

## Machine Learning Engineer / Conversational AI Engineer - Technical

In this test you will build a system that helps to classify simple conversations. This test can be treated as an opportunity to show skills and knowledge, as a learning exercise and as a prompt for further interviews in your process with DigitalGenius.

The test is split in two parts. Please complete both if time allows. There is a bonus exercise that is not required and will only be considered as a nice-to-have.

### Part 1

We have provided you with a small dataset containing customer service conversations for a company called *TopService*. The dataset contains conversations with two different topics - people who want to cancel an order at *TopService* or people who want to check an order's status. The dataset is very small and is just to demonstrate your solution - you should not have problems fitting even simple machine learning models on this data.

Use a machine learning model to predict what the customer is trying to do, either:

- cancel an order
- check an order's status, or
- something else.

### Part 2

Build a Flask app in Python exposing an API for the model in **Part 1** to respond to the first message in customer requests.

For example, the request body (in JSON format) might be:

```
{
  "message": "I want to cancel my order please",
  "timestamp": "2001-02-03T10:11:12",
}
```

And the response might be:

```
{
  "Prediction": "cancel_order",
  "prediction_confidence": 0.85,
  "suggested_message": "Of course. Please share your order number and account number with me and I will try to help."
}
```

You are not required to follow the above schemas - you may design new ones as you see appropriate.

If your system does not predict either `cancel_order` or `order_status`, make sure that nothing is suggested in `suggested_message` - i.e. return null.

## Notes

- Please write modular and idiomatic code.
- Please write unit tests for the vital functions of the application.
- You may make use of any Python libraries you wish to solve this problem (for example: Sklearn, NLTK, or regex).
- Please write a few short paragraphs describing how you would deploy this service to production.

## Bonus

Fit a Neural Network on the data using PyTorch, TensorFlow or similar for **Part 1**. This is definitely not required, so don't push it unless you're extra keen.