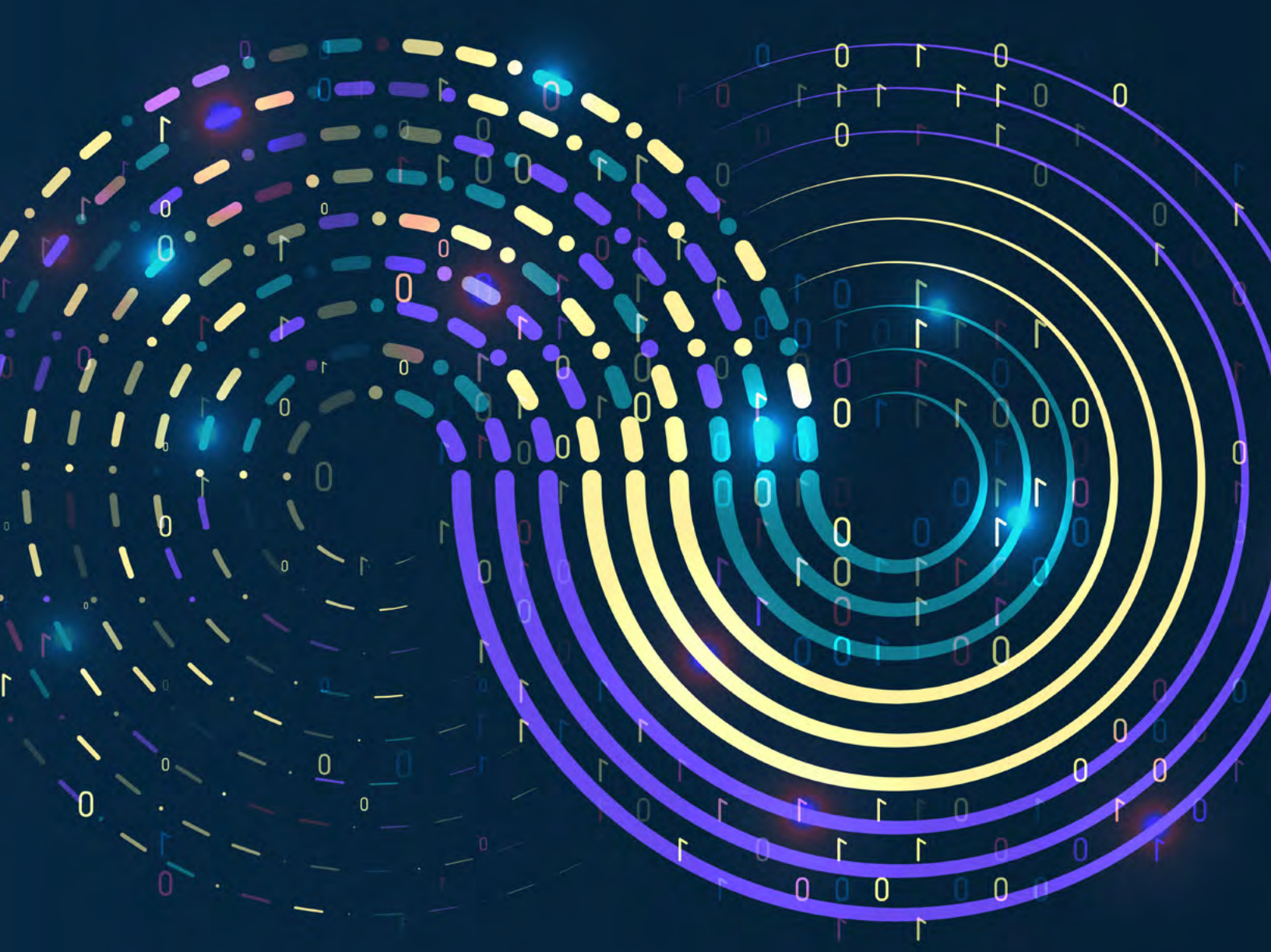


LF AI & DATA

Convergence of AI, BI and Data Analytics Future

Open Business and Artificial Intelligence Connectivity (OBAIC)

OBAIC

Contents

1. Where are we heading?
Current state of data science, democratization of ML and AI, and the rise of citizen data scientists 3

2 . Who is involved?
Converging paths of Citizen Data Scientists and Business Intelligence Users 6

3 . Why are there so many roles?
Evolving use cases of multi-persona data workers 8

4 . When will they really come along?
Business Intelligence and AI — so near yet so far 11

1. Data access 11

2. Creating Predictions 12

3. Visualizing Predictions 12

Expectation VS Reality 13

Multiple Disciplines 13

Accessing Data 13

Merging Data 13

The Connecting Bridge for AI and BI 14

5 . How can this be achieved?
OBAIC — The Vision 15

From ODBC to OBAIC 15

“That AI + BI = CI is too marketing and high level...” 16

What is NOT OBAIC then? 17

6 . What is the proposed plan?
Implementing OBAIC protocol 18

Model 19

MLSource 20

OBAIC Skeleton Code 23

7 . Which way can I embrace this?
Proposed plan for adopting of OBAIC 24

AI Vendors 25

BI Vendors 25

Conclusion and what’s next 27

BI & AI Committee Members 28

1 Where are we heading?

Current state of data science, democratization of ML and AI, and the rise of citizen data scientists

At the dawn of the century, it was stated that technology will play such a significant role in enterprises and organizations that they will have to become a technology company. Every single company will be a technology company was the prediction and it has come true. We are at another inflection point now. Every company will have to become an AI, ML and Data Science company. A report by McKinsey highlights the stakes at play: by 2030, companies that fully absorb AI could double their cash flow, while companies that do not could see a 20% decline without AI.

Data science is becoming more of a self-service proposition, and that is a good thing. Democratization of data and data science through a growing set of knowledge-sharing tools is expanding data science capabilities increasingly to business users or "citizens" in the broader working community. One of the top benefits of data democratization is improved Artificial Intelligence (AI) and Machine Learning (ML). Not only does data democratization reduce the likelihood of AI bias, but it also paves the way for democratized AI whereby, with the help of low-code solutions, companies can build armies of "citizen data scientists" capable of producing their own AI-powered applications.

Commoditized AI is propping up everywhere in the form of cloud-based APIs that you can use in AI applications and data flows. They provide you with pre-trained models that are ready to use in your application, requiring no data and no model training on your part.

These services leverage the latest deep learning algorithms and are also an indication that we have reached commoditization state in AI, at least in certain areas. These APIs provide services like text analysis, speech recognition and anomaly detection in time-series data, etc. They are consumed over HTTP REST interfaces and require minimal development effort for state-of-the-art AI services.

> "Who creates or generates models that use advanced diagnostic analytics or predictive and prescriptive capabilities, but whose primary job function is outside the field of statistics and data science."

