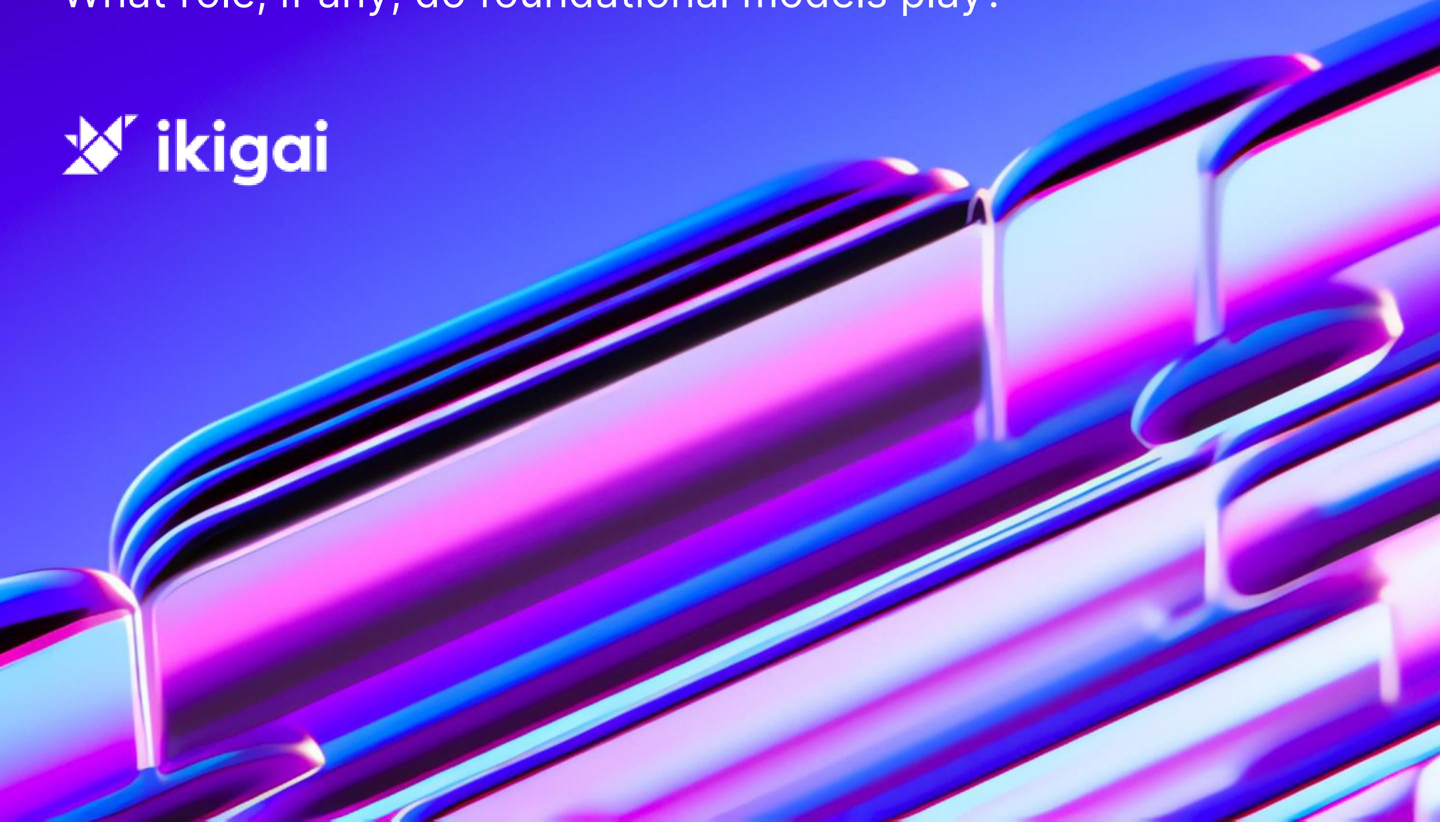
Ikigai AI/ML Team Perspective 3/25/2024

Time Series Forecasting: Why, What, and How

What role, if any, do foundational models play?

ikigai



Contents

Introduction

In this report, we share why Time Series Forecasting is an essential capability for every enterprise, what time series forecasting involves, and how it can be implemented. We compare three implementation approaches for an enterprise workflow: using a foundational model, building an internal solution, or using Ikigai's Generative AI for tabular, time series data platform approach with aiCast^{TM}.

