Abstract

This final project simulates a modeling team's capacity to build new statistical models and then monitor their performance over time. Key insights relate to time to complete a set batch of work, while monitoring adherence to key regulatory deadlines.

Introduction

I manage a team that is responsible for forecasting my bank employer's interest rate risk position today and at points in the future. To ensure we are making decisions based on sound analytics, we pay close attention to the reliability of our balance sheet forecasts. This work falls on my Modeling and Analytics team, who are collectively responsible for building and maintaining behavioral time series forecasts to project volume and rate for our bank's major product lines. Being in a highly regulated department (Treasury) in a highly regulated industry (Retail Banking), the team's work is tightly watched: models are constantly being monitored for reasonableness and are redeveloped on an agreed cadence regardless of performance.

This team is stretched, and the scope of models the team can appropriately manage covers an insufficient fraction of our balance sheet. In other words, given the current team size the Modeling and Analytics group is failing to deliver a robust set of models that covers the entire bank balance sheet. While we have a rough sense of how large the team needs to be, this final project simulation is another angle to validate our need to grow the team size.

The simulation runs two separate tracks that interact with one another. The first track is staff management, which controls the pace existing modeling staff attrite and that open positions are filled. The second track is model management, which controls where each of N models are in their respective "build / monitor / rebuild" phases and whether tasks are being completed by their dynamically-assigned due dates. The tracks interact because modelers are assigned to model management tasks, and insufficient staff will result in missed deadlines.

The simulation confirms the intuitive conclusions that a larger team a) can get a static number of models developed and in production quickly than a smaller team; b) will have extra capacity to do other interesting analyses once all models are developed. The simulation also demonstrates that ability to manage workload - defined as the fraction of model monitoring attempts completed successfully - is sensitive to boundary conditions. For example, a team of 10 or more achieve about 97% of monitoring attempts when managing 15 models, but success drops to 90% for a team of 8 and 15% for a team of 6.

Methods

The simulation is executed using SimPy, an open-source python package designed to support event simulation. SimPy is natively equipped with useful features that accelerate time-to-deployment of a simulated environment, including multi-threaded event tracking and priority queue and standard simulation objects like "resources". Each time increment was assumed to be one week, and simulations were run for 10 years or 520 weeks.

One piece of the simulation tracks the number of modelers available to work on models. Another piece tracks where models are in their build / monitor / rebuild lifecycle.

The staffing module takes as inputs the current staff level at the start of the simulation, and the target size the modeler team will grow to. Inherent in these assumptions is the expectation that the team is at most fully staffed, and that we do not need to fire a modeler at time 0. Modelers are managed as a SimPy "resource", which allows for useful simulation features like forcing a model to request a modeler and then wait in a queue for said modeler to become available. A "fire staff" event was used to handle the concept of understaffing - both at the start of the simulation and as the result of a modeler quitting during the simulation. When this event triggers, a modeler resource is consumed for the duration until "time to rehire".

The pace of hiring and firing are both modeled as random processes with normal distributions. The inputs are generally based on my team's lived experience with staff turnover and with the time investment necessary to onboard a new employee. For simplicity, the time to hire is quicker than time to fire, which ensures staff levels trend to full capacity relatively quickly.

The model management module creates *n* "Model" objects, each of which manages that model's evolution through its life cycle. When a modeler is available, a Model begins its "build" phase. Once that phase ends, the model needs to be monitored each quarter for the next six quarters, at which point it needs to be rebuilt. After the rebuild, it perpetually executes in the monitor / rebuild cycle. At the end of each touchpoint event, the next touchpoint event is created with "start after" date and a "due by" dates, reflecting the fact that model management must happen within a given window of time. If the simulation passes the due date without the task being executed, then a failure counter is incremented.

Time to complete model (re)building and monitoring are set as uniform distributions. As with staffing, this generally reflects the lived experience of the modeling team. The number of models under management is a key input that dictates total workload.

Results and Conclusions

My team was 4 FTE at the start of 2022, and we got approval to grow to 10 FTE. Through working with my modeling team, we have identified 15 models we need to build. These two facts anchored my sensitivity testing.

For each set of inputs, I ran 1000 simulations, capturing three key facts: the time period when all models had been built, average available FTE after all models were built, and the fraction of model monitoring events that failed to be completed on time.

Inputs		**Outputs**		
Max FTE	**# Models**	**Time Complete**	**Avg Capacity (FTE)**	**% Monitor Fails**
6	15	150	0.8	12%
8	15	120	1.2	10%
10	**15**	**100**	**2.1**	**3%**
15	15	80	6.3	0%
10	25	150	1.1	13%
15	25	115	2.7	5%

Given my team's expected volume of effort, the 10 FTE target team size seems appropriate: this results in an acceptably low monitor failure rate, and allows for a sufficient number of staff available to address other, non-modeled analytics that could crop up.