Technical Case Study: Advanced Retail Forecasting Using AI

Introduction:

We were approached in the summer of 2021 by a large retail chain. Their objectives were to: enhance their current forecasting methodology to significantly reduce waste and increase profits.

Given the scale, with a spread across 500 unique item ranges in over 1,000 stores, even a small increase in the accuracy would account for an enormous decrease in the amount of waste and the enormous increase in profit, by reducing out of stocks, and increasing customer retention.

Background:

Previously, the retail chain utilized the triple exponential smoothing algorithm for forecasting. Despite already producing decent results, the system was not without its flaws. It struggled to consider more granular variables that influenced customer buying patterns such as the weather and promotional periods. It also struggled in the seasonal periods where buying patterns fluctuated far from the mean.

The Challenge:

We had to design a solution that:

- Considered more specific variables influencing purchase behaviours.
- Showcased heightened accuracy for forecasts, especially with relation to seasonal sales.
- Operated with minimal human intervention.
- Efficiently produced a daily quota of 7 million forecasts (500,000 unique Store Item pairs for 14 days in advance) in under 4.5 hours.
- Enabled prompt forecasting for new item introductions or store additions.
- Executed 7 million daily forecasts with acceptable server costs.

The Solution:

Our strategy pivoted around powerful AI agents built for speed and scale. During our discovery, we found that we would need to build models on an individual Store/Item level to maximise accuracy. Bananas in a store in Central London would be affected differently by the variables as Bananas in a store in Liverpool, for example.

To capture this effectively, 500,000 individual AI models would need to be generated and queried individually. We could do this using this tool, as well as accurately forecasting with as little as 14 data points for this specific use case, which would minimise the downtime for new Items/Stores.

Key Attributes:

Within our discovery phase, we evaluated the most important attributes for this client, the system is infinitely configurable for different variables as needed. Due to the need for individual Store Item Pair models, we regularly had less than 700 days of Sales Data. Consequently, we could not select more variables than this as we ran the risk of having models that are too sparse for the number of variables.

Addressing Time Series:

Once we had chosen our attributes and settled on a tool that enabled us to fulfil the forecasting within time demands, we had to find a way to eliminate the time series' factor. A Bot Named SUE could not take a date as an input, and as time series is a critical element that impacts retail sales, we could not accurately train the models with sales data that was over a few months old as customer buying patterns change year upon year. The factors that influence buying patterns for specific Item's won't change as much as the total Sales for the specific Item.

To get around this, we used a combination of Sales Per Hour, along with a Trend. We'd reduce the final Sales number for a day to a Sales Per Hour figure, and then we would cross reference this vs the average Sales Per Hour of the previous month to get a Sales Per Hour Modifier. These modifiers were the forecasting output of the ABNS Tool. This way, we forecast the change in Sales Per Hour compared to the previous month's average.

Then we modify the previous month's average by the modifier and the multiple by the number of hours the Store is open to get the final Sales forecast. Solving the time series issue, being able to adjust for changing Store opening hours as well as providing more accurate seasonal adjustments. An example of this may be as follows:

We are predicting 22nd December. The previous months trend may be 10 Sales per hour. The modifier is high, being this close to Christmas, so let's say this is 3.1 – that's predicting 31 Sales per hour. We now look at the number of hours the Store is open for. If this is 14 hours, then we'd forecast 434 Sales for that day.

Out of Stock Challenge:

Due to the relatively small number of data points we had for each Store Item Pair, it was important that were able to use as much of this data as possible. Let's say for example we had 30 days of Sales data, but of those 30 days, the item went out of stock on 10 of them. We have suddenly lost 30% of our available data as we can't include the out of stocks in the training as it would not be reflective of what we should order in (an item sells out as 11am, but the Store is open till 10pm – we don't want to order in the amount that caused us to sell out early). If we order too few, we lose out on Sales.

To maximise the amount of data we must train the models, and to minimise the impact of out of stocks, we converted any Sales where the item went out of stock to an estimated Sales. We also didn't do this linearly; we implemented a solution that looked at the bell curve data of a specific item and extrapolated from there. Different Items have different Sales curves. If Bread had sold out at 2PM, there is a different curve to if Cheesecake had sold out at 2PM (The majority of bread sales are made early, then tail off. Cheesecake has a different pattern).

Using the bread sold out at 2PM example, we would look on the Bell Curve to see, on average what percent of total sales for bread had been made at this point. If it was 90%, then we'd increase the actual sales by 10%, and use that as the Converted Sales, from which we'd calculate the SPH modifier as discussed earlier.

Platform & Automations:

The whole solution was implemented within OutSystems. This enabled us to build extremely quickly and integrate with APIs seamlessly. We used a combination of Timers and LightBPT's (Lightweight processes that could run on 20 threads in parallel) which enables us to batch process the number of forecasts within the time limit, as well as ingesting new data each day, running housekeeping, generating new models. Automating the whole process as quickly and efficiently as possible.

OutSystems also enabled us to build front end screens where the data could be analysed, information imported (such as Historical Last Time of Sale Data and Bell Curve Data). Despite being built to be as automated as possible, we also built it to be extremely configurable. Using Site Properties that stored values such as the forecast length, the length of the Trend and the Warehouses we wanted to forecast for.

We also enabled the removal of certain data periods for Stores where the Sales Patterns would not generally be reflective of Sales Patterns going forward such as times during the COVID Lockdown, or specific Store issues such as roadworks happening nearby which would impact Sales figures for a short time.

Conclusion:

Our solution delivered an advanced forecasting system, characterised by its pinpoint accuracy and efficiency. The system excels in automation yet remains adaptable. Its optimised data processing ensures timely and accurate forecasting, even in the face of challenges like time series complexities and out of stocks. This robust system, combined with daily data validation and housekeeping, has helped in the retailer's journey toward minimised waste and enhanced profitability.