Our evaluation led us to the following conclusions:

- While foundational models are accessible and building internal solutions is possible**, effective time series forecasting requires more functionalities than most foundational models or open-source models provide.**
- In their current or anticipated form, foundational models are far from ready to be useful for accurate, effective time series forecasting.
- Moreover, building an internal solution is talent- and time-intensive.
- **Ikigai's "Embedded AI" approach maximizes performance & accuracy** with more functionality, production-level ready applications, higher data integrity, and cost-efficiency.

Why is Time Series Forecasting essential for enterprises?

Every enterprise workflow involves forecasting and planning with complex data under uncertainty:

- For Product & Sales, how do they **forecast demand** and **plan supply** for Sales and Operations Planning (S&OP)?
- For Talent, how do they **forecast work** and **plan for skills** for workforce planning & optimization?
- For Customer Relationship Management, how do they **forecast customer interest** and **plan for sales resources**?
- For Finance, how do they **forecast cash burn** and **plan for growth** for Financial Planning and Allocation (FP&A)?

In each case, the key challenge is a **balancing act** between two opposing variables: supply vs demand, cash vs spend, skills vs tasks, customer interest vs sales resources, marketing channels vs marketing budget. To achieve an effective balance across each pair, enterprises require forecasting and planning.

Consider supply vs demand: the demand is typically unknown and future looking. Anticipating demand requires accurate forecasting. Supply planning is then dependent on knowledge of demand and other uncertainties like shipping lag time, and vendor availabilities. Planning requires making appropriate decisions subject to various constraints, with the goal of achieving the right revenue, profit, and risk profile.

The data for each of these enterprise workflows is tabular and timestamped. In fact, enterprise data can be thought of as a collection of spreadsheets, each cell containing either a numeric value or a short text and associated with a timestamp. Forecasting on this tabular, timestamped data is essential to addressing the balancing act at the core of all enterprise workflows.

Businesses have traditionally used deterministic models on disparate data to make educated guesses, **but the real world is stochastic/probabilistic**. A natural and impactful **aspiration for enterprises is to leverage AI to effectively capture uncertainty** and achieve these enterprise balancing acts.

What does Time Series Forecasting involve?

Time series forecasting is the process of leveraging tabular datasets, including at least one that is timestamped, to predict future outcomes. More than a single number, forecasts are about capturing overall uncertainty by utilizing all structural properties of it.

A common misconception is that time series forecasting involves only one time series and one functionality – forecasting future values. However, effective time series forecasting involves multiple types of time series data and several functionalities.

Here, we separate time series forecasting up into two broader conceptual components: time series datasets and the forecasting functionalities used on those datasets. **The combination of times series datasets and multiple forecasting functionalities can yield complete, information-rich forecasts and business insights.**

Times Series Data

Outcome Time Series Data: Time series data captures how a value or outcome varies numerically over time. In a business context, there are many values of interest captured through multi-variate time series data. For example, product sales over time, capital expenditures, and workforce requirements are all sets of values that change over time.

Auxiliary Data: Time series data are frequently associated with and impacted by other various auxiliary multi-variate time series data. For example, product sales are affected by the cost of raw materials, capital expenditures are affected by the market credit rate, and workforce requirements are affected by the labor market job reports.

Intervention Data: Time series data are also impacted by intervention multi-variate time series data. For example, promotions affect sales, price of goods sold affects capital expenditures, and compensation packages affect workforce requirements.

Meta Information: In addition, there is useful multi-dimensional non-temporal information about these time series data. For example, product meta information and hierarchy can help explain product sales, types of revenue and expenditure can help explain capital expenditures, and people's skills can help explain workforce requirements.

