Alten Canada Data Science Technical Test

January 25

|  |  |
| --- | --- |
| Reviewer | Email |
| Antoine VILLATTE | Antoine.villatte@alten.com |

# General information

## Description

The objective of this project is to build a Tag recommendation system when posting a question on Stack Overflow.

The recommendation model must be a multilabel NLP model packaged into an endpoint function that takes a text as input and returns the recommended tags as either a list or a dictionary

## Expectations and goals

**Section A – Data retrieval**

The first task is to use the Stack Exchange Data Explorer to query the stack overflow database using SQL - [Query Stack Overflow - Stack Exchange Data Explorer](https://data.stackexchange.com/stackoverflow/query/new)

The query template is:



Modify this query so that you take only answers that:

* Have at least an answer
* Have 5 tags or more
* Order by descending favorite count, then view count
* Every tag that is like ‘Python’, ‘Python3’, ‘Python3.x’ is to be coalesced into ‘Python’

We know that the stackApi package can do that in python, but we specifically want you to do this in SQL so we can assess if you have basic knowledge in this language. For the same reason, please do the ‘Python’ mapping directly in SQL and not pandas.

**Section B – Model building**

Using this data, you will build a tag recommendation NLP model. You are free to choose the architecture and model, but the result must be multilabel, meaning that zero, one, or several tags can be output by the model if they are relevant.

We expect this work to not be too time-consuming, so we heavily recommend reducing the complexity of the data, for instance by filtering the tags to reduce their cardinality.

**Section C – Packaging**

The final step is to package your model into an entrypoint. You can make it either a script or an app, but we do not want a notebook.

For instance, it could be a script that runs like this:



If there are packages and dependencies to install to make the script run, please indicate them in a requirements.txt file.

Please share your model weights through a hosting service. We recommend Huggingface.

Please share your training scripts and performances on a testing set. These can be in a notebook if you prefer.

# Deliverables

* Deliverables can be shared in a GitHub repository (preferred) or zip file.
* SQL query in a text file
* Entrypoint in a .py script or an app
* Requirements.py and instructions if deemed necessary
* Processing and model building steps, as well as performance reports, in any readable format you like