B2Wise Technical Challenges

Overview

What follows are two technical challenges, to be completed by candidates for a Senior developer position. The intent is for the candidate to spend a fixed amount of time (capped to a weekend) on these two challenges, followed by an in-depth discussion of the results. The discussion will cover the candidate's choices in terms of design, code, process and testability.

Please upload the code to Github or a similar source repository.

Challenge 1:

This challenge is focused on problem-solving ability. The solution should not be interactive. It should take a CSV file as input and output the final solution as a completed form of the CSV file.

The Problem:

Sudoku[1] is a grid-based logic puzzle that requires the player to complete a 9x9 grid partially filled with numbers. The puzzles range from basic to extremely difficult, with difficulty based on the number of missing entries in the grid.

Write a solver that takes a CSV containing 9 lines of 9 numbers (digits 1-9 or blank), solves the puzzle, and outputs the puzzle solution as a CSV in the same format as the input. You can use the puzzle on the first page of the Sudoku Wikipedia entry as a sample.

Part of the discussion will include how the difficulty of the puzzle influences the solution and why.

[1] https://en.wikipedia.org/wiki/Sudoku

Challenge 2:

This challenge is focused on design, coding style and testability. The solution should not be interactive. It should take several files as input and produce a CSV file as output. Additionally, it should produce a PNG graph of the output. Basic tests should be included, to demonstrate the types and manner of testing the candidate would usually produce as part of a completed task ('done').

Feel free to email questions to request clarification of the challenge.

The Problem:

Using Python/Pandas, implement a 'toy' version of B2Wise's single part processing 'pipeline'.

The spec:

- Using the provided files
    - Master.csv (Product Id, Active 1/0)
    - Sales.csv (Product id, Date, Sale amount)
- Do not output for any parts that are not active ('0' in the 'Active' field)
- Sum the sales into weekly buckets
- Generate a weekly moving average of the sales
- Use the moving average as a 'forecast', 52 weeks in the future
- For each week in the forecast:
    - Output the forecast amount
    - Assume that we start with 1000 units of a particular part (Stock on Hand), subtract the current week's forecast from the previous week's Stock on Hand, output this to the CSV
    - When the Stock on Hand is below 0, output a '0' to the CSV, otherwise '1'