Ikigai

Enterprise AI Apps Platform: Easy, Fast, Powerful

By 2030, there will be 1.5B users of AI Apps within enterprises[1]. To achieve AI's potential in the enterprise, such apps need to enable an integration of experts with AI. Ikigai's AI Apps platform make it easy to build, deploy and manage an Expert-in-The-Loop (EiTL) AI app towards this goal & more.

Introduction

What is an AI App? Imagine **you** are ***planning inventory*** at billion $ retail org; ***deciding daily labor*** *supply* at distribution centers across the country; ***auditing insurance reimbursements*** at an insurance back office; ***complying transactions*** *in financial markets*; ***reconciling disparate data*** *sources* aka *entity resolution;* or ***making real-time Know-Your-Customer (KYC)*** decisions to balance fraud and revenue in a fintech – what you would like is an **AI enabled enterprise-wide App** that, not only allows you to track various in real-time and provides predictive capabilities, but also analyzes hundreds of thousands of scenarios for you, gathers input from you and your team to **recommend decisions**, and enables you to take **actions** - while the underlying **system learns from your actions** for future.

Building such AI apps are far more complex than building traditional software apps and software apps themselves are not easy to build, even now.

Ikigai's Enterprise AI App platform enables building enterprise AI apps, end-to-end and that learn with expert-in-the-loop feedback at scale, in hours without requiring a team of engineers, scientists, special purpose cloud, and engineering resources. Currently, Ikigai's AI Apps are utilized in supply chain (labor planning, sales and operations planning), retail (demand forecasting, new product launch), insurance (auditing, rate-making), financial services (compliance, KYC), banking (customer entity matching, txn reconciliation), manufacturing (predictive maintenance, quality assurance); and the list is growing.

[1] See Points 5 and 6 in IDC FutureScapes.

Confidential © Ikigai Labs, 2023.

Ikigai's platform is powered through its innovative technology out of MIT on computing with the Large Graphical Model (LGM). Enterprise data is messy, lives across disparate systems, and is largely tabular. LGMs are precisely the type of model that works well in this environment. LGMs are effectively Neural Networks on Steroids; and Ikigai's tech can be viewed as what **GPUs did for Neural networks,** but it is **not resource or data hungry.** It uniquely enables such a platform through three proprietary models: ***DeepMatch*** for AI-powered data prep, ***DeepCast*** for predictive modeling with sparse data, and ***DeepPlan*** for Reinforcement learning for scenario analysis. Ikigai's tech enables unique product features such as ***Expert-in-The-Loop (EiTL),*** where the system can gather inputs from experts in real-time and continually learn to get the best out of AI and Experts.

At Ikigai, we believe that this is just a start. There is an opportunity to grow through both technology and product innovation. The powerful LGM will enable further models, including **real-time recommendation systems,** and more. The intuitive interface of Ikigai and its tech combined with innovation in the modern ecosystem will continue to make it easier to use and build with.

