we do not have a ground truth, but I believe those metrics are a good estimate of the model performance in real time.

2 kinds of estimated accuracies have been computed and logged into Azure Insights:

* Total Accuracy (total\_accuracy) : Ratio between the number of successful predictions and the total number of predictions of the model. This metric is calculated by taking into account the confirmation prompt displayed after each LUIS prediction.
* Entity accuracy (entity\_accuracy): Ratio between the number of successfully predicted entities and the entities falsely predicted. This metric is an estimate because the current version of the bot does not implement a list prompt due to Python BotFramework limitations. The users are either able to select one wrong answer or a specific (multiple wrong answers) fields that will erase all data.

One important limitation to outline is that the bot is **unable to differentiate between bad predictions and wrong user input.** This means that if a user sends the bot a message with no booking information to retrieve, it will still send a 0-accuracy metric datapoint to Azure Insights.

It is thus important to outline that these metrics are actually **pessimistic estimates of the actual model performance.**

## Log metrics

After each LUIS prediction, I have implemented a logging of the query and of the entities returned by the model. This data can be used to compute several kinds of metrics:

* Actual accuracy: This could be done by asking employees to label the queries themselves and to compare the result to what is returned by the LUIS model.
* Per entity accuracy: In the same way, it would be possible to compute the accuracy for each entity as opposed to a global accuracy for all entities. This would allow **further refinement of the model by adding utterances** that would allow the model to better identify the entities with the lowest accuracies.

# Diagram of the Model Evaluation Mechanisms

All metrics

ACCURACY

MANUAL QUERY LABELLING

END OF DIALOG

USER SCORE

LOGGED QUERY AND RESPONSE

SUCCESS / ERROR METRICS

ESTIMATED ACCURACY

LUIS API Prediction

USER QUERY

Logged to Azure

Insights

# Model Retraining

Model retraining can be performed with the saved logs of good predictions (labelled “Good Prediction”).

The most efficient way of retraining the model however is to ask employees of the airline to manually label the logs of wrong detections (labelled “Wrong Prediction”) and to use these utterances to retrain the model.

* Retraining threshold : The model should be retrained when the **overall accuracy falls below 50% or the user score falls below 3**. Alerts have been created within Azure Insights to visualize when these thresholds have been crossed.
* Retraining frequency : Exact retraining frequency depends on the number of daily interactions, but the model should be retrained **every 10k predictions from the good predictions and every 1k manually re-labelled wrong predictions**.