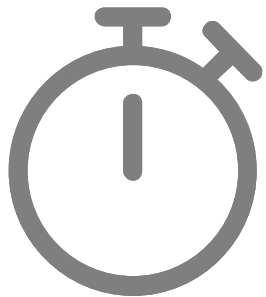




Potential User Interfaces for the Computational model

Carsten Behring
Data scientist

AGENDA



- — Model vs user interface
- — Three use cases
- — Data sensitivity
- — Three demos
- —



COMPUTATIONAL MODEL VS USER INTERFACE

- A computational model as discussed today as such cannot be used by end users, it requires programming to be usable
 - It is a function to be called giving it "text" and it returns a prediction (label + confidence)
 - Programming language specific, in our case "Python"
- A good practice is to wrap and deploy the model in an API and standardize its input / output being in JSON format, which makes it easily usable from any programming language
- We have done so, and the model as API is available to anybody at <https://efsa-innovation-01.efsa.europa.eu/efsafsnlpapi/api/1.0.0>
 - -> For easier usage from code, any programming language, and well documented
 - The model API does not store any submitted data, it takes a food fraud description and returns a prediction (and forgets the data)



USER INTERFACES FOR THE MODEL

- Given the availability of the model as API, it is now rather easy to create user interfaces, either via "programming" or "low-code coding tools such as Office365 PowerApps"
 - "Low code tools" allows to do this in very short time, couple of hours by Office power users
- For the moment we have identified three use cases for such user interfaces
 1. A user might want to try the model a few times with a few food fraud descriptions in order to get a feeling for it
 2. A user might want to validate the model with a larger number of food fraud descriptions
 3. EFSA might want to start setup an on going process which continuously uses the model on incoming food fraud description



USER INTERFACES FOR THE MODEL

- We will see demos for these three use cases in a moment
- All demos use the same "model instance" hosted as an API
 - The model is not "inside the tool", but get "called" from each of the tools
- All demos use the available tools / programming languages in Office365
- We will see concretely these three demos, one for each use case
 - A Office365 PowerApp for doing a single prediction on a single food fraud description
 - An Excel Online file having some code embedded which calls the model repeatedly and fills Excel cells based on model output
 - A workflow implemented using Office365 planner, which illustrates how a combined model/human workflow could be done



STORAGE OF SENSITIVITY DATA

- The computational model as such does not store any data
 - It takes the text, calculates and returns the prediction and forgets the text
- Some of the user interfaces are meant to store the data for longer, the workflow for example
- The user interfaces we see today are running concretely in the Microsoft Azure Cloud, paid by EFSA.
- Overall EFSA IT considers this a "production IT environment" , suitable for storing sensitive information
- In the demos today the access control to the food fraud descriptions is managed by KNOW, just using the usual Teams / SharePoint (= Office365) features for access control



THREE DEMOS

- Single prediction demo of an Office365 PowerApps
[prediction - PowerApp](#)
- Excel Online file with automatic cell filling based on model prediction
[predictFoodFraudFSConcern.xlsx](#)
- Workflow using Office365 planner where the model prediction annotates the planner tasks
[food fraud cases - planner](#)

