CB+ | TEST FOR DATA PROFILE

This test is divided into two parts:

- Exercise #1: A data analysis
- Exercise #2: A small backend dev test using the Django framework.

Please do the exercises in this order, as you may not have the time to complete both of them.

Exercise #1: Data analysis

a. Exercise 1.a

Context, and problem to solve

ExpiryApp is a CB+ application that helps teams in stores to manage expiry dates of products. It relies on the following principle: at the init of the the application (its first day of use), the user creates an aisle (e.g. "Rayon des Yaourts"), sets alerting parameter for this aisle (e.g. I want to be warned 3 days before expiry), and for each product reference present in the shelves he.she saves in the app all the shortest expiry dates present. Then, the following days, the application asks the user to control the dates of products a few days before their last known expiry date. At each control, the new shortest date is set.

For example: If today is 27th of August, and if at the init of the store 1 month ago the user set that the expiry date of Danette was the 30th of August, then the application will ask him/her to control that product today (if the parameter set was expiry date minus 3 days)

The problem to solve is the following: when new product references are physically added in the shelves, the teams in store do not always recall to init these product references and to set expiry dates for them. Therefore, *ExpiryApp* does not know them, and products can be missed during the daily control routine.

Task

You will be tasked with analysing a provided set of data, to extract meaningful information and code the algorithm in python that would do the analysis to return the results in a shell program.

The analysis aims to determine if *ExpiryApp* successfully tracks all the meaningful products inside a specific shop, and if not, suggest references to track.

To perform this task, you will be provided with the following data:

- A fake set of products currently tracked by our system in a specific shop
- A fake extraction of the shop internal system which contains its vision of the existing references and the current shop assortment.

To help you: A product reference belongs to the assortment of a shop when it is supposed to be held by (and physically present in) the shop. A given retailer could have many references in its product references table (100k+), but only a fraction of those references belongs to assortment of each shop (10k max for fresh consumer goods).

From the attached csv files:

1. Determine what references which are in the shop assortment are not tracked by our *ExpiryApp*, and are meaningful to be tracked? (e.g : if no eggs have been initialised in *ExpiryApp*, it means that the team does not want to use the app to manage dates of eggs, and that therefore the algorithm should not propose eggs for initialization. However, if a new Yogurt flavour Pistache is physically present in the shop, and if other yogurts have been initialised in *ExpiryApp,* then the algorithm should detect the new Yogurt flavour Pistache to propose it for initialisation)
2. Bonus: For those references, include in the algorithm that it determines in which aisle should our users add these new product references.

Code the algorithm to analyse the data in python and display the result as a command line tool.
The command line tool should have the following features:
- open and parse the files
- return the result of the algorithm in a human readable form

Expected result
As soon as you read the instructions, please send us:
a) A time estimate for completing this exercise

After fulfilling the exercise, please send us:
b) The command line tool with instructions to run it
c) A link to your remote git repository (the repo needs to be public so we can clone it)
d) A detailed explanation of the analysis performed.
e) A short text explaining your technical decisions and possible areas of improvement of the data analysis.

b. Exercise 1.b

Context, and problem to solve
When a product close to its expiry date is found in the shelves, *ExpiryApp* helps teams in store to print discount stickers (-30%, -40%, -50%, etc.) in order to increase chances to sell the short date products and avoid sending them to garbage.

CB+ would like develop an algorithm to optimize the discount rate depending on the context. For example, it might not be useful to discount the price of a sandwich (even if close to its expiry) when it is the last sandwich of the shop and if it is 11am on a business day in a business district… On the other hand, if you have 100 units of Foie Gras expiring soon and the end-of-the-year season is over, 50% might be necessary (or even not enough), and it might be worth giving some units to charities and collect tax rebate associated to the giving.

Confidential – This document is CB+ property. Please do not share without our consent. – Contact@cb-plus.eu

Expected result

Describe how you would proceed to develop a tool enabling *ExpiryApp* to suggest teams in store the best discount option according to the context to optimize the benefits of the retailer and to minimize food waste.

What we want to know here is the methodology you would follow, the tools you would use, the step you would do and the time it would take to develop this tool.

Exercise #2: Backend dev

Please do the small backend dev test using the Django framework. Details of the test are available here: https://bitbucket.org/jdenais/django_mini_test

Good luck, and many thanks for applying for this CB+ job!