Democratization of machine learning has also taken place with the development of simple drag-and-drop studios and tools accessible to those without doctorate degrees or deep data science training. These tools now feature convenient user interfaces that allow rapid building of analytic pipelines in web and collaborative environments with suggestive components that both guide the process and mitigate inherent complexity. Several ML platforms now offer access to many sophisticated ML models through a simple graphical UI. Microsoft, Amazon, SAS, and Google have rolled out software on cloud-based platforms that enable similar functionality. They enable you to create, manage, and view all the assets in your workspace and provide graphical tools such as a drag and drop interface for "no code" machine learning model development. Additionally, automated machine learning is also a permanent resident of these platforms. Automated Machine Learning (or AutoML as they are popularly called) is a wizard driven interface that enables you to train a model using a combination of algorithms and data preprocessing techniques to find the best model for your data. Users of these platforms can achieve all this without having a Mathematics or Data Science degree.

Now that we can access the compute power and data volumes necessary to operationalize tasks such as pattern recognition, anomaly detection or diagnosis, customer analytics, pricing, and predictive planning, we want ML systems that can learn to automatically prepare and perform data science functions with minimal programming. The irony in all this is that ML is often deployed as a digital surrogate for the data scientist, but one that requires the skills of a data scientist to be brought into existence. This is where the citizen data scientists come into the picture. This is a new breed of information workers that want to do more with the data but do not want to wait in the queue of the data science team. These citizen data scientists are more than mere consumers of analytic output. Gartner research

describes a citizen data scientist as someone: "Who creates or generates models that use advanced diagnostic analytics or predictive and prescriptive capabilities, but whose primary job function is outside the field of statistics and data science."

Let us take a simple example of a company dealing with customers. If Sally Sue, one of the sales managers at this company, wants to use data to target its customers most effectively with the items that they want to buy from this company, that is going to require analytic applications such as segmentation, propensity modeling, ad targeting, lead scoring, dynamic pricing and a range of concomitant data engineering and data science capabilities. All this requires a lot of effort from the resident data science team which is stretched thin. This gap is being closed by citizen data scientists by using the AI, ML and analytics operations tools mentioned earlier that automate the application of sophisticated underlying functions, techniques, and processes. There is no scenario where data scientists are going to update Sally's sales team on company's marketing progress, but marketing professionals are now wanting to create their own analyses of market interest and visualizations of customer engagement models in fulfilling that role. Citizen data scientists bring their own domain expertise and an understanding of the business problem to the table, then use commoditized AI, AutoML and drag-and-drop studio interfaces to perform specific data science functions like building, testing, and validating models. This democratization of data science is being executed across a population of people who do not have PhDs in statistics, but still need to digitize various business problems and demand tools that allow them to do so independently.

It is not a stretch to say that someday, soon, everyone in the enterprise will be a citizen data scientist to one degree or another, especially in the light of prediction that every enterprise needs to be an AI company. According to Gartner research, "Most organizations don't have enough data scientists consistently available throughout the business, but they do have plenty of skilled information analysts that could become citizen data scientists." It is clear that success of organizations will depend largely on the success of these citizen data scientists. However, two questions remain unanswered. Will there be enough citizen data scientists, and will they have the right tools to be successful?

2 Who is involved?

Converging paths of Citizen Data Scientists and Business Intelligence Users

The ongoing shortage of data scientists has been well documented. According to a McKinsey report, the United States was facing a shortage of approximately 140,000 data scientists in 2020. This number will only grow exponentially in the current decade. Even as the business world grows increasingly digitized and reliant on big data modeling and analytics to drive value and profit, those possessing the requisite education and expertise in mathematics, statistics, data prep, programming, and distributed computing to meet data science challenges are rare. The ability to make sense of the enormous troves of transaction, customer, and equipment data across digitized industries has become a premium skillset, and the recent explosion in ML and AI capabilities has compounded the problem.

There has also been a growing recognition that better utilization of those rare skillsets can address the shortage in data scientists. A large part of what data scientists do today are functions that can be divided and redistributed. Skills in data wrangling, cleaning, and preparation or in data modeling and data processing systems have led to classifications such as "data engineer," "data architect," "data wrangler," and the like — which serve to remove some of the workload burden from data scientists. However, organizations are increasingly stating that it is not enough. What they need are citizen data scientists who will create the data driven business outcome and not just prepare data for the increasingly overworked data scientist.

IDC big data analytics and artificial intelligence research director Chwee Kan Chua notes that in the face of the data scientist shortage, "lowering the barriers to allow even non-technical

business users to be 'data scientists' is a great approach." The idea is to utilize intuitive, often ML-enhanced, tools within the enterprise that enable these citizen data scientists to develop and administer focused analytics models for specific kinds of business analyses via wizards, templates, and dashboards. Further, the results of these efforts can be interpreted and applied for the benefit of other line-of-business users. There is a collaborative aspect underpinning the citizen data science phenomenon that aims to amplify data science knowledge, but also to scale domain expertise and business acumen as well.

An e-book, **The History of Business Intelligence and its Evolution,** reveals how organizations are gradually trusting the citizen data scientists to use automated BI tools for data-driven insights. In other words, the citizen data scientists have been empowered to make sense with data without having a sound knowledge of statistics or mathematics. However, the citizen data scientist role is not for everyone and identifying candidates for it remains a challenge. It is best suited for individuals with an affinity for information systems, but also patience, excellent communication and consultative skills, and an ability to translate between the business problem and the tools that can be used to solve it — a grasp of both the competitive requirements of the business and the practical constraints of the IT infrastructure. As industries have digitized, and information systems have become more widely recognized as strategic assets, many organizations have gotten better at identifying the people who have these qualities, but many are not able to see what is hiding in plain sight — Business Intelligence (BI) users. BI users have these qualities and have been playing with the data for decades. In addition, they might be the only group of IT that has equal understanding of business and technology.

The rise of citizen data scientist shows that the lines between BI and data science have started to blur. It is also an indication of how business users are increasingly becoming data workers. Their needs are changing from just using data to report "what happened last quarter" to project "what will happen next quarter" and to predict "the next best action for their business". Industry must take steps to provide tooling to these business users that are data workers and now being rechristened as citizen data scientists. These citizen data scientists are increasingly the BI users of yesteryears. Empowering BI users might be the only way to cover the skills gap and address the shortage of data scientists in today's organizations.

There is no dearth of BI users in organizations. This is a demographic dividend that is in favor of data science. It will be a disservice to data science if we do not cash this demographic dividend effectively by empowering BI users to become citizen data scientists.

3 Why are there so many roles?

Evolving use cases of multi-persona data workers

Sally Sue Somebody never intended to be a Data Scientist; her career just seemed to evolve that way. She graduated business school, ready to change the world. In her first role, she found herself sitting at her desk for hours on end, with a giant straight edge evaluating numbers on reams of green-bar paper containing detailed data values from one system or another. When she spotted anomalies, she would scurry through a maze of cubicles to her bosses' office where they would spend an hour walking through sticky notes and highlights. The best part was at the beginning of the next month, a new shipment of green bar reports would arrive and Sally Sue would do it all over again.

