Decision Trees in Automatic Forecasting Algorithms: The Implementation in Forecast Pro

SARAH DARIN

INTRODUCTION

Forecast Pro, a popular off-the-shelf business forecasting software, has long employed an "expert selection" feature to automatically select a forecasting method from the characteristics of the time series at hand. The feature has been evolving for over 30 years and was the driver of the software's considerable success in the M3 and M4 competitions (Makridakis and Hibon, 2000; Darin and Stellwagen, 2019). The expert selection algorithm used for the M4 entry did not include machine learning, but ML has now been incorporated into the current algorithm.

Stand-alone ML did not perform particularly well in the M4 competition but fared quite well in the M5. In particular, *extreme gradient boosted trees* emerged as an important and effective forecasting methodology, and offered convincing evidence that ML models could add solid value in terms of forecast accuracy. Ideally, they should be considered in an expert selection algorithm alongside traditional statistical approaches like ARIMA and exponential smoothing.

Unfortunately, forecasting with ML is complicated: you need to choose a technique (algorithm), prepare your data, generate features, train a model, and use it to generate forecasts, often one period at a time. The ML models used successfully in the M5 and other forecasting competitions were customized and trained by data scientists. Given that Forecast Pro business users are not data scientists, integrating ML into Forecast Pro required both automation of the ML forecasting process and expansion of the expert selection algorithm to determine cases where ML would likely outperform traditional statistical approaches.

While it was not clear at first that this was achievable, after several months of development we succeeded in embedding an *automated extreme gradient boosted tree algorithm* into expert selection. Our own testing revealed that the ML methods outperformed traditional statistical forecasting methods about one-third of the time.

ML IMPLEMENTATION

Forecast Pro now allows users to build both single- and multiple-input ML models based on the time-series history.

Target Data and Feature Generation

Gradient boosted trees use features to forecast a target variable, typically a demand history for a product or service. Entering a demand history in Forecast Pro requires specifying the number of periods per year (365 for daily data, 12 for monthly data, etc.), the periods per cycle (e.g. 7 for daily data, 12 for monthly data, etc.), and the starting date. The program uses this information to generate a base set of features for its single-input ML algorithm.

Automatically generated features can include moving averages, seasonal periods, date period (used to identify level shifts and outliers), and lags of the target variable. A rule-based algorithm then decides which of these features to consider for a given demand series. Exogenous features that are thought to drive the behavior of the target variable, such as prices, income, weather, and demographic factors, can also be included as features in a multiple-input model.

Hyper-Parameters

As the tutorial from Evangelos Spiliotis explains, hyper-parameter specification can have a large impact on forecast accuracy. Too great a maximum tree depth may lead to overfitting, while an overly simplistic tree depth may not capture enough of the data pattern to forecast well.

Selecting the hyper-parameters that maximize forecast accuracy is best achieved through cross-validation, which can be an iterative and expensive process. Estimating a single gradient-boosted model is far more resource intensive than estimating an exponential-smoothing model. Estimating multiple gradient models is obviously even more so. Therefore we designed Forecast Pro's automatic ML to identify hyper-parameters in as few iterations as possible.

Traditional time-series forecasting of a single time series is a unique application of gradient-boosted trees. Sample sizes tend to be considerably smaller than in most other ML applications, making overfitting particularly concerning. XGBoost includes a wide range of hyper-parameters, each with a reasonable default value for larger-scale ML applications.

Forecast Pro's ML algorithm identifies which hyper-parameters, such as maximum tree depth and maximum number of trees, have the greatest impact on forecast accuracy, and uses cross-validation to optimize only these. The remaining hyper-parameters are assigned default values appropriate for two-plus years of daily, weekly, or monthly demand. These default values are the same as those in XGBoost except for the learning rate (eta), which is set to 0.1. We found that using a lower learning rate than XGBoost's default (0.3) generated more accurate forecasts.

Forecast Accuracy Measurement

We use rolling-origin evaluations to measure accuracy for alternative ML models. In essence, for a given time series, we compare a range of appropriate maximum tree depth as well as a pair of objective functions (Tweedie or squared error). We estimate a gradient-boosted model for each tree depth in that range, using our prescribed stopping logic to determine an appropriate number of trees. The stopping logic leverages XGBoost's out-of-sample RMSE evaluation, where possible. If there is not enough data to support out-of-sample evaluation, the automatic algorithm creates trees until within-sample improvements fall below a minimum threshold. After generating a boosted tree model for each tree depth (with the number of trees used identified as described), the algorithm picks the tree depth that has the best out-of-sample performance.

Multiple Input Models

If exogenous variables are specified in the ML model, the Forecast Pro algorithm will automatically determine if they should be retained as features. Because XGBoost only uses the features that help improve accuracy, variable selection is essentially automated. The same validation approach described above is used to identify maximum tree depth and number of trees automatically, although you have the option to specify these hyper-parameters manually.