Domain Properties: Finally, there are domain properties or relationships that impose additional structure captured through multi-dimensional data. For example, geography and transportation networks can constrain product sales, payee and collection information can constrain capital expenditures, and organizational structure can constrain workforce requirements.

Forecasting Functionalities

Imputation: Data must be prepared before forecasting can occur. Data preparation includes accounting for, or “imputing” missing values by probabilistically generating what the missing values could be. For example, in sales forecasting, missed sales values could be imputed based on inventory data. Imputation is both necessary as a preparatory step towards creating a better and helpful to understand gaps in historical records.

Anomalies: Another important data preparation step is identifying anomalies in the data. Anomaly identification can reduce or remove errors before forecasting.

Change Points: The *dynamics* or *model* of time series data are likely to change due to various external events. For example, the online purchases before and after the Covid-19 pandemic have been very different: for example, strategic changes led to different spending patterns in business, organizational restructuring led to different resource requirements. The ability to identify the points in time where the dynamics change, or “change points,” can help determine which segments of data should be used for model learning to enable better forecasts.

Cohorts: Within a collection of multiple time series, some may behave similarly. These similarly-behaving time series form *cohorts*. Identifying cohorts has massive implications for decision making. For example, cohort identification can indicate which products can replace each other or sell together; which financial instruments are replaceable; which individuals follow similar career paths; and more. Restricting co-learning to cohorts can also improve forecasts performance.

Hierarchical Reconciliation: Single time series have a natural structure within them: any value aggregated over each day of a month needs to be consistent with value at the monthly time scale. More generally, a collection time series may observe additional relationship amongst themselves due to structural hierarchy, e.g., product categorization, spending categorization, or organizational structure.

Patterns Of Time Series: The natural way to capture the patterns within time series data is through *trend*, *seasonality* (aka periodic behavior) and *effect of recency*. In addition, the ability to understand *similarity* across time series and across time ranges is useful to *explain* forecast behavior.

Forecasting: Forecasting projects the outcome(s) of interest over a given time range in a stochastic, non-deterministic way. The output is not only a number or probabilistic time series, but also a *well-calibrated* confidence interval with accuracy statistics, e.g., with 95% chance, the 90% confidence interval for the value week ahead is within range 134.27 and 151.64.

Interpretation: A forecast is only effective when it can be interpreted, understood, trusted, and used by those who consume it.

Classification: Classification of a time series is necessary to apply an associated attribute or label. For example, classification could be used to label top-selling products, well-performing employees, or faulty manufacturing equipment based on sensor data.

Expert Feedback: An expert should be able to provide human feedback to improve forecast results. This includes providing feedback on change point, anomalies, imputation, forecasting, calibration and more.

Synthesis: Synthesis is the ability to use all inputs, e.g., data and expert feedback, to create the best holistic forecast of outcomes of interest.

How can enterprises implement Time Series Forecasting?

Before diving into tactical approaches, we consider **four pillars of a successful AI implementation:** Embedded AI, experts in the loop, education, and enterprise readiness.

Embedded AI is critical to retain and enhance existing functionalities for workflows. Enterprise workflows, as discussed earlier, are run through a collection of software applications. These are feature- and functionality-rich applications that have evolved to a mature stage over the past few decades. Increasingly, they are becoming service- and cloud-based. To bring AI to such workflows, it is essential that enterprises embed it within existing workflows rather than creating new ones. The **natural way to achieve embedding is through API-based integration.** This type of embedding is critical to rapid adoption because it naturally retains existing functionalities while making them AI-powered.