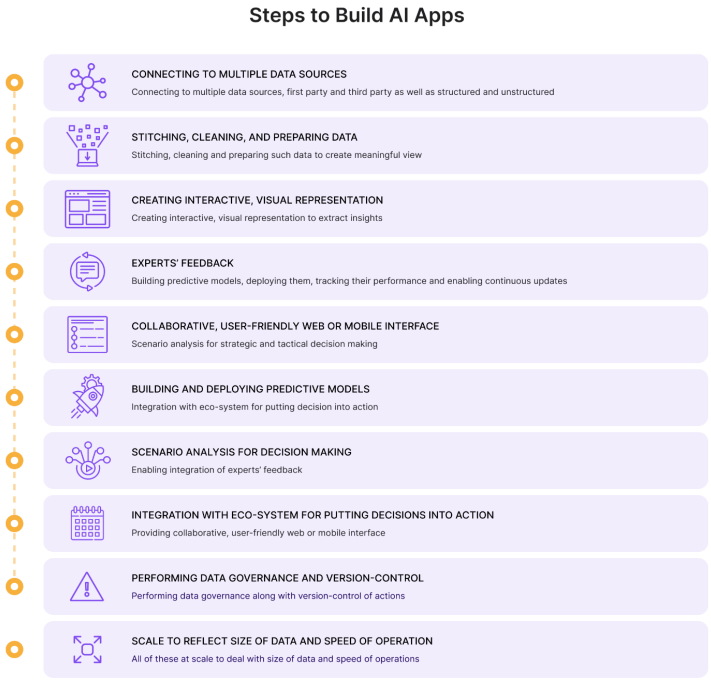
Current Status Quo: Building AI Apps

Building an AI App is very challenging. Currently, such apps are built in one of the two primary ways — **Do-It-Yourself (DIY)** with an internal team of data, ML, infrastructure, and application engineers or use an existing **Point Solution.** In either case, building it requires (i) connecting to multiple data sources, first party and third party as well as structured and unstructured, (ii) stitching, cleaning and preparing such data to create a meaningful view, (iii) creating interactive, visual representation to extract insights, (iv) building predictive models, deploying them, tracking their performance and enabling continuous updates, (v) scenario analysis for strategic and tactical decision making, (vi) integration with existing ecosystems for putting decisions into action, (vii) enabling integration of experts' feedback, (viii) providing collaborative, user-friendly web or

Confidential © Ikigai Labs, 2023.

mobile interface, (ix) performing data governance along with version-control of actions, and, (x) all of these at scale to deal with size of data and speed of operations.

To deploy such an App at scale, a zoo of software infrastructure tools (Kubernetes, Dockerized Service Mesh, Load Balancing, and more) need to be orchestrated in the modern cloud infrastructure (AWS, Azure, GCP). In addition, to enable AI functionalities, MLOps infrastructure along with the ability to incorporate human feedback is required. Typically, this requires utilizing various AI/ML software tools in addition to performing custom application development. To build and maintain such an App, a team of software engineers, data engineers, product managers, data scientists and business analysts are required. Typically, it could take up to a year to build one such App. And, when the need for any modifications for the app or the next app comes around, the same team needs to be put into motion again.



Indeed, this has led to development of Point solutions as fully fledged businesses for some of these "standard" Apps. Though they are available instantly, typically these Apps require custom integration work that can last months and need continual support. Finally, they are brittle; they do not lend themselves to customization as typically needed in most settings.

Challenges. In summary, the key challenges with the status quo are (a) infrastructure for building AI Apps is fundamentally lacking, (b) cloud infrastructure and vendors provide a feasible solution that is expensive, time consuming and requires team of experts, and (c) point solutions are brittle, take time to integrate and are expensive. **As a consequence, despite AI being ready for prime time, we are unable to integrate it in organization. The key to unlocking AI's potential is to go from the traditional *model***

Confidential © Ikigai Labs, 2023.

view (suitable to engineers and scientists) of AI to its *App* view (suitable to everyone).

Enter Ikigai

Ikigai has built AI infrastructure with an easy-to-use product that allows for building enterprise AI Apps at scale using a combination of drag-n-drop interface and spreadsheet-like grid. In effect, it provides an ability to

1. ***One-Click-Connect***. Connect (in and out) to hundreds of data sources at scale within clicks.
2. ***DeepMatch.*** AI enables data prep, stitching and cleaning.
3. ***Intuitive, collaborative Viz.*** Intuitive interface for interactive visualization at scale.
4. ***Unstructured data.*** Extracting information from unstructured data (PDFs, etc.).
5. ***DeepCast.*** Predictive model selection, building, deployment & management within clicks. This makes MLOps and ML observability a one-click functionality.
6. ***Expert-in-the-loop (EiTL).*** Experts-in-the-loop model improvement and incremental model updates.
7. **Scenario analysis** through simulations using historical data.



8. ***DeepPlan***. Decision recommendations using reinforcement learning.
9. ***App.*** Building user-friendly web app interface for interactions and collaboration via drag-n-drop.
10. ***Data, Versioning, Control.*** Data governance, access management and version control.

Confidential © Ikigai Labs, 2023.