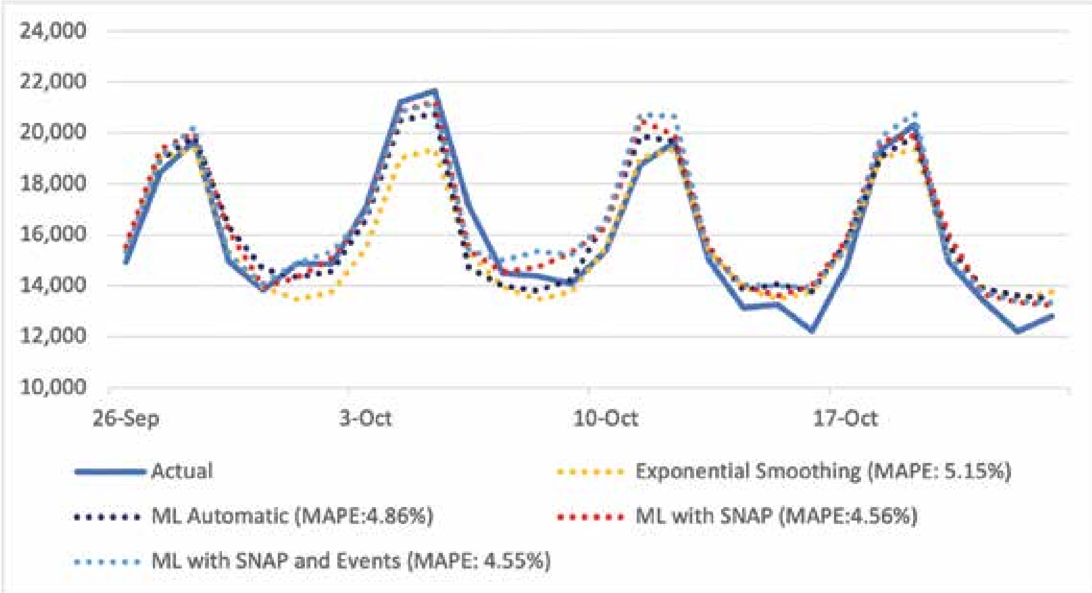
PERFORMANCE FOR THE M5 SERIES

The Spiliotis tutorial used an M5 data series (Daily Unit Sales in California) to demonstrate the relative accuracy of a single decision tree, a random forest, and a gradient-boosted tree model. We have used this same data series and horizon to evaluate forecasting accuracy in expert selection, as shown in **Figure 1**. The exogenous feature considered is information about the USDA's Supplemental Nutrition Assistance Program (SNAP)

Figure 1 compares the out-of-sample forecasts from three XGBoost ML models plus an exponential-smoothing model:

- The *ML Automatic*, based only on the demand history features.
- A multi-input ML model, the *ML with Snap*, which brings in this exogenous variable.
- A multi-input ML model, the *ML with Snap and Events*, the events being those

Figure 1. Forecasting Performance Comparison for Forecast Pro ML Models



provided in the M5 data set denoting holidays, sporting events, and other.

Both multi-input ML models used Forecast Pro's automated parameter selection.

The optimized hyper-parameters were

ML Automatic: tree depth = 4, number of trees = 1322

ML with Snap: tree depth = 4, number of trees = 553

ML with Snap + Events: tree depth = 3, number of trees = 860

Forecast Pro's expert selection chose the ML automatic model shown above. In previous versions of Forecast Pro (which did not include ML), expert selection selected the exponential-smoothing model shown above. The ML automatic model slightly outperforms the exponential-smoothing model, reducing the MAPE by 6%, so adding ML to expert selection does improve accuracy for this series. Including SNAP in the model improves the MAPE by an additional 6%, and a very marginal improvement results with the addition of Events as a feature. These models appear to forecast more accurately than the illustrative models shown in the tutorial, all of which used predefined but reasonable default parameters.

ACCURACY VS. EXPLAINABILITY

We recommend use of gradient-boosted trees if your primary goal is forecast accuracy, but you do not need to explain *why* the forecast is what it is. "Explainability," it is argued, should be considered a key component of the costs of forecasting, and ultimately the cost-benefit of a forecasting system (Yardley and Petropoulos, 2021).

One of the most important differences between statistical time-series forecasting and machine-learning forecasting is that the statistical approaches give you information about the relationships in the data, while machine-learning models often do not. For example, a seasonal exponential-smoothing model produces seasonal indices that portray *how* the seasonality varies throughout the year. While a single regression tree can be explained, multiple regression trees cannot. You can compute the relative importance of your features to your machine-learning forecast, but you cannot succinctly summarize the relationship between features and demand. If you need to explain your forecasts, do not use gradient-boosted trees.

So, should you use machine learning if you only care about the forecast accuracy? You should consider it, but ultimately you should use the methodology that is most accurate for each time series you wish to forecast. Automatic AI-driven algorithms, such as expert selection, can be used to automate this process. For the specific M5 data series analyzed, expert selection determined that the ML model was appropriate, performing better than the alternative statistical methodologies in out-of-sample testing.

If you have a theoretical understanding of the forecasting dynamics, i.e. how the target variable is influenced by its drivers, specifying that known relationship in a regression will likely produce more explainable forecasts. It makes sense to consider ML, however, when you don't have a clear comprehension of the demand-generating process. When you don't understand how the features affect the target variable, it is easier to include those additional drivers with automated gradient-boosted trees than it is with regression.

CONCLUSIONS

While machine learning has emerged as an important new forecasting technique, it is not a replacement for statistical time-series methods. You should use ML when you do not need to explain your forecasts *and* it outperforms traditional statistical methods. Forecast Pro's automatic ML with XGBoost automates feature selection and hyper-parameter specification, and can outperform traditional statistical methods for certain time series. Automation makes ML accessible to business users, but lack of explainability may be a concern.

Sarah Darin is a Principal at Business Forecasting Systems, Inc.

sdarin@forecastpro.com

REFERENCES

Darin, S. & Stellwagen, E. (2020). Forecasting the M4 Competition Weekly Data: Forecast Pro's Winning Approach, *International Journal of Forecasting*, 36(1), 135-141.

Makridakis, S. & Hibon, M. (2000). The M3-Competition: Results, Conclusions and Implications, *International Journal of Forecasting*, 16(4), 451-476.

Yardley, E. & Petropoulos, F. (2021). Beyond Error Measures to the Utility and Cost of the Forecasts, *Foresight*, Issue 63, 36-45.