PROGNOSIX

Forecasting & data product case study

Thank you very much for taking your time and interest in applying with us.

The following cases are not an "exam". They serve as examples of challenges that Prognosix meets, for you to see what could be expecting you. And they serve as an entry-point for the tech-interview in our application process.

Please note that you do not have to be an expert in the very specific tasks. What is much more important to us is your flexibility in thinking and a structured, optimistic approach to the challenges. We do not expect "perfect" solutions and we know that your time for preparation is limited. Since we are interested in real world problem solvers, please use the real world at your disposal to approach the cases (web, people, etc. as needed).

We are looking forward to discussing your "solution" / approach to the two case studies in the tech-interview.

Case 1: Forecasting daily sales and evaluating forecast quality

One of our customers wants to forecast sales of tomatoes on a daily basis (1-day-ahead), aggregated over approximately 100 stores in order to optimize supply chain planning. The data is stored in "tomatoesAndFeatures.csv".
Currently, the customer uses a simple forecast, he uses the sales one week before on the same weekday as forecast for the week to come (forecast for coming Tuesday = sales last Tuesday).
We were asked to build a more sophisticated forecast solution, involving promotions, weather forecast, holidays (and of course also weekdays and potentially some more features). The forecast accuracy should be compared to the customer's current methodology.
Based on the data in the attached csv file, prepare a forecast framework and its evaluation, prepare to explain your methods in very simple terms to the (non-tech) supply chain manager of the customer. This explanation should involve a number (how much better is your forecast than the one used by the customer), as well as a (simple) visualization of the timeseries, the customer's forecast and your forecast, and an idea on the importance of the features for the sales.

Hints

- Please use R or Python.
- Algorithms that allow you to include features as described in the task are for example regression tree, random forest, xgboost. Implement one of these (i.e. use the available R / Python packages) or try something that you consider interesting.
- The MAPE (mean absolute percentage error) is a measure suitable to indicate forecast error in an intuitive way. You can use a comparison of MAPE for the customer's forecast and your forecast or use ideas on your own (if you consider them more suitable).
- Split your data into training and test data (we suggest to use the last year in the data as test data). Evaluate on the test data. We advise to keep the split fixed, do not (yet) implement a rolling approach where training data would be extended with every new forecast day.
- If you want, you can extend the suggested features by more features, e.g. sales of same weekday in last week, yesterday's sales,…
- The feature importance can be assessed from regression tree / random forest results. You can plot the respective importance to get a comparative view of the importance-distribution over the features.

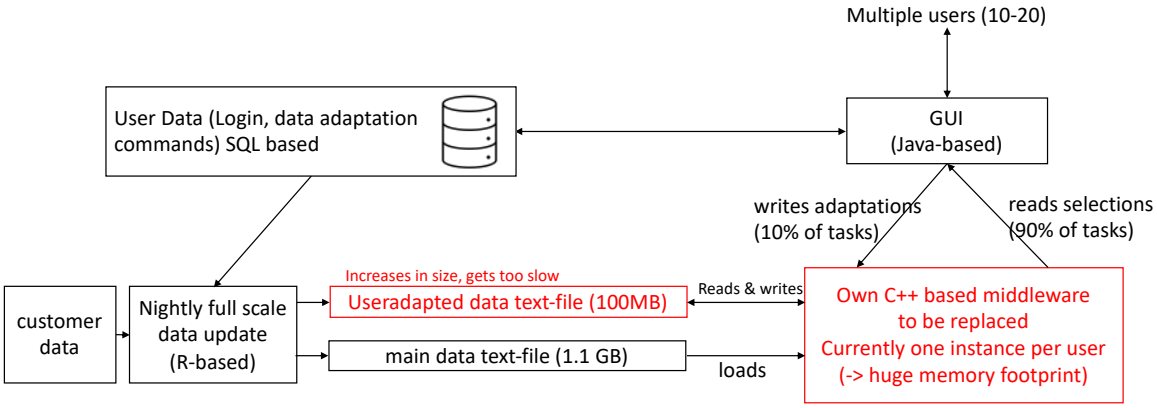
This could be a difficult task (if you are not yet used to the tools). Please let us know in the tech-interview about the challenges you experienced. We are more interested in intellectual flexibility than in perfect mastery of the tools.

PROGNOSIX

Case 2: Data product with real-time aggregation

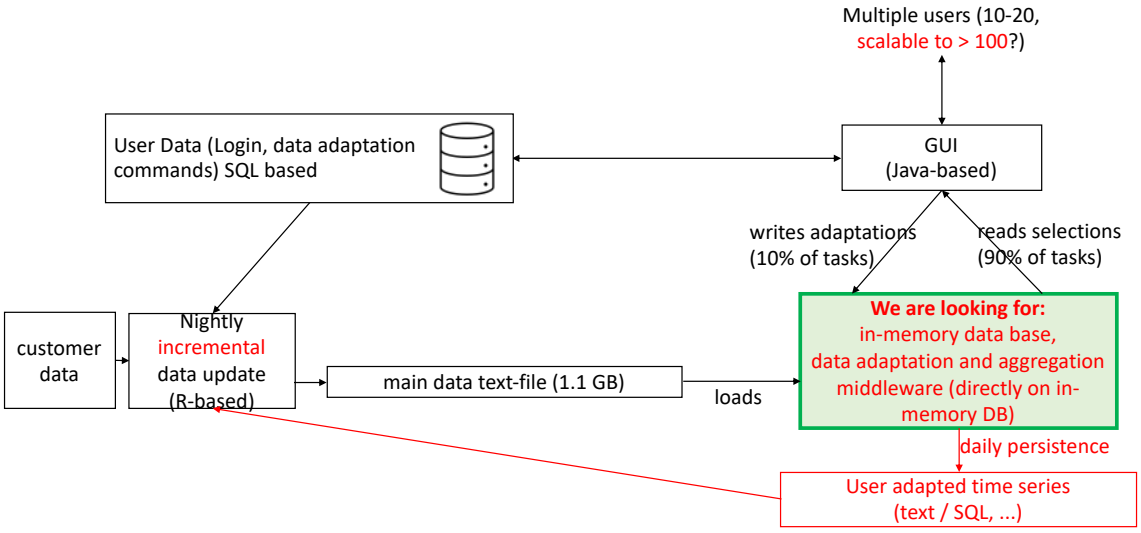
We need to visualize and adapt time-series data of 12'000 articles in realtime for several users (15 – later on > 100). They all share the same ground-truth. We need an in-memory DB with efficient memory footprint and very fast accessibility. To implement a realtime selection / adaptation processes (for example aggregate from daily to monthly data, aggregate all articles in category easter-products, …) we need some middleware between our Java-based Web-Application (screenshot to the left) and the in-memory DB. This middleware should be able to operate directly on the in-memory DB (no extensive data-flows out of in-memory DB in operations).

Details current workflow (mutli-instance)



The „useradapated data text-file" is used for communications of updateds between the single user-instances. It is reloaded after every user update.
The process is fully reconstructible from scratch every night (User data + customer data).

Details desired workflow with in-memory solution



The process remains fully reconstructible from scratch (for emergencies), regular updates are only incremental.

Task: This is a difficult challenge for us, and we do not yet have the optimal solution ourselves. We mainly want to discuss it with you in the tech-interview. Please prepare a short, informal structured approach to discuss this issue with us, mainly involving questions (e.g. what data, why approach X, bottlenecks with current approach, etc.). If you have experience with technologies for real-time aggregation, you can bring these in, of course.