Experts in the loop enables continually improved AI performance and business insight. Enterprise workflows involve a collection of experts who make decisions daily, weekly, monthly, and quarterly. These experts use the available information, their domain expertise, their enterprise priorities, and their instincts to make the best possible decisions. The **traditional view of AI has been that it provides predictions on data without requiring human input,** but **experts have important roles to play**. Exceptions are common when using AI, and experts can correct for it. Experts can use AI to receive recommendations and make the best decision with additional information that they have. This, in turn, **enables the AI to continually learn from the expert-enriched data. We need AI, and in particular, time series forecasting, that has expert in the loop to produce better results.**

Education drives AI adoption. Working with AI is different from working with traditional deterministic software applications. For example, consider the following AI result: for this value of interest, the 90% confidence interval is the range between 134.27 and 151.64 with model accuracy of 95%. **Understanding and correct interpretation of AI results is required to avoid making wrong decisions with consequential business impacts**. It is essential to educate the entire enterprise workforce to understand how to work with AI, not just teach a small fraction of the workforce how to build AI solutions or how to architect AI solutions.

Enterprise readiness is necessary for lasting impact from AI. **Any AI functionality**, including and especially time series forecasting**, needs to continually update as new data is received, expert input is incorporated, and conditions change**. This means that continuous model updates and redeployment of embedded AI is necessary. From an end user perspective, it is essential to continually provide model performance updates. To be useful and build trust, **explainability and interpretation around model outcomes are essential.** Finally, the data and model lineage are important to maintain auditability and attribution.

There are **three tactical approaches** to implementing time series forecasting in enterprise workflows. In the next section, **we compare each approach across five criteria:** performance and accuracy, functionality, production readiness, cost, and data integrity.

Approach 1. Foundational Model

The foundational models like Large Language Models (e.g., GPT, Gemini, LLaMa, Falcon) or emerging time series specific models (e.g., Google's recently announced model) provide forecasting APIs out of the box. To obtain good, accurate forecasts with such models, context data needs to be provided. Specifically, given context data and zero-shot learning, the forecast or prediction query should be an API call to a foundational model. This requires using a hosted instance (e.g., GPT from Open AI) or installing and maintaining an open-source model (e.g., Falcon or LLaMa).

Approach 2. Internally-built model

An alternative to using a foundational model with zero-shot learning and prediction (i.e., using a Foundational Model with no previous classification examples given to the model) is to build an in-house model. This can be achieved in two main ways:

- Starting with a foundational model, it can be *fine-tuned* using data and the use-case specific requirements to enable better forecasting. This requires hosting, fine-tuning and maintaining an available foundational model and may not be within the means of most organizations given the required resources and highly-skilled AI teams.
- Available open-source models for time series (e.g., Prophet from Meta) can be trained on enterprise data. Such a process is simplified due to the availability of co-pilots enabled by various solutions built on LLMs.

Approach 3. Ikigai's Time Series Forecasting with aiCast™

Ikigai provides its time series functionalities based on its GenAI for tabular, time stamped data out of the box within the platform. It also provides the ability to select best performing models across an ever-growing collection of time series models in an automated manner. This includes Ikigai's own model, and open source models, among others. Because Ikigai's Time Series Forecasting leverages LGMs, the approach is domain agnostic and be applied to any use case or benchmark dataset. It can be embedded into any application via Ikigai's robust APIs seamlessly.

Comparison: Performance & Accuracy

The key performance metric in forecasting is the accuracy of the prediction compared to the real outcome, which can be reported in terms of Mean Absolute Error (MAE) or other metrics. We include comparisons between Ikigai's aiCast^{TM}, open-source models, aiLLM (PISA) using hosted foundational model, and Google's recently released TimesFM, a foundational model for time series.

Performance is heavily influenced by auxiliary data, multiple time series together, and meta information of attributes. However, not all models have these additional functionalities. In the spirit of apple-to-apple comparison, we discuss the accuracy of the three approaches using benchmark data with base functionalities only of aiCast^{TM}. In other words, Ikigai's approach uses aiCast^{TM} restricted to only functionalities that other models have (e.g., no auxiliary data, meta information incorporation).

Comparison of three approaches across benchmark datasets. Model performance reported in form of R^2 (best = 1). Best performance for each dataset is **highlighted,** with percent improvement over next best model result in parentheses.