Over the course of several years, the company provided new and improved technology: first for reporting and then for business intelligence. Sally Sue began writing requirements for her organization so that others could build aggregated reports and applications for her. Once the requirements were estimated, prioritized and delivered, she could analyze the data from a much wider angle. She had become a business analyst instead of just being the data analyst and data steward for her organization. Quite a change for Sally Sue, and her organization.

She soon became very adept at taking screen shots from her applications, pasting them into presentations and annotating them to help reviews go faster. Thus, Sally took on the unofficial persona of PowerPoint Guru. As access to data increased, and the speed with which Sally Sue could access it increased, even that role became old news. Sally Sue soon found herself as the Chief Data Storyteller for her organization. She brought decision makers to tears when tears were needed, and to gut wrenching laughter when that was needed in order to get them to actually act on results instead of just listening to them.

Figure 1: "The Evolution of Sally Sue" AI-generated by MidJourney

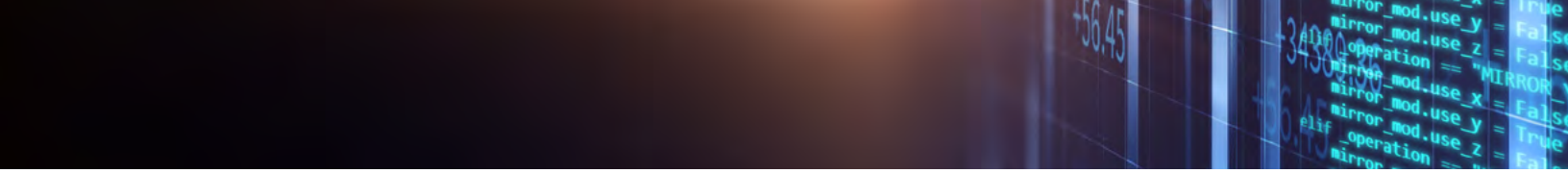
Before you knew it, the company was heading down the self-service route. Instead of writing requirements documents for others, Sally Sue found herself trying to surface her own data. Of course, it had been years since single data sets were enough to answer questions. The company needed her to report insights from a myriad of data sources: Right, Inner, Left, Outer, Full, Cartesian Joins weren't things she was comfortable with. Becoming a SQL Coder was a persona the company needed her to perform. So, she did.

Her organization began realizing that their data was streaming in faster than they could have others cleanse and transform it for Sally Sue. As you probably guessed, in addition to everything else she handled, Data Engineer became her newest persona. If that seems like a lot, her story didn't end there. You see, in addition to a myriad of other issues in their raw data, it seems that the company had 87 different spellings just for the city of Philadelphia, Pennsylvania. Sally Sue needed to add Data Quality Engineer to the myriad of her growing talents.

Soon the company wanted to democratize data and put it in the hands of more people. Sally Sue was required to start building screens for others to use as well as herself. Application Designer was a nice persona for her to add, but it proved to be the most challenging she had faced. She had to learn how to train people who were not data literate, to access data few had any real business knowledge about, all while handling a constant stream of change requests.

Somewhere along the way in her career the company introduced a series of data governance initiatives. She believed in what the process stood for, but quite often, it seemed like roadblocks slowing her down. This was the last thing she needed, as many of the projects she had already completed needed to be redone as the company was moving its storage for some systems to the cloud.

Sally Sue is of course a fictional character, yet I'm quite sure you have either worked with her, or look at her image in the mirror every morning. She represents the millions upon



millions of data workers who have gone through immense transformation for their companies as part of this digital age. When she started, she saw data in a protective bubble. But as her companies needs grew, she had to come out of that bubble and learn how to collaborate. She had to grow from asking others what she needed, to having others ask her for what they needed.

Undoubtedly, some have taken on mind boggling new personas because their companies are just thinly staffed for cost cutting measures. But for most, the organizational uses cases around data simply demanded the evolution.

Dara Such, Vice President and Publisher of TechTarget, shared a great article back in 2020 to help companies market their products better titled "Expert Insight on Evolving Your Personas for Changing Enterprise Tech Environments." One of the most powerful headings Personas were never real people really captures the heart of what so many have missed. Positions only exist to fulfill the needs of the use cases the company has at the time.

Organizations, like Sally Sue's, have gone from turning raw data into information into knowledge, knowledge into insights, insights into driving decisions, and decisions into taking action — all while driving those things closer and closer to the decision maker's fingertips, quicker and quicker each day. But they aren't resting on their laurels, and neither will Sally Sue.

Sally's organization has enjoyed some fantastic return on investments made in their data science initiatives. Now they are undertaking measures to help drive predictions nearer and nearer to the decision makers fingertips via automated machine learning. Sally Sue will of course be on the bleeding edge of this movement.

No one can guarantee many things in life because there are just way too many variables; but what can be guaranteed is that Sally Sue is going to fail often as she learns her new role as a citizen data scientist. Just as she failed initially in taking on all her other personas. But soon she will be able to navigate her way through AutoML to generate things like customer churn and credit risk predictions. Soon after that, she will be able to deploy them within the business intelligence tool for others to take advantage of, while she learns how to solve more complicated tasks.

If you happen to see Sally Sue walking down the hallways in your organization, or on your next video conference, and she seems really frantic … please be kind to her. She has come a long way from just using a straight edge on green bar paper.

4 When will they really come along?

Business Intelligence and AI — so near yet so far

Business Intelligence and AI go hand-in-hand. One of the most important aspects of AI is visualization. Whether it's graphs, a number on a webpage, or a dynamic dashboard displaying key performance metrics and predictions, AI and ML models thrive off of visualization. These models are often complex and difficult to understand, but their resulting predictions are generally easy to understand, such as customer churn, credit risk, or identifying pedestrians in a crosswalk. In the business world, BI tools are the number one way to display results of AI/ML models.

All AI/ML models require two important pieces of information:

- ***Data*** — Input data is fed into the model which produces a prediction
- ***Model*** — A packaged model or code that can produce a prediction from data

If you have these two pieces of information, you can harness the power of machine learning and add predictions into a BI tool. It seems simple enough, but why is it so difficult? Let's consider the steps needed to get those predictions into a BI tool.

1. Data access

Today's data resides in a number of formats and technologies: traditional relational databases, NoSQL databases, unstructured data, and proprietary formats. There are hundreds of ways to store and represent data. Accessing data in any of these formats is an inherent

challenge. It requires authentication, knowing how to read its format, performant querying, and a secure way of transporting it to its final destination. Modern BI tools and analytical platforms enable low-code, no-code access to data via data connectors.

Sally Sue's organization has finally hired a visualization expert, John Doe. He can easily load data into a BI tool and start creating visualizations right away. He knows very little about code or authentication to access the data. User-friendly tools for data access and authentication are already provided for him in the BI application. But what if he wants to create predictions from that data?

