Junior Data Scientist @ Tymit

You are given a data file in csv format. The content of the file is a series of tasks that our operations team had to do in the past weeks with some characteristics of these tasks. The two main things we are interested in for the exercise is resolution time (how much time a task is open), the workload (how many tasks are solved per unit of time, how many tasks are outstanding at a given point in time).

This exercise will allow us to have a general understanding of your technical knowledge and skills with regards to data. It is generally non-conclusive but gives us an idea of what are your sources of strength and what areas may need some improvement. If you make it to the next phase your solution will be discussed in some detail.

While we make sure that the exercise is self-contained and without unnecessary ambiguities, you may need to make a couple of assumptions. In those cases, please make them explicitly.

Questions

1- *Please deliberately focus on getting a quick-and-dirty solution for this question.* **Descriptive analysis:** Given the context above, pick 2 or 3 data representations (charts) that you believe could be interesting to understand the characteristics of the dataset. At least one of the representations should use a "derived" characteristic (a transformation/combination of the existing fields).

2- *Please deliberately focus on having a quality solution that will enable scaling (code clarity, reusability, comments for non-evident steps).* **Trend analysis/data management:** What can you tell in terms of the evolution along time of the tasks? How many tasks are generated per week? How many are resolved? What is going on with waiting times?

3- **Modelling and prediction**: Some data was missing in the initial set, for which we didn't have a time to resolution value. To estimate the real total workload, we need to estimate what was the resolution time based on the existing characteristics (same fields as the original excel, except for "Date_Resolution" and "time_to_resolution"). Build a predictive model to fill the gaps. This is not a Kaggle competition so don't worry about the perfect performance, but more about choosing a coherent model, and the process of how to build the different steps. If you have time, calculate performance metrics anyway. For those steps you don't manage to finish, at least explain with as much detail and specificity to the problem what steps you would have done with more available time.

4- **Communication:** Write a succinct summary of the learnings of the analysis as you would tell it to a non-technical person who is curious about what you've been working on (while doing this exercise).

Considerations:

- The ideal solution will come in a Jupyter Notebook[1] built in Python, as it’s a good combination of code visibility and well-presented results. If you prefer a different format, let us know in advance and we will also take it.
- You are not supposed to dedicate more than 2 hrs on this task, and it’s ok to not finish it at all (except for point #4, that one should be done in every case). Although, it’s best to try finishing every exercise even if it’s not perfect than leaving one totally empty.
- You can and are expected to incorporate popular data science libraries (Pandas, Matplotlib / Seaborn, NumPy, Scikit-learn, etc.)
- You can (and are expected!) to use the internet to find whatever help you need to work with the material.

1- Azure Notebooks (https://notebooks.azure.com) is a simple and free solution that gives you access to a Notebook set up. Either sharing the notebook or an open link to the notebook would work as submission.