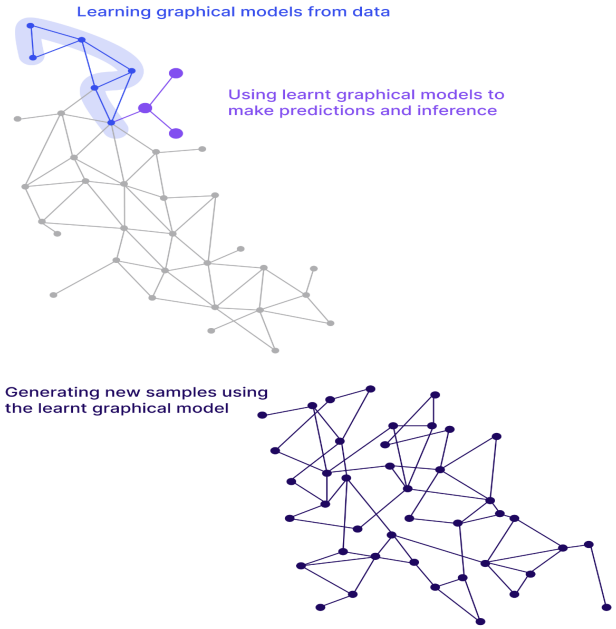
11. ***Process Automation.*** Automation, alerts, scheduling and integration with the ecosystem.
12. ***Resource Auto-Scale.*** Auto-scaling of resources for supporting App.
13. ***Sandbox to Production, Instantly.*** Instant production-grade deployment as soon as it is built!

In addition to the unique AI infrastructure and architecture, the Ikigai platform's distinctive capabilities of DeepMatch, DeepCast, EiTL and DeepPlan are fundamentally enabled by its core technology of Large Graphical Model – for example, it can power

scientific computation at >13x speed on a laptop compared to Spark on 68 m/cs!

Core Technology: Computing with Large Graphical Models (LGM)

What Is a (Large) Graphical Model. The generative view of data posits that all observations are *samples* of an underlying *probability* distribution – be it text generated by ChatGPT, image generated by Dall-e or simply a number generated by a random number generator. A simple generative model for tabular data is a (*giant*) spreadsheet with columns of it being (random) variables and rows of it being samples of the (probability) distribution over these (random) variables. A *computationally* efficient way to represent such a distribution is through a (*large*) graphical model. In effect, the graphical model is simply the "fuzzy" or "probabilistic" view of the "data cube" – the traditional view of tabular data driving all modern data infrastructure.



In terms of representation power, graphical models are at least as powerful, if not more, compared to the traditional neural networks – in effect, neural networks are *deterministic* versions of graphical models. That

Confidential © Ikigai Labs, 2023.

is, **graphical models are *neural networks on steroids*.** Therefore, if graphical models can be utilized well, they hold the potential to have impact way beyond neural networks.

Historically, graphical models have played an important role in enabling earlier language models, image processing, audio processing as well as empowering Global Positioning Systems (GPS) and the communication codes utilized in modern communication systems. Unlike neural networks which have been empowered by GPUs, graphical models have fallen out of fashion due to how expensive they are to compute.

In terms of comparing with recently popularized and remarkably successful *Large Language Models (LLM)s*, it is worth noting that a text document is fundamentally "linear" (in English, from left to right) with multi-resolution structure (relationship across different parts of the document) captured by the very high-dimensional encoding enabled through highly parameterized model structure. Indeed, the earliest form of the graphical model capturing language or speech were simple "linear" Hidden Markov Models (HMMs), which can be viewed as natural "cousins" of Recurrent Neural Networks (RNNs). Just the way RNNs have evolved (through multiple layers) to enable modern language models, graphical models can evolve to such models as well. However, all of them still have the goal of capturing the "linear" structure.

The Large Graphical Models have the ability to go beyond this "linear" structure and can capture relationships across multiple dimensions. The tabular data fundamentally requires such a model and for that reason, LGMs are much more suitable for them. In the world of neural networks, Graph Neural Networks (GNNs) come close to it.

Computational Challenge with Large Graphical Model. Towards operationalizing large graphical models beyond domain specific applications like GPS, it needs computationally efficient algorithms for: (a) learning graphical models from data, (b) using learnt graphical models to make predictions and inference, and (c) generating new samples using the learnt graphical model.

The first decade of this century saw tremendous progress towards task (b) in terms of developing computationally efficient algorithms. As a byproduct, it also provided an ability to generate samples from a distribution efficiently and, thus, solving (c). The

Confidential © Ikigai Labs, 2023.

second decade has seen tremendous progress towards addressing (a). All of these algorithms naturally lend themselves to what is called "message-passing'' architecture[2].

However, this has not been enough. The above progress corresponds to a "back propagation" style algorithm for training a neural network. But something like GPU for neural networks has been missing – computational architecture for efficient, scalable implementation of "message-passing'' algorithms is required to empower large graphical model learning.