2. Creating Predictions

Citizen data scientists like Sally Sue are well aware of the process to create predictions: save your model into a transportable format such as ONNX, Pickle or PMML, feed input data into it, and get predictions. The process is well-documented, straightforward, and easy to use for programmers; however, John Doe is not a data scientist well-versed in most data science programming languages. He can run basic SQL queries and build beautiful dashboards through point-and-click interfaces. He can even perform predictive modeling within his application if it supports built-in automated modeling tools, but he does not have the skillset to use an external model created by a data scientist. If John needs a specific predictive model, he must put in a request to Sally to add those predictions to data that he can access. This is a multi-step process requiring not only a data scientist, but often a data engineer who can merge predictive results into a final output table that will be used by John. Sally Sue will work to create a production-ready, end-to-end process that satisfies John's request. Depending on the complexity of the model, this could take weeks or even months to build and test.

3. Visualizing Predictions

Because John is building a dashboard that requires predictions, he needs to have those predictions within the data he is using to build the dashboard. Sally Sue creates a process that merges the predictions into the final dashboard data. It is then uploaded to a database accessible by his BI application. John's modeling request is fulfilled, and he can start building a dashboard with predictions, KPIs, and more. There is one problem: the data is static. John's stakeholders would like to perform what-if analysis directly through the application and vary input values to see how it affects predictions. To do this, John starts a new project with Sally Sue and a web developer to create a custom application that enables what-if analysis for models. His department has not built a tool like this before: the project takes months to complete due to its overall complexity and even requires outside consulting help since his company does not have all of the necessary skillsets to complete the task. They need to authenticate to access external data, provide the ability of users to upload their own data, allow users to save scenarios, prevent people from overwriting each other's scenarios, perform security testing, and more. In the end, the project is completed, but it took a great deal of effort and requires constant post-production support from already thinly spread teams.

Expectation VS Reality

The overall concept is simple: provide input data, feed it into a model, and output it somewhere. This should be easy for someone like John Doe to do - how did it get so complex so quickly?

Multiple Disciplines

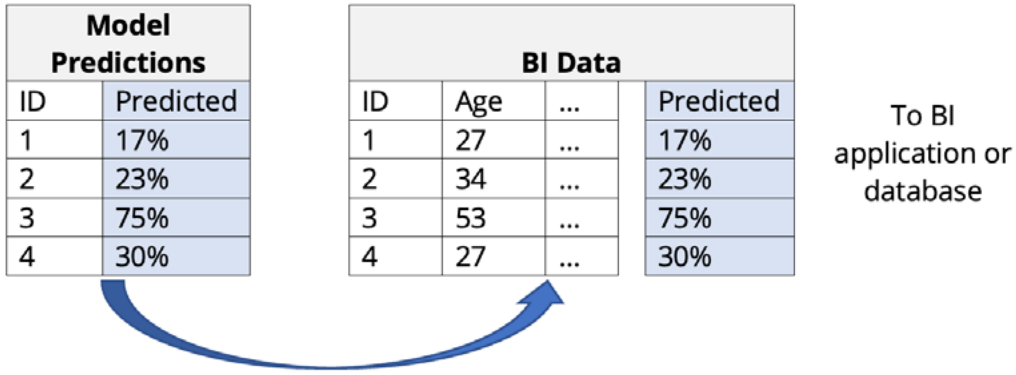
Creating a model and visualizing it for BI purposes is a multi-disciplinary process. At minimum, it requires a data engineer, data scientist, and a visualization / BI expert. Sally Sue is a citizen data scientist and a generalist. She can help out in each of these areas, but she is by no means a dedicated expert in each area. As a result, running all of the steps to complete this task takes longer if she alone performs them all. Instead, she enlists people like John Doe to help create compelling dashboards that enable her to tell a story.

Accessing Data

Data access is complex and must be done securely. BI tools access data in numerous ways, and some may pull in data to run in-memory. This means data is copied from a remote database and stored in a potentially proprietary format that is ideal for reporting. Sending data to and from these databases and in-memory formats can be a multi-step, potentially complex task.

Merging Data

Because predictions cannot be made directly from the BI application, they must first be generated and merged back with the desired dataset:



This is typically done programmatically and as a part of the final ETL process before loading into a BI application or database. If the data is large, this can take a significant number of computing resources to complete. Additionally, the predictions are entirely static. The entire process must be run again to generate new predictions. In a cloud-based environment, this costs money.

The Connecting Bridge for AI and BI

In the end, a model merely needs inputs to create predictions. To create visualizations, a BI tool merely needs a dataset. If John Doe or Sally Sue already have data loaded into a BI tool with all of the inputs needed to make a prediction, what if they could call a model to generate those predictions on the fly and eliminate the intermediate steps? All they would need to do is upload a dataset and select a model. They can even vary those values to generate new predictions and perform what-if, bypassing the need to construct a custom application. To do this, they would need something that can get data from their BI application to a model, then receive predictions. Since there are so many databases and analytic platforms, it would have to be standardized so they can access models using any BI tool and any analytic platform. Enter: OBAIC.

5 How can this be achieved?

OBAIC — The Vision

As we explained in the previous sections, there are 3 key intermingle components impacting the way how different roles in the Analytics process evolved over the past years - AI, BI, and Data. When the BI & AI committee discussed and pondered what can and should be done to take the advantage of these 3 areas to streamline the flow, we realized that a similar problem was actually raised and solved 30 years ago in 1992, when there was an exploding number of new databases management systems (DBMSs) coming to the market. That made communicating to the databases a nightmare because users needed to learn different languages when using different databases. The solution? Open Database Connectivity, or better known as ODBC.

From ODBC to OBAIC

By standardizing the interface for applications to access various DBMSs, ODBC opened the door for coopetition and allow databases to focus on creating their own "secret sauces" for building a better database while enabling consumers to access all databases in a unified way. Can we adopt this idea into now a more complicated scenario when we have AI, BI, and Data in the mix? This is the question leading to the birth of Open Business and Artificial Intelligence Connectivity (OBAIC) — pronounced "O-Bay-egg."

Figure 2: OBAIC logo

OBAIC borrows the concept from ODBC with a vision to unite AI, BI, and Data together and reduce the friction in the analytics process. As shown in the OBAIC logo, the 3 arches represent AI, BI, and Data. OBAIC is like a bridge connecting them all. OBAIC defines a protocol to facilitate exchanging machine learning models and data between AI and BI. This makes each component focus on what they can do the best but at the same time complement each other to enhance the overall user experience in analytics. For example, in the past 20 years, BI has grown from just a table with rows and columns, to a simple line chart representing trends over time, to a very rich interface allowing users to interact and perceive data right inside a well-designed dashboard. The interface is very intuitive and provides direction for guiding people to make sense of data in a visual way. However, as AI started blooming in the past few years, everyone jumped onto the wagon. This also creates a lot of great quality open-source projects and communities just contribute thousands of lines of code to make AI the steroid of analytics — finding trends and hidden patterns from data at the speed that no one human being is able to attain. When combining the speed of AI can learn from data with the direction BI can guide us intuitively, that is the Cognitive Intelligence (CI) - the next generation of analytics we should seek for.

