alpas

Data Science/Machine Learning Engineer Challenge:

Designing a system for Entity Name Resolution

Remark: If you apply for the DS position, Task 1 is compulsory and Task 2 is optional. If you apply for the MLE position, both tasks are compulsory. Please bear in mind that we expect roughly the same time spent on the whole assignment (3-5 hours) whether you apply for the DS or MLE position, so please plan the level of detail of each task and the time between the tasks accordingly.

I. The Challenge

- Matching entity names is an extremely important task at Alpas since we are using hundreds of data sources containing different entity information.
- Company names from different data sources don't always match because of different word orders / spaces & special characters/ abbreviations/ typos/ changes in company type (GmbH -> AG) / prefixes and suffixes, etc.
- This is a common data science problem: some examples here and here
- We also need to deploy the model as a part of a data operation workstream, so that other operational components can use the produced results.

II. Data

- Please download the data from here.
- The data will contain a large list of companies with both positive and negative matches.

III. Task / Expected Outcome:

1. Using the given dataset, please suggest a model choice/algorithm to solve the problem of determining whether two names, i.e. Entity Name 1 and Entity Name 2 are similar. Please explain your choice accordingly and evaluate the effectiveness of the model in solving the task and a relevant

metric. Feel free to explore the outliers at the end and reason the pros/cons of chosen model choice/ algorithm.

Deliverable: Well-structured document (any format) explaining your steps and executable codebase (preferably Jupyter Notebook) that describes each step and short reasoning for each step in a readable manner.

2. Deploying and Designing MLOps workflow: **Select one of the two options 2.1 or 2.2 below**

   2.1. How would you deploy this model into production? Please develop a prototype to deploy it.

Deliverable: ideally you can deliver the solution in a python project structure that could be easily dockerized (and called via an API if possible). A simple workflow graph and document describing your decisions is welcome.

   2.2. Given that the operational goal of this model pipeline is to flag any named entity pair that is dissimilar for further quality assurance (QA), how would you design a system to train, test, and deploy the model to meet the expected goal?

Some additional background information about the setting: labelled training data is increasing weekly, the necessity for continuous model performance improvement and monitoring is crucial, and the frequent use of systems for experimenting with better algorithms is also a common practice.

Deliverable: A document containing the workflow graph/steps with explanation and tech stack suggestions within reasonable assumptions.

Additional Notes:

- The only solid requirement for the solution for task 1 is that **the method shouldn't be** rule / heuristics based. Rule based methods are really fast to get started with, but very difficult to maintain over time.
- In task 2, it is optional to choose one of the two tasks (2.1 or 2.2) based on the time you have available, but the answer for both is also welcome.
- Expected time for the whole assignment is around 3-5 hours, so we do not expect a perfect model or solution in both tasks, yet it needs to be executable when it applies. We would rather like to understand your thinking process, your past

experience to give certain reasonable assumptions and your logic, so please time yourself accordingly.

- Any further questions please feel free to reach out to luu.nguyen@alpas.ai, guilherme.msants@alpas.ai