Computing for LGMs with PubSub. At Ikigai, we have achieved this breakthrough by developing novel computational architecture using PubSub-like architecture. As mentioned earlier, data as a graph naturally leads to a "message-passing" computational architecture where computationally necessary information is exchanged between units of data (or nodes of a graph). A computationally scalable way to implement such an architecture is by utilizing the classical PubSub infrastructure – what is utilized by modern data bus e.g. Apache Kafka.

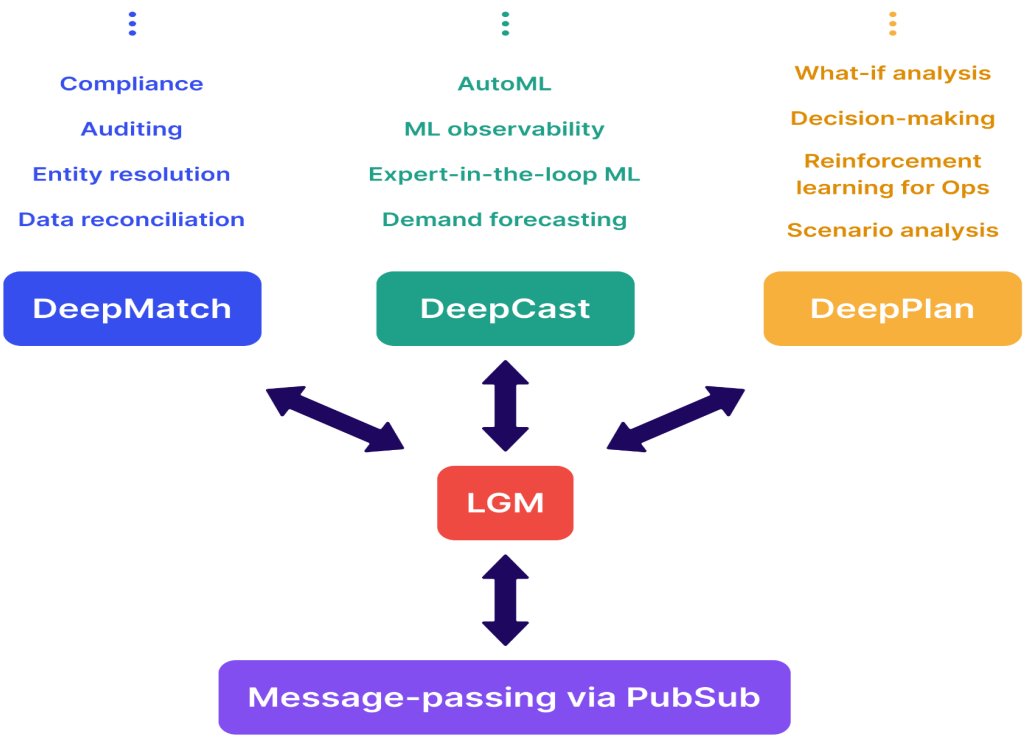
In an MIT patent [3], we outline the architecture to explain how it enables computation to scale with data. **In effect, it brings *computation to data* rather than the traditional approach of bringing *data to compute*.**

Implications of Computing LGM. As detailed in the thesis [4] of Vinayak Ramesh, the ability to compute with *LGM enables >13x faster scientific computation on a commodity laptop compared to modern infrastructure using 68 parallel machines! There are other*

[2] A good reference to learn about this is a course taught by Shah at MIT on Graphical Models [1][2].

Confidential © Ikigai Labs, 2023.

case studies described in the thesis [4] which exemplify the power of computing LGMs by comparing with state-of-art for a hard optimization as well as scalable linear algebra.