"That AI + BI = CI is too marketing and high level..."

If you follow our publication, this concept of AI + BI = CI is not new. In fact, this idea was shared back in our 2019 publication **BI Endgame — When AI meets BI.** Over the past 2 years, this concept has been scrutinized by our committee to bring it to practicality. We will see this from 3 perspectives:

1. BI : "OBAIC is an extension of my brain"

As long as a BI tool understands ODBC, it can connect to all databases supporting this protocol. Similarly, before OBAIC, when a BI tool wants to incorporate a new AI platform, a new module is developed to connect to that one single AI platform. When another new AI comes, the process repeats. With OBAIC, as long as the BI tool speaks this standardized protocol, it can connect to all AI platforms compliant with OBAIC. Information about the ML models being managed by the AI platform will then be available to the BI tool, via OBAIC. The BI tool uses this ML information to offer the result to its users using its own unique interface and features (i.e., the BI tool's "secret sauce"). For example, a BI tool could incorporate features to make it easier to select the correct model based on the data loaded into the tool. Assume that among 100 models, 5 of them match the criteria of fraud analysis the user is looking for. The data set the user currently possesses aligns with 2 of the 5. Hence the BI tool can offer the 2 models to its user with the scoring of the 2 models provide already included in the ML information. The user can review and initiate an inference request by using the data he/she owns with the selected model. The request will then be sent to the AI platform, again via OBAIC, and results will return after the execution.

2. AI: "OBAIC provides one channel of multiple outlets"

In ODBC, once a database provides an implementation once, it can be reached by a number of customers querying the database. For OBAIC, if an AI vendor implements this protocol once, customers including BI and other applications can call and request services provided by the AI platform. One service can include helping the requestor to process inference and return the result as mentioned above. The AI platform can also provide a model training service by accepting a data set, by value or by reference depending on the situation, and returning a trained model for future inference. The main customer base for an AI platform is a data scientist, who knows how to write in data science-oriented languages and apply ML libraries like TensorFlow, PyTorch, and Scikit-Learn. OBAIC democratizes the work of the data scientist by exposing the ML models to another untapped market. This market wants a more powerful way to analyze the data, but it may not have the right resources and talents to achieve that.

3. Data: "OBAIC optimizes the use of me"

BI users and data scientists are two groups of people with different skill sets. Their commonality: find value in the data. OBAIC unleashes the data by enabling both groups to extend beyond their original circle. BI users can find hidden trends which may never be found manually due to the number of dimensions in the data set they are dealing with. Data scientists can deliver their model to the hands of someone who knows the business context inside out and can point out the direction in the analytics from their heart. That's how OBAIC optimizes the use of data.

What is NOT OBAIC then?

We have a big vision for OBAIC. However, we are proposing a protocol for how this should work in this Phase 1. We are also planning to recruit more brainpower to develop a framework for people to use in Phase 2. However, we are NOT trying to implement OBAIC for AI or BI platforms. The development is totally up to the vendor to be compliant with the proposed standard. Just like ODBC, it's up to the database or third-party vendor to implement the driver, and the caller to configure it properly and query the database accordingly to the standard. The following section describes the proposed plan to implement OBAIC so third-party vendors can integrate it with their tools.

6 What is the proposed plan?

Implementing OBAIC protocol

The principal vision for OBAIC is to define a set of portable interfaces to encourage similarity between Python SDKs built for various ML engines and one that can be readily adopted by Business Intelligence (BI) Clients. To that end, a proposal for OBAIC API is provided below that allows clients to access model catalogs and run predictions against a dataset. Python has been chosen as the programming language for the SDK because of its wide use among the data science and business intelligence developer community.

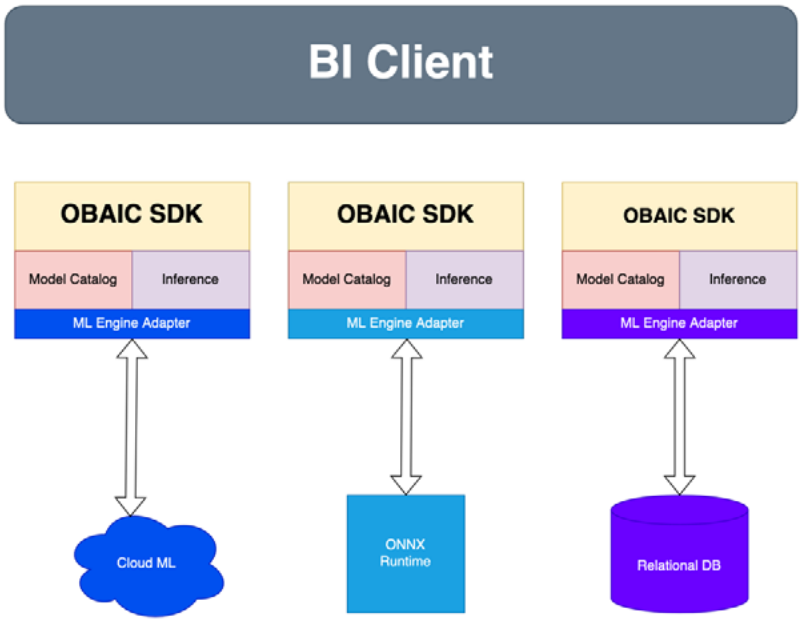


Figure 3: OBAIC High-Level Architecture

The OBAIC SDK layer exposes a common set of interfaces to connect securely to an ML Engine, query its model catalog and run inferences against it on a given dataset. Anyone desiring to support OBAIC for any given ML platform would have to provide an ML Engine Adapter that implements the interfaces required by the OBAIC SDK. The ML Engine Adapter is solely responsible for interactions with the ML platform and translates the requests from OBAIC SDK to requests supported by those platforms such as REST API, gRPC or even SQL. The BI client is thus able to access any ML platform through the abstraction provided by OBAIC.

Model

https://github.com/odpi/OBAIC/blob/main/obaic/model.py

OBAIC proposes a representation for an ML model ObaicModel that is suitable for BI clients to drive their workflows. The ML engine adapter retrieves a model in their native format and enhances it using one or more optional attributes described below before returning it to the client. BI clients may use these additional attributes as filters when retrieving models and to picking a model best suited for the problem at hand.

ML engine adapters may choose to cache OBAIC enhanced models in local storage or in a remote repository for better performance.

Each model has a unique identifier id, along with `name, description, revision, creator`, and `format` such as ONNX or PMML.

`url: str`

- Provides the URL to retrieve the native model from the ML source

`algorithm: str`

- The algorithm used by the model such as “Artificial neural network”, “Decision trees”, “Bayesian networks”. If the algorithm is not open, it may be indicated simply as “Proprietary”.

`tags: List[str]`

- A set of tags indicating related application categories such as “Banking”, “Anamoly detection”, “Sentiment analysis” and so on.