	Approach 1. Foundational Model	Approach 2. Internally-built model			Approach 3. Ikigai's Time Series Forecasting
Benchmark dataset	aiLLM (PISA)	LSTM	Prophet	DeepAR	aiCast^{TM}
SF Traffic Occupancy rate of traffic lanes in San Francisco, sampled every 15 minutes	0.3	0.761	0.462	0.771	**0.776** (2%)
Electricity 15-minute electricity loads for 370 households	-10.5	-.1261	0.197	0.764	**0.829** (9%)
Exchange Daily exchange rates of eight foreign currencies between 1990-2016	-14.48	-1.825	<-10	0.579	**0.756** (31%)
Synthetic Fundamental time series synthetically generated with noise autocorrelated that can be used to improve predictions	-3.05	0.381	-0.445	0.323	**0.476** (25%)

Comparison of Approaches #1 (Google's TimeFM) and #3 (Ikigai's aiCast^{TM}) across benchmark datasets.[1] Model performance reported in terms of Mean Absolute Error. Best performance for each dataset **highlighted,** with percent improvement over next best model result in parentheses.

	Approach 1. Foundational Model	Approach 3. Ikigai's Time Series Forecasting
Benchmark Dataset	Google's TimesFM	aiCast^{TM}
AUS Electricity Demand Half-hourly electricity demand of 5 Australian states	523.12	**224.08** (57%)
Bitcoin 18 daily time series of potential influencers of bitcoin price, including hash rate, block size, etc.	1.97 x 10^{18}	**6.94 x 10^{17}** (65%)
Pedestrian Hourly pedestrian counts captured by 66 sensors in Melbourne city starting in 2009	58.35	**44.78** (23%)
Weather 3010 daily time series representing for rain, min. and max. temperature, and solar radiation across Australian weather stations	2.17	**2.02** (7%)
Tourism (monthly) 366 monthly time series used in Kaggle Tourism forecasting competition	2916.02	**1871.69** (36%)
COVID-19 Deaths 266 daily time series representing COVID-19 pandemic-related deaths over 8 months	555.24	**124.1** (78%)
FRED-MD 107 log-transformed monthly time series showing macro-economic indicators from the Federal Reserve Bank	**1823.16** (30%)	2621.15
NN5 111 time series representing daily cash withdrawals from ATMs in the UK	4.09	**3.92** (4%)
SF Traffic Hourly Occupancy rate of traffic lanes in San Francisco, sampled every 15 minutes	**0.01** (0%)	**0.01** (0%)
Saugeen River Flow Single, long time series of daily mean flow of the Saugeen River in cubic meters per second over 64 years	24.53	**21.64** (12%)
US Births Single, long time series of number of births in US over 20 years	408.49	**382.47** (6%)
Hospital 767 monthly time series representing patient counts related to medical products over 6 years	19.34	**18.25** (6%)
Solar Weekly 137 time series representing weekly solar power production in Alabama in 2006	1184.01	**721.59** (39%)
CIF 2016 72 monthly time series from the banking domain	**896414.41** (35%)	1371901.38

[1] We have compared aiCast^{TM} against Google's foundational model separately because Google has only reported benchmark scores as MAE instead of R^2. There is no public access available to Google TimeFM.

Performance & Accuracy Comparison Results:

Approach 1. Foundational Model

The hosted foundational model performs the most poorly across the other approaches in Table 1. Google's TimesFM foundational model has higher accuracy than Ikigai's approach for only 2 out of 14 benchmark datasets.

Approach 2. Internally-built model

The open-source models perform worse than Ikigai's aiCast™ for every benchmark dataset.

Approach 3. Ikigai's Time Series Forecasting with aiCast™

Ikigai's approach is superior to the other approaches even when its capabilities are restricted to enable apple-to-apple comparison. In the context of enterprise workflows, Ikigai's additional time series functionalities would likely improve the accuracy even further beyond what would be out-of-the-box due to its additional functionalities: expert feedback, anomaly detection and removal, imputation, change points, auxiliary data, hierarchical reconciliation and intervention data incorporation. **Ikigai's aiCast™ performs better than Google's TimesFM foundational model in 11 out of 14 datasets** in Table 2, in some cases with error an order of magnitude more. **When Ikigai's aiCast™ outperformed Google's TimesFM, it did so on average by 30%.**