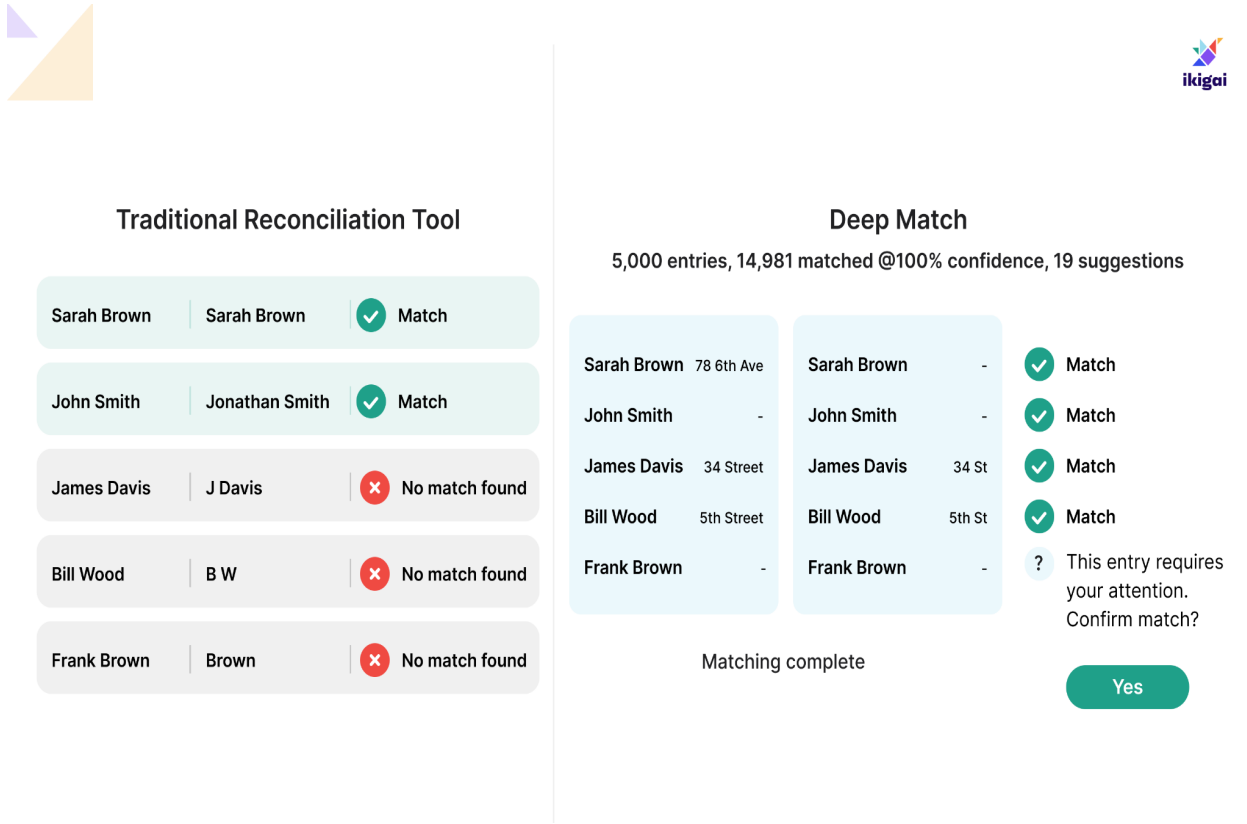
Within the Ikigai platform, LGM enables three models of Ikigai: (i) DeepMatch, (ii) DeepCast, and (iii) DeepPlan. These models make Ikigai's AI apps platform natively usable across use cases and verticals.

DeepMatch. *Stitch Data Sources Together*. The quintessential pre-task of most data-driven analysis is "stitching" multiple data sources together. Traditionally, in the database language, this is achieved through "joins." In many modern settings, however, this does not work as they may lack a shared column or have mismatched entries, thus, the correct relationship is likely to be missed. In a traditional setting, this results in either a lot of manual work or careful, case-by-case, data processing work.

By viewing data as a large graphical model, Ikigai provides *DeepMatch,* which attempts to learn the relationship between columns of a dataset and then uses it to further match

Confidential © Ikigai Labs, 2023.