`features: List[ObaicModelField]`

- A list of type `ObaicModelField` which defines the features accepted by the model. `ObaicModelField` describes the operations, data type, taxonomy (business meaning of the feature if available).

predictions: List[ObaicModelField]

- A list of type `ObaicModelField` which represents the outputs of the model. The prediction of a linear regression model may be continuous while that of a classification model may be distinct labels.

performance: List[ObaicModelMetric]

- A list of type `ObaicModelMetric` that provides various performance metrics of the model such as accuracy, precision, recall, ROC, Log loss and so on.

rating: int

- Represents a model rating given to the model

MLSource

https://github.com/odpi/OBAIC/blob/main/obaic/source.py

The MLSource class provides the common set of interfaces for OBAIC workflows.

CONSTRUCTOR

```
def __init__(self, ml_src_parameters: dict[str, str],
                   ml_src_credentials: dict[str, str])
```

- An MLSource object is instantiated using two keyword arguments `ml_src_parameters` and `ml_src_credential`.

ml_src_parameters: dict[str, str]

- A dictionary that holds parameters required to connect to the ML engine as key-value pairs

ml_src_credential: dict[str, str]

- A dictionary that holds parameters required to authenticate to the ML engine as key-value pairs. The BI Client may authenticate with the ML engine using a variety of methods such as passwords, token, X509 certificate, or key file.

MODEL CATALOG METHODS

```
@abstractmethod
def list_models(self, model_query: str) -> Iterable[ObaicModel]
```

- The function fetches models from an ML model repository. The function accepts a `model_query` as a keyword argument and returns an `Iterable` to a collection of `ObaicModel`

model_query: str

- A query string key-pair format representing a filter query to retrieve models. A typical query such as "`algorithm=ann&limit=5`" returns 5 models that use ANN algorithm.

```
@abstractmethod
async def list_models_async(self, _model_query: str) -> Iterable[ObaicModel]
```

- The function fetches models from an ML model repository asynchronously.

INFERENCE METHODS

```
@abstractmethod
def predict_with_features(self, model: ObaicModel,
                features: list[ObaicValue])-> Iterable[ObaicOutput]
```

- The function labels or scores features using a model and returns an `Iterable` of `ObaicOutput`. Since the feature data is sent to this function, the responses are returned synchronously. Typically, this function is used for feature datasets that are small.

model: ObaicModel

- A model to use for labelling or scoring.

features: List[ObaicFeature]

- A collection of `ObaicFeatures` that need to be labelled or scored by the model. Each `ObaicFeature` is a Sequence of values for that feature.

```
@abstractmethod
async def predict_with_data_ref_async(self, model: ObaicModel,
                data_src_parameters: Dict[str, str],
                data_src_credentials: Dict[str, str],
                data_query: str) -> Iterable[ObaicOutput]
```

- The function labels or scores features using a model and returns an `Iterable` of `ObaicOutput`. This function returns responses asynchronously. The feature data is not sent to this function. Instead, data source parameters, credentials and a query string are passed to this function.

The ML engine adapter may use these parameters to retrieve the feature dataset.

In case, an ML engine supports OBAIC natively, then this function may simply passthrough the call without fetching the dataset. The ML engine would use the data query to fetch feature dataset thus minimizing the movement of data.

model: ObaicModel

- A model to use for labelling or scoring.

`data_src_parameters: dict[str, str]`

- A dictionary that holds parameters required to connect to a data source as key-value pairs from which to retrieve features

`data_src_credential: dict[str, str]`

- A dictionary that holds parameters required to authenticate to the data source as key-value pairs

`data_query: str`

- A query on the data source to retrieve features for inference.