Comparison: Functionality

Comparison of approaches for functionalities enabled out-of-the-box

	Approach 1. Foundational Models	Approach 2. Internally-built model	Approach 3. Ikigai's Time Series Forecasting
Traditional forecasting functionality	Yes	Yes	Yes
Anomaly, changepoint, and cohort detection and classification out-of-the-box	No	No	Yes
Hierarchical reconciliation	No	No	Yes
Pattern decomposition, explainability, and synthesis	No	No	Yes
Expert feedback incorporation	Requires building	Requires building	Yes

Functionality Comparison Results:

Approach 1. Foundational Models

The foundational model primarily enables the forecasting capability. It can provide expert in the loop capability by incorporating it explicitly in the data.

Approach 2. Internally-built model

Internally building a model requires building each of the forecasting capabilities using open source or innovation. While some of the functionalities (e.g., change points) can be enabled using different open-source solutions, many functionalities require additional engineering, such as cohorts, anomalies, expert in the loop, explainability, interpretation and more.

Approach 3. Ikigai's Time Series Forecasting with aiCast™

Ikigai provides all the functionalities above and described earlier in the report out-of-the-box, at scale and in the production environment.

Comparison: Production Readiness

Comparison of approaches for production-readiness

	Approach 1. Foundational Models	Approach 2. Internally-built model	Approach 3. Ikigai's Time Series Forecasting
Simple model management and deployment	**Yes**	No	**Yes**
Simple instrumentation and maintenance, performance tracking, auditability	No	Requires building	**Yes**
Ease of embedding in AI workflows (e.g., via APIs)	Requires building	Requires building	**Yes**

Production Readiness Comparison Results:

Approach 1. Foundational Models

As discussed earlier, the Foundational Model simplifies the task of prediction, model management and deployment. However, it still requires instrumentation and maintenance for bringing the right data continually, enabling expert in the loop and performance tracking. Auditability remains challenging, if not infeasible. Finally, embedding AI into the workflow requires additional instrumentation.

Approach 2. Internally-built model

As discussed earlier, the in-house model approach requires serious instrumentation around entire production workflow ranging from DevOps, MLOps, Expert in the Loop Ops, and embedding AI into the existing workflow. In effect, it requires building the entire infrastructure in addition to all the time series forecasting functionalities.

Approach 3. Ikigai's Time Series Forecasting with aiCast^{TM}

Given the platform architecture and time series functionality capabilities, **Ikigai provides ability to get up and running into *any* data environment seamlessly.** It comes with all the instrumentation required and thus simplifies the task of embedding AI into existing workflow drastically.

Comparison: Cost

Cost Comparison Results:

Approach 1. Foundational Models

The zero-shot learning of hosted foundational models can be extremely expensive. For example, as per recent pricing in March 2024, with just 1GB of data a hosted GPT model can cost thousands of dollars for a single prediction query[1]. Indeed, hosting and training your own foundational model can require a lot more resources – both compute and talent.

Approach 2. Internally-built model

The in-house model approach requires a very talented team with a range of skills, e.g., experienced data engineers, experienced AI/ML scientists, business analysts, infrastructure engineers, front-end engineers, and product manager at minimum. Implementation is time consuming and requires resource investments for compute power.

