# Readme: Dataset and Task

## Objective

We are keen to understand how you would   
  
1. Build an ML model with text and ordinal data,

2. Design and build a RESTful API,

3. Dockerize the application.

We do not expect you to build a comprehensive solution, but keen to understand:

1. Your Python and machine learning skills (implement a simplified/essential version of a classifier to demonstrate your ML + Python skills),
2. Your skills on RESTful API and Docker.

## Data Specification

The dataset represents responses captured from job candidates in a screening assessment where they were asked to respond to a set of multiple-choice and open-ended text questions. (Note that the individual data values are synthetic, created with public domain data for the purpose of this exercise. You are free to make fair assumptions about the data.)

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Comments** |
| MCQ\_A10 | First Multiple-Choice Question | These are personality related questions where the responses are integers (ordinal) such as 1,2,3,4,5 where each number represents an answer from 5- Strongly Agree to 1- Strongly Disagree.  Example:  I get annoyed easily   1. Strongly Disagree 2. Disagree 3. Neutral 4. Agree 5. Strongly Agree     I like to spend time with other people   1. Strongly Disagree 2. Disagree 3. Neutral 4. Agree 5. Strongly Agree |
| MCQ\_B98 | Last Multiple-Choice Question | See above. |
| \_id | Response ID | Candidate Identifier |
| role\_family | Job category | Category of the job candidate has applied for. E.g. Sales, Retail etc |
| gender | Gender | Male/female/null |
| selected | 1= hired; 0=declined;  -1=information not available | Hiring outcome of whether the candidate was hired or declined. |
| text\_response | Text response given by candidates to open ended interview questions | Responses are separated by “|||” You can ignore or remove this. This is due to concatenating publicly available text for this exercise as sample data. Therefore, the text content may not be meaningful. |

## Task

1: Build a model in Python to predict “selected” based on both multiple-choice answers, and text responses.

1. We will not care about what you get for performance metrics such F1 score etc, so don’t spend time trying to improve on those (given the random nature of the data, those values will not mean much).

2: Build a Restful API in the local machine with the input of multiple-choice answer and text response, and the output of a likelihood of being selected

1. You might design the exact format of the input and output,
2. We will not care about the performance of this RESTful API such as response speed, throughput.

3: Dockerise the Restful API. When the docker application starts, it will start a local server with the RESTful API.

1. The whole implementation (task 1-3) can be simple (~5hrs effort)

4: Write a short description of the methodology and key results you would consider in a complete solution. For example: (1) Regarding the mode building, how would you increase the accuracy of the model. (2) Regarding the docker deployment, what are the services you would consider if you are to deploy the docker as an actual web service in the cloud, e.g., AWS. (3) How would you reduce the response time and increase throughput of the deployed service in the cloud?

## Effort

About 3hrs