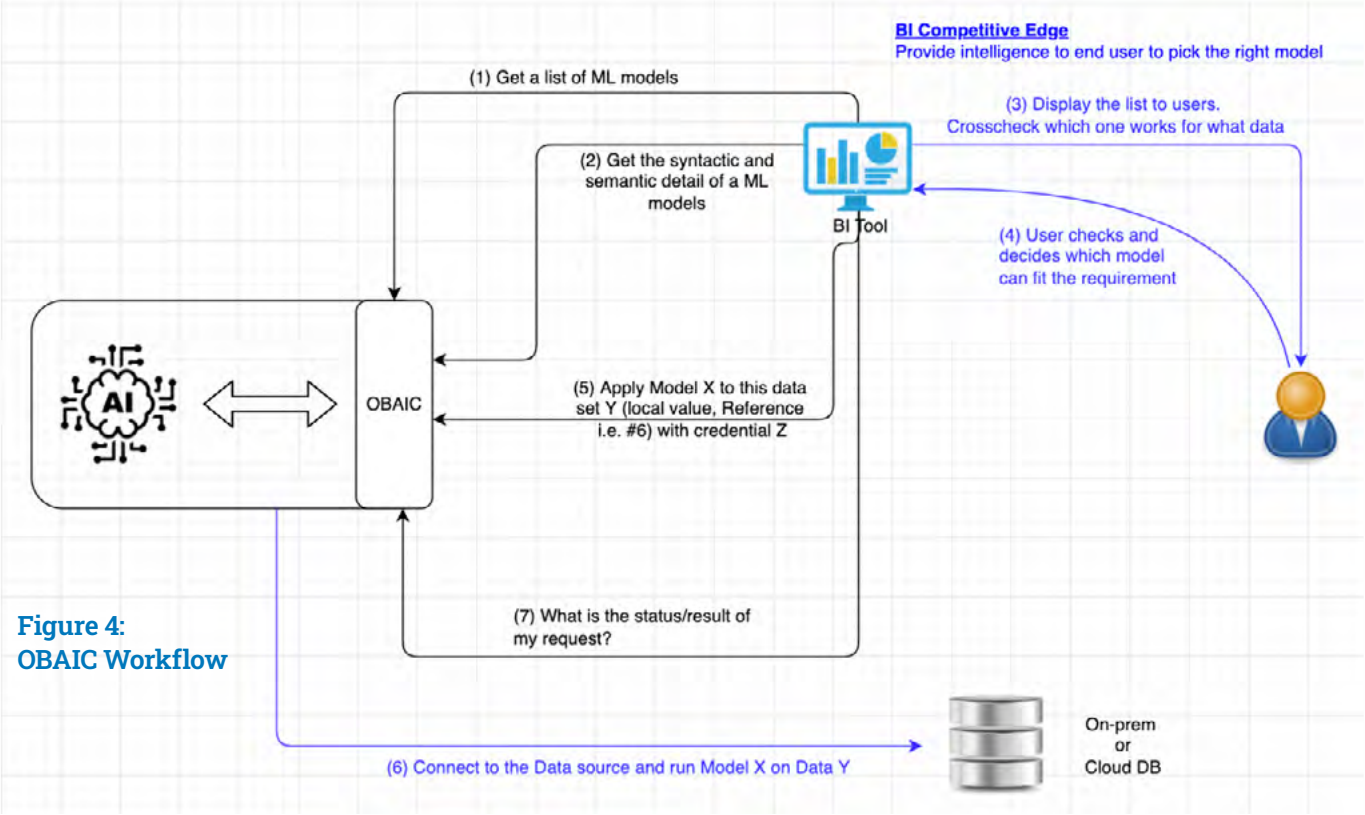


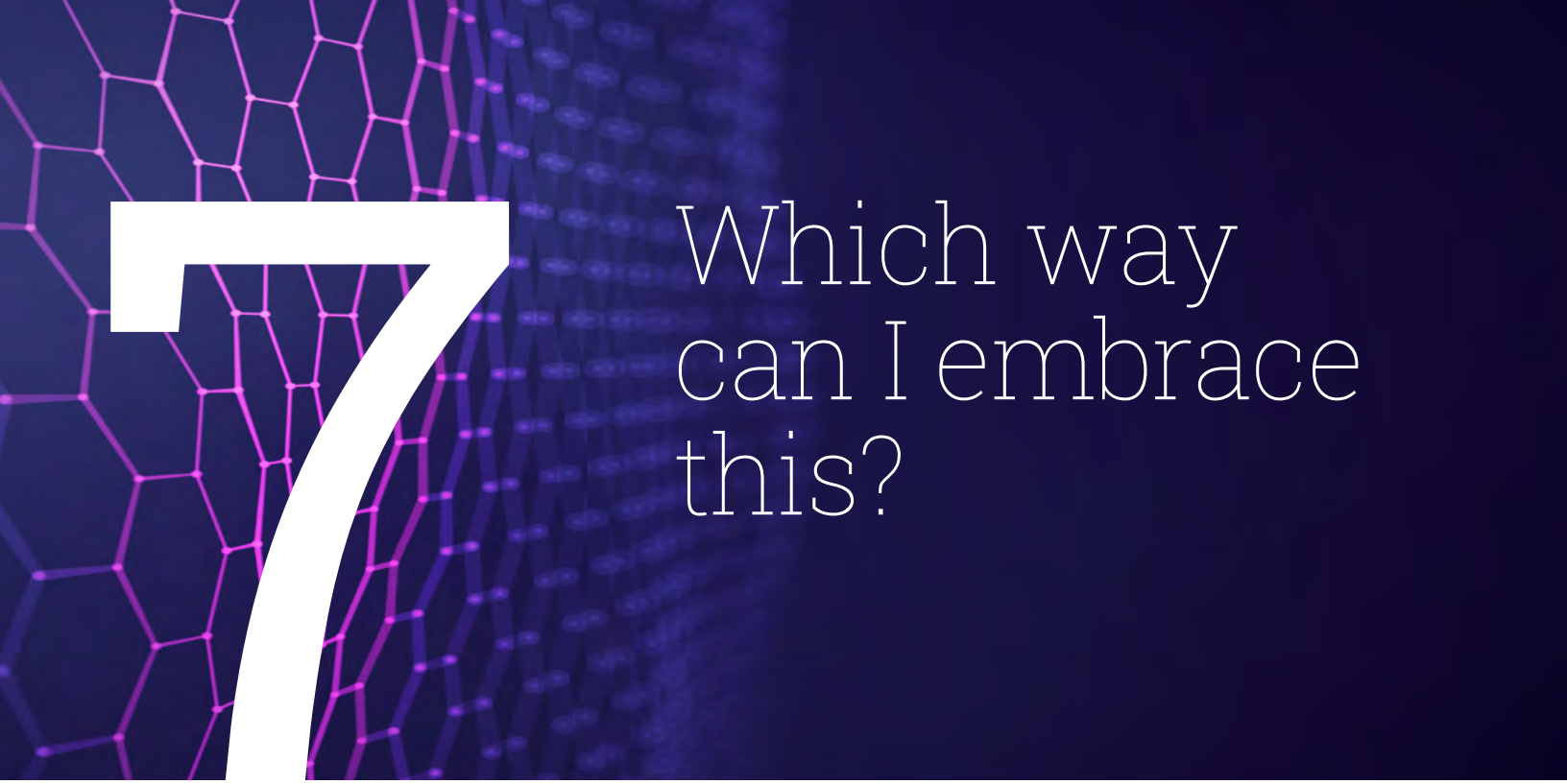
Figure 4:
OBAIC Workflow

OBAIC Skeleton Code

https://github.com/odpi/OBAIC/blob/main/sample.py

A skeleton code showing the OBAIC SDK calls is provided here to bring all of the pieces together.

```
from collections.abc import Iterable
from typing import List, Dict

from obaic.model import ObaicModel
from obaic.source import MLSource, ObaicFeature, ObaicOutput

def main():

   ml_src_parameters: Dict[str, str]
   ml_src_credentials: Dict[str, str]

   # define an ML source where predictions will be done
   ml_src: MLSource = MLSource(ml_src_parameters=ml_src_parameters,
                               ml_src_credentials=ml_src_credentials)

   # 1. define a model query and fetch a list of models from the ML source
   model_query: str
   ml_model_iter: Iterable[ObaicModel] = ml_src.list_models(model_query=model_query)

   # 3. pick a model from the list mlModels using some criteria
   for ml_model in ml_model_iter:
        if ml_model : # check for some criteria
            select_ml_model = ml_model
            break

   features: List[ObaicFeature]
   # 4. define a prediction request and run a prediction using the model selected above
      prediction_response: List[ObaicOutput] = ml_src.predict_with_features
      (model=select_ml_model, features=features)
```



7 Which way can I embrace this?

Proposed plan for adopting of OBAIC

Facilitating the three key intermingled components is challenging. No single vendor can easily navigate the journey alone, given the diversity in current and future ecosystems jointly developed by enterprises and open-source communities. Connecting one BI platform and one AI platform requires expertise in both platforms, not to mention significant time and effort needed behind the scenes. Thus, the best strategy is to adopt OBAIC and move forward with vendor implemented adapter / driver.

At the early stage of OBAIC adoption, vendors might need to consider implementing their own version of an adapter / driver and an OBAIC manager following the protocol. However, we are expecting some independent software vendors will join in to provide a cross platform OBAIC manager along with a set of adapters / drivers for selected vendors, like Progress or Simba's commercial offering for ODBC, as well as open-sourced alternatives like unixODBC.

OBAIC is a multi-vendor, multi-source technology for accessing data and ML models across broadly separated, loosely deployed, multi- and heterogeneous platforms. Even though the API examples given in the implementation section is using Python, vendors could choose other languages best suited for their platforms when adopting OBAIC.

AI Vendors

The OBAIC adapter for ML engines is implemented on top of the APIs provided by AI platform. If the OBAIC adapter is in place, capabilities of the AI platform are ready to be consumed by BI platforms. The benefit of providing a vendor version of the OBAIC adapter

is that implementation details can be hidden nicely. No matter what heterogeneous environment the AI platform is, and the internal format of ML models used, OBAIC clients can talk to the AI platform merely through the adapter. However, when developing AI models, remaining compatible with open standards for model formats like PMML and ONNX is a good choice as OBAIC leverages these open standards when designing the protocol.