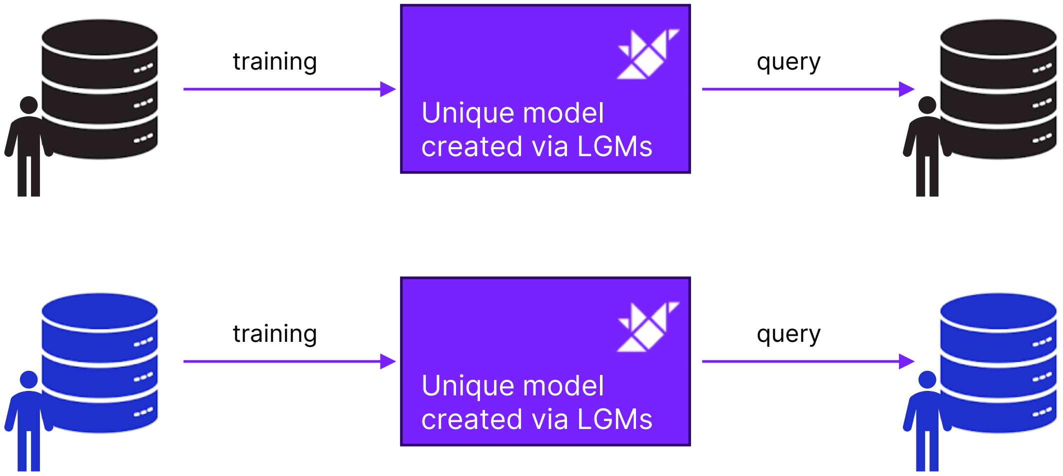
Approach 3. Ikigai's Time Series Forecasting with aiCast™

Ikigai is built for scale with very efficient technology. **Ikigai's proprietary technology is built using CPUs (not GPUs), which allows its publicly hosted SaaS offering to be extremely cost effective.** It scales to allow for TB size data.

[1] Assuming ~$0.00006/token, ~$0.000004/byte (1 token = 4 char = 16 bytes), and therefore ~$4,000 for a query on 1GB.

Comparison: Data Integrity

Ikigai's approach creates unique models for every use case using Large Graphical Models (LGMs).



Data Integrity Comparison Results:

Approach 1. Foundational Models

Publicly hosted foundational models naturally integrate the end users' data every time a query is made. This results in data leakage.

Approach 2. Internally-built model

A **privately hosted in-house model retains data integrity** and avoids data leakage.

Approach 3. Ikigai's Time Series Forecasting with aiCast™

Ikigai's LGM technology creates models that are unique the data provided to it; it is neither trained on nor applicable to others' data. **This explicitly retains the data integrity and prevents any form of data leakage.**