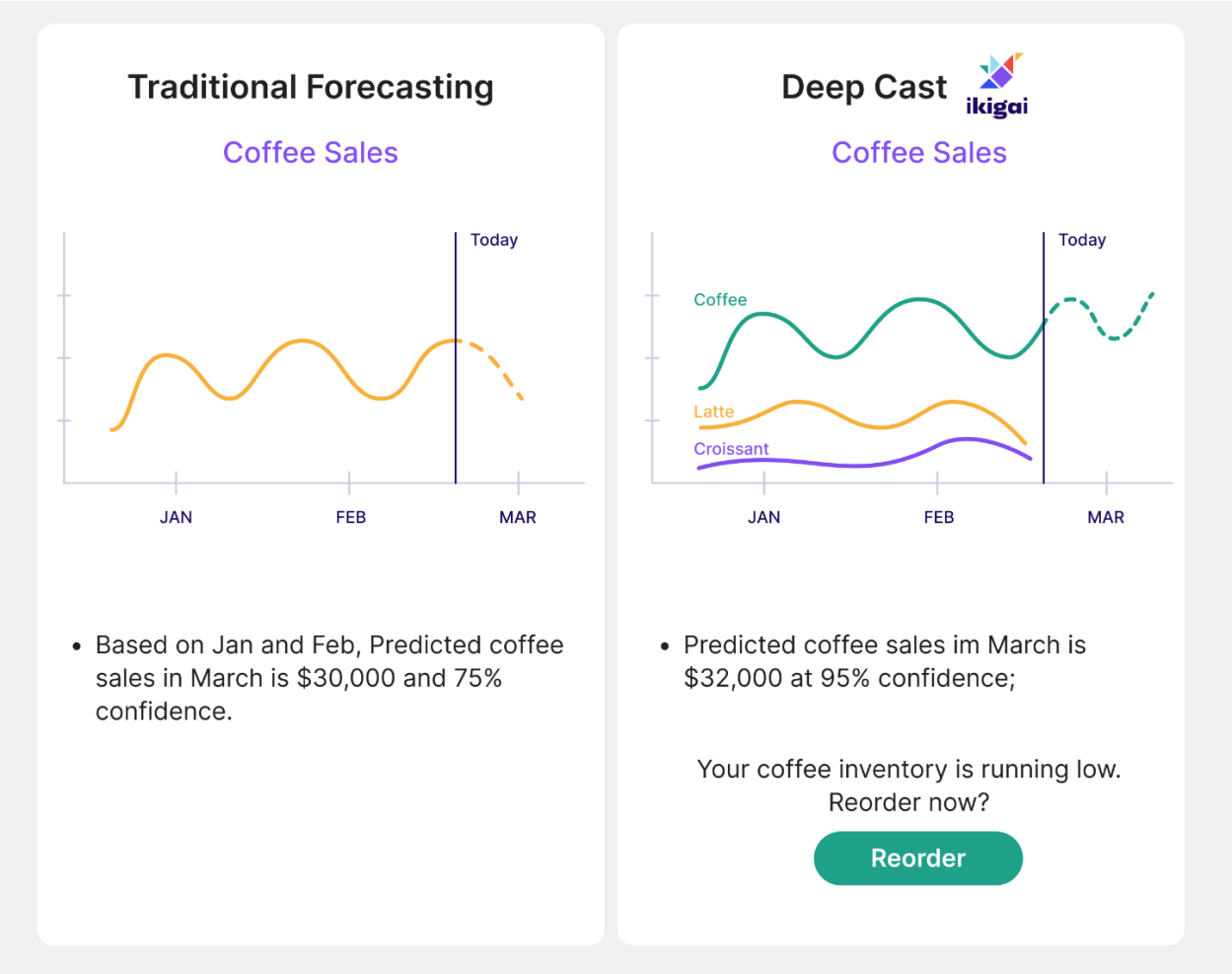
the rows. The Expert-in-The-Loop (EiTL) feature enables an end user to provide minimal supervision to correct the inaccuracies (exception is not an exception but a norm when using AI) and to improve the outcome.



DeepMatch [5] makes Ikigai's platform unique in its ability to serve various data reconciliation use cases including SKU mapping and new product launch in apparel retail, auditing in insurance, compliance in financial market, entity resolution across verticals and more.

DeepCast. *Predictions* (& inference) with *limited* data (+ MLOps). The task of prediction can be viewed as that of filling missing values in a spreadsheet.

Confidential © Ikigai Labs, 2023.



If it is temporal data, it would include imputation prior, historical data, or a forecast of future data. By viewing data from the lens of a large graphical model, such tasks can be answered by estimating the missing values, given other data as partial observations.

By combining the spatial and temporal structure of the data, Ikigai enables predictions with minimal data. The predictions come with uncertainty quantification as well as interpretation. What makes Ikigai's platform unique is its ability to serve various prediction use cases with limited data rather than requiring *a lot* of it. In simple terms, this seemingly counterintuitive goal is achieved by utilizing data *across* different sources. For example, to predict demand for *Cappuccino* at a cafe, in addition to using past data about *Cappuccino* sales, we also use information about sales of *Croissant* and *Latte*. This ability to learn across data sources utilizes graphical models in a similar manner that enables *DeepMatch*. To learn more about accurate demand forecasting in retail, see here [6].