The following capabilities should be planned when implementing the OBAIC adapter:

- Collect and organize ML models created on AI platform into model catalog
- Extract tags from each ML model and expose tags along with syntactic details in model catalog
- Provide connectors to digest raw data, or references to data carried in OBAIC API, and then connect to external source to fetch data
- Manage computing resources needed for making inference using user specified models
- Provide the status and result of model inference, as well as the explanations to the results
- Design and expose a set of parameters for training a ML model and manage computing resources needed for the training job (*)

OBAIC manager is the bridge that connect requests from BI to the capabilities of AI vendor. It provides a set of common APIs that can be used to connect to multiple AI platforms with corresponding adapters. AI vendors choosing to provide OBAIC manager may build the function into the OBAIC adapter, like the existing CLI drivers provided by Database vendors. Consider the following capabilities to be added to OBAIC manager:

- Allow user to register an OBAIC adapter of specific AI platform
- Load an OBAIC adapter route corresponding request to the adapter
- Manage model catalogs exposed by different adapters

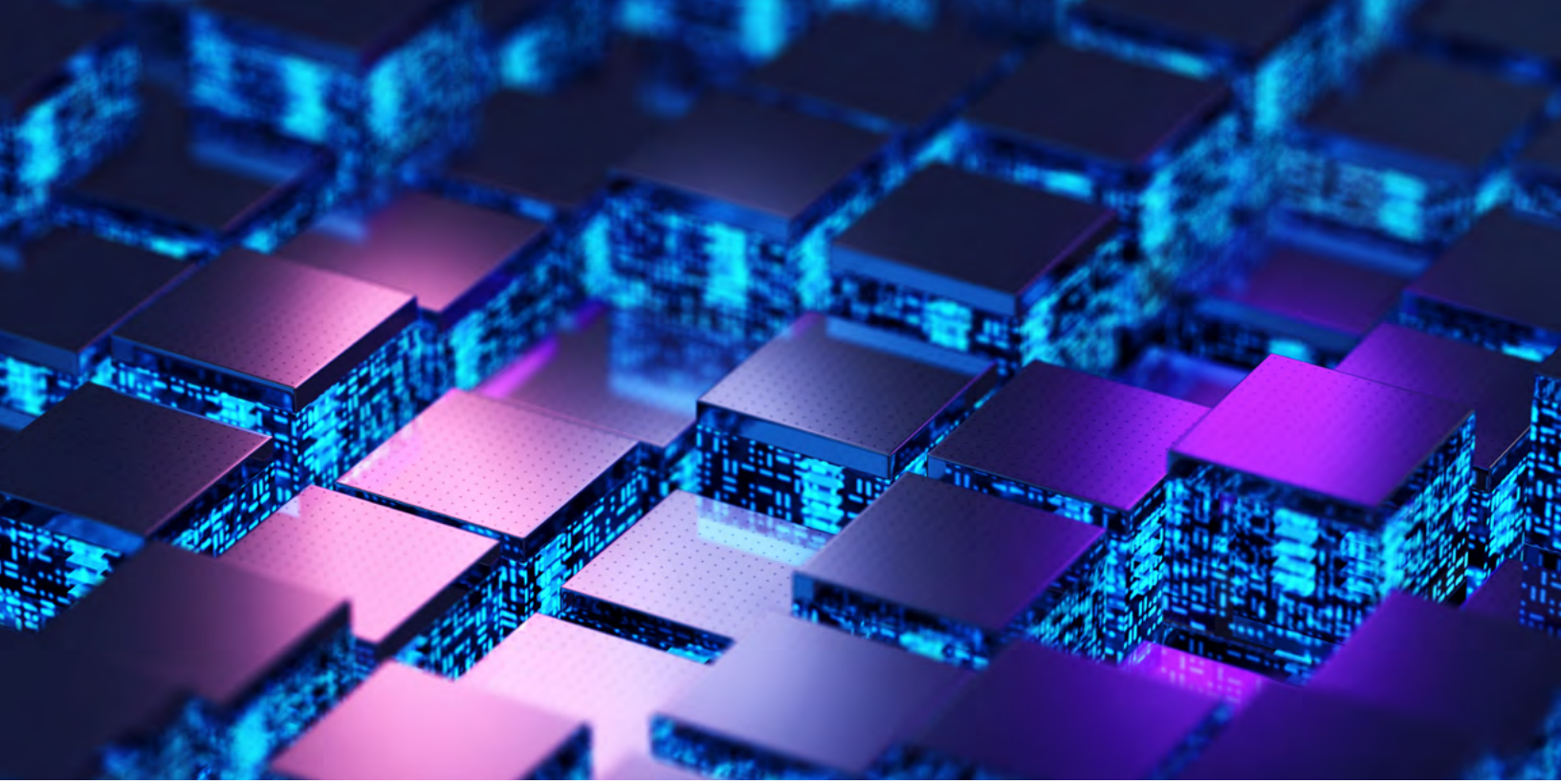
BI Vendors

BI vendors are the major drive for OBAIC. They understand and translate user's intent into queries for AI vendors. Although the majority of the inferencing work is done in an AI platform where the model resides, the BI vendor adopting OBAIC needs to plan a BI Client that integrates with OBAIC. Instead of coding against a specific AI platform, or ML frameworks, a BI Client primarily talks to OBAIC manager. The following capabilities will be planned when implementing the BI Client:

- Connect to one or more AI platforms through OBAIC manager
- Map BI users' requirements to filters and options of model exposed in OBAIC model catalog
- Pass data-in-memory or as reference using OBAIC API

- Fetch inference results through OBAIC adapter
- Query or accept status updates of AI model training job
- Interpret and visualize results of model training / inference fetched via OBAIC adapter

The BI vendor will plan to hook up the real-time inferencing results with their dashboards, if not already done so. BI vendor will also elaborate on how to organize and consume various models from the many catalogs exposed by AI platforms via the OBAIC adapter.



Conclusion and what's next

We hope this white paper has provided you with a pretty good idea of the **Where, Who, Why, When, How, What, and Which** questions about our proposed direction for AI, BI, and Data. However, it is just the beginning of an exciting journey. None of this can become a reality without a dedicated group of people contributing to the project by brainstorming requirements, discussing design, writing code, testing results... etc. Therefore, we hope you can join us to build the future of analytics together, please join our #bi-ai-committee channel on LF AI & Data Slack **https://lfaifoundation.slack.com**, or contact our chairperson Cupid Chan at **cchan@pistevodecision.com**.

Looking forward to working with more bright minds and revolutionizing Analytics together!

BI & AI Committee Members (alphabetical)

Cupid Chan, Pistevo Decision

Chairperson of BI & AI Committee in LF AI & Data. Cupid has been a consultant for years providing solutions to various Fortune 500 companies as well as the Public Sector. He is also a Senior Fellow and Adjunct Professor in University of Maryland College Park. In his latest venture, his company develops a decentralized healthcare data platform allowing