Conclusion

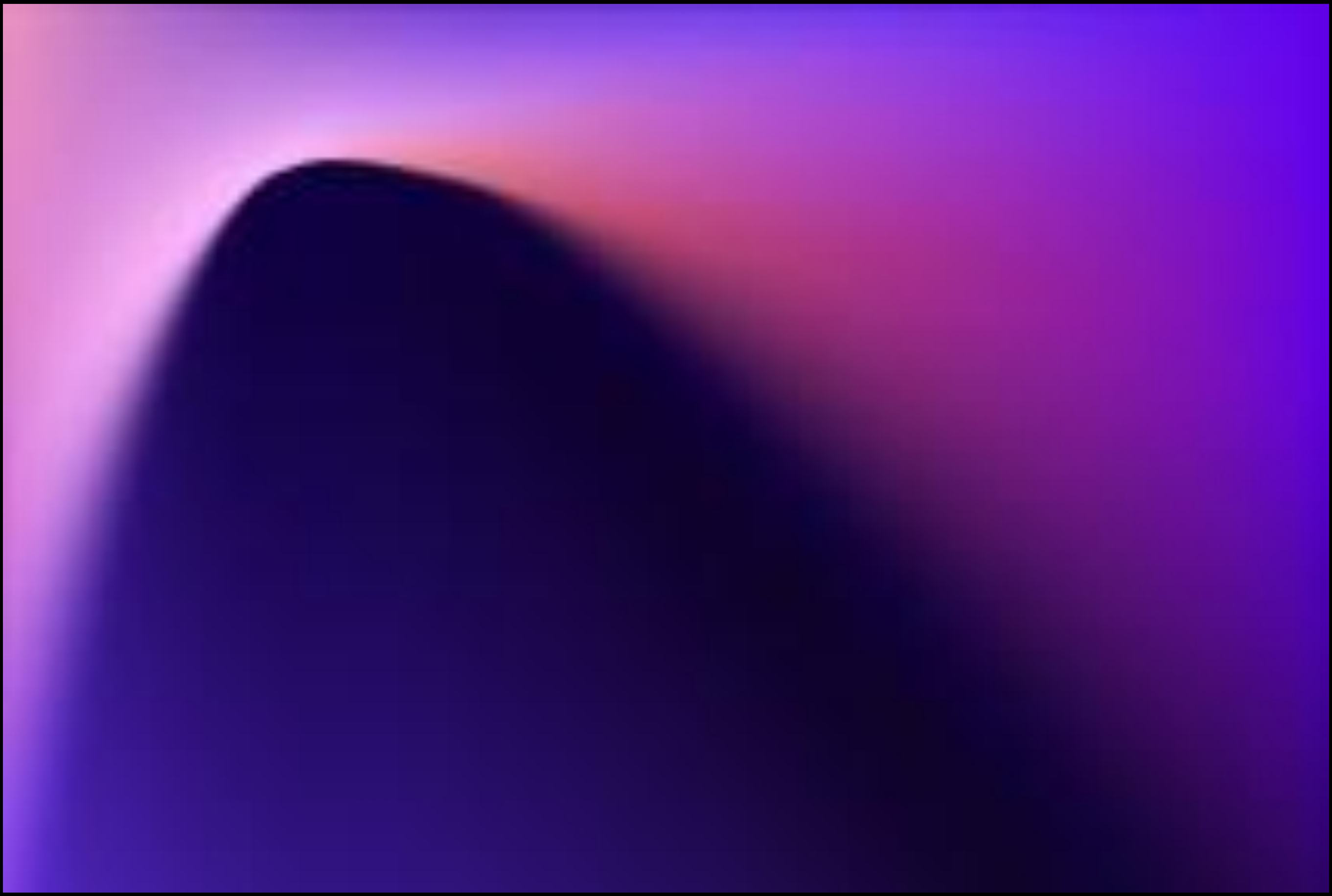
For Time Series Forecasting, **Ikigai's platform** using Large Graphical Models and aiCast^{TM} **surpasses foundational models and internally-built model approaches in terms of performance and accuracy, functionality, production-readiness, cost, and data integrity.**

While the Foundational Models approach simplifies the requirements around model building, model management, MLOps and more, **embedding Foundational Models into an enterprise workflow requires instrumentation to continually function** (e.g., to incorporate the right data from existing enterprise environments for zero-shot learning, to bring back the predictions, to enable expert in the loop, explainability and interpretation, attribution and more). **In addition, foundational models have risks for data security and leakage.**

There are issues with building in-house models as well. **Internally-built models require taking the model from proof-of-concept stage to full production stage, which in turn requires infrastructure and skills around capabilities** like data engineering, compute for ML, cloud DevOps, MLOps, Expert in the Loop, explainability and interpretation, attributions and more. Another issue is that most organizations are only able to build a solution for one use case, at best. Enterprises will have multiple use cases – and building, fine-tuning, evolving, and maintaining multiple models becomes hard to scale if they will be internally-developed every time.

Ikigai's time series forecasting functionalities include all the functionalities discussed in the earlier report to provide richer business insights. These functionalities can be enabled on disparate data without requiring intensive preparation.

Additionally, the Ikigai platform has hundreds of data connectors available out-of-the-box that simplify the process of applying Ikigai's AI to enterprises' data, even when the data sources live in multiple environments. Going from data to time series forecasting capabilities is seamless. The **outcomes of these functionalities can be embedded in existing workflows through APIs.** The continuous model updates, model management and redeployment, expert in the loop, MLOps and observability, explainability and interpretation, and attribution is available out of the box. The platform scales to terabyte-size data seamlessly, and can also enable schedules, alerts, triggers, and model performance review and dashboarding. **Ikigai's embedded AI platform is fully-featured, with a robust set of functionalities around time series forecasting, powered by its GenAI for tabular, time stamped data.**



About the authors:

Abdullah Alomar, Arth Dharaskar, Siddhant Dube, Eliza Knapp, Nate Lanier, Parvathi Narayan, and Devavrat Shah

Visit our website to learn more:

https://www.ikigailabs.io/