Confidential © Ikigai Labs, 2023.

DeepPlan. *Generate scenarios* using historical data. A decision maker needs to weigh various scenarios from the lens of a collection of objectives and constraints to ultimately make a decision. From long-term strategy to daily tactical decisions, they require such scenario analysis.

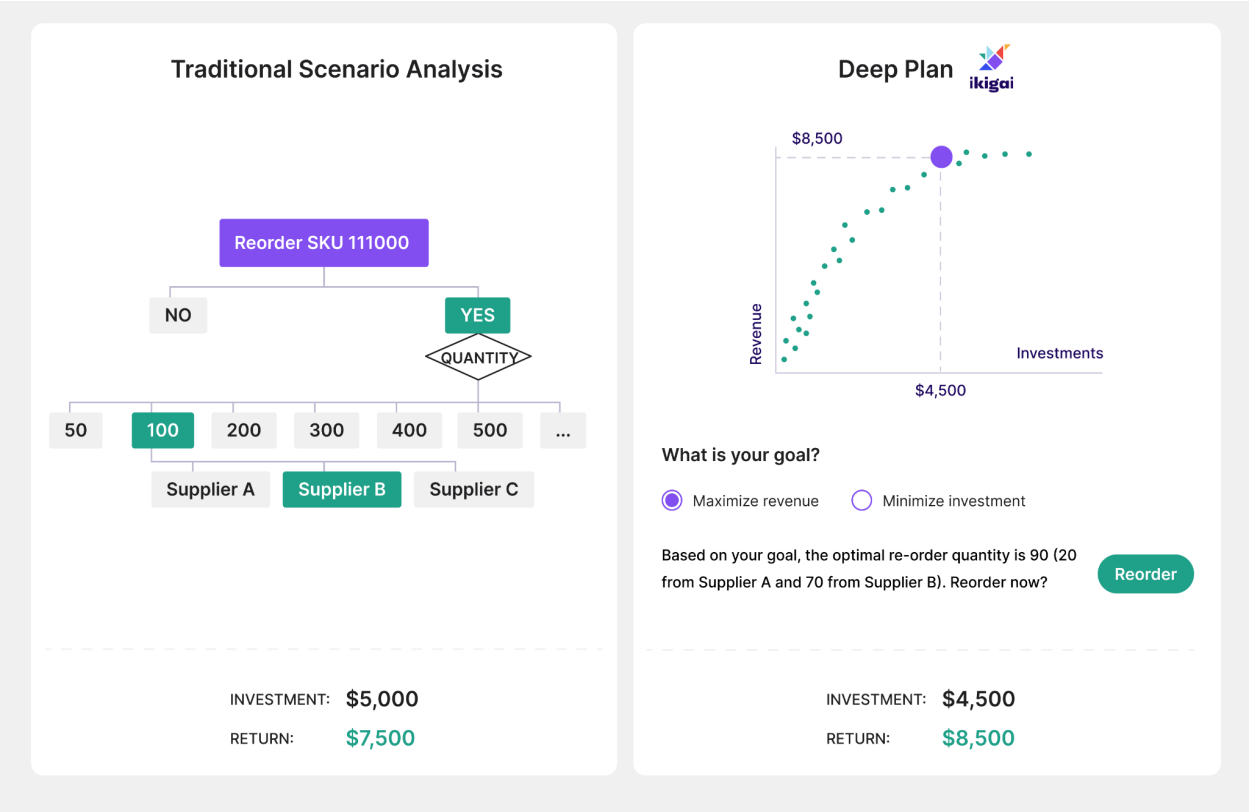
For example, consider coming up with a policy for deciding whether to accept a new customer based on limited Know-Your-Customer (KYC) information soon after they join your payment portal - if you are too careful, there will be *zero* customers on your portal; if you are too lenient, there will be a lot of losses due to fraudulent activities. It's about deciding amongst all plausible trade-offs between risk and reward.

Similar dilemmas occur with operational decisions like labor planning in supply chain, inventory planning, insurance rates for a product, and more. To evaluate potential trade-offs, in an ideal world, one would run multiple experiments. In each experiment, a different decision rule is utilized in operations, outcomes are measured, and risk versus reward are evaluated. But such experiments are not feasible - you have *only* one reality. Towards that, one would like to utilize historical data to run such experiments by building a simulator of the operations. Typically, this is a lot more complicated than simply making predictions.

For example, if an extreme KYC rule was implemented in reality, *no* customer would join the portal, in which case, it would be *impossible* to understand what the *fraudulent* behavior looked like under lenient KYC. Similarly, if there is no inventory to sell, it is not clear what demand or sales would realize if there was some inventory. Such challenges of *biased* data are addressed by what is called *Causal* inference [7][8]. Graphical models have provided universal language (for example, see a book by Turing Prize awardee Pearl [9]) and approaches to address these challenges of causally consistent simulations.

Ikigai learns a large *causal* graphical model to build a causally justified simulator. Using scalable reinforcement learning computational infrastructure [10] in addition to its computational technology, it has been able to provide the ability to perform scenario analysis for 10^{19} scenarios.

Confidential © Ikigai Labs, 2023.



To learn more about how it enables scenario analysis in the supply chain and its immediate business impact, see here [11].

Going forward

Ikigai's mission is to build self-driving organizations by bringing Experts and AI together. This will be achieved through building more than 500M-1.5B enterprise AI apps over the coming decade across verticals. And Ikigai's unique product powered by unique tech has provided initial proof that this daunting task can be achieved with its AI Apps platform - it is already powering AI Apps across supply chain, retail, financial services, banking, insurance, manufacturing and more.

But, there is a lot more that can be done beyond simply enablement with existing platforms. For example, LGM can enable other models such as **real-time recommendation systems.** As a consequence, it will provide the ability to *spin-up the modern* market-place entirely on Ikigai's platform within days. The computational

Confidential © Ikigai Labs, 2023.

architecture of PubSub-style message-passing, when enabled at hardware level, can provide unprecedented gains not yet realized. In short, the sky's the limit.

About Founders.

Vinayak Ramesh is co-founder, CEO at Ikigai Labs. He is a CS/AI MIT alum. He is a Forbes 30 under 30. He is a co-founder of WellFrame which is now part of HealthEdge (Blackrock). His graduate thesis at MIT invented the computing architecture for LGM.

Devavrat Shah is co-founder, CTO at Ikigai Labs. He is an Andrew (1956) and Erna Viterbi Professor of AI+Decisions at MIT. He founded the Statistics and Data Science Center, the academic unit, at MIT. He is a distinguished alumni of his alma-mater IIT-Bombay. He is a (Kavli) fellow of the National Academy of Sciences. He is a co-founder of Celect which is now part of Nike. He has made fundamental contributions to computing with graphical models, causal inference, stochastic networks, computational social choice, and information theory. His research has been recognized through paper prizes and career awards in Computer Science, Electrical Engineering and Operations Research.

References

[1] Algorithms for Inference (using Graphical Models) (6.438) by Devavrat Shah, MIT OpenCourseWare (OCW) [weblink].
[2] Academics contributing to graphical models from Devavrat Shah's group [weblink].
[3] Method and apparatus for graph based computing, Devavrat Shah and Vinayak Ramesh, Patent US 2020016714A1 [weblink].
[4] Message passing for large-scale distributed computation with factor graphs, Vinayak Ramesh, MIT [weblink].
[5] When you can't join, DeepMatch. Blog Post at Ikigai Labs [weblink].
[6] Forecasting with limited data. Blog Post at Ikigai Labs [weblink].
[7] Nobel Prize in Economics for Methodological Contributions to Analysis of Causal Relationships, David Card, J. Angrist and G. Imben [weblink].
[8] Synthetic Controls: Causal Inference with Observational Data, by A. Abadie, V. Misra and D. Shah, NeurIPS Tutorial, 2019 [weblink] .

Confidential © Ikigai Labs, 2023.

[9] Book of Why: New Science of Cause and Effect, Judea Pearl and Dana Mackenzie. [weblink].
[10] Ikigai at Ray Summit, 2022 [weblink].
[11] We know how much $$ you lost this past season, Blog Post at Ikigai Labs [weblink].

Confidential © Ikigai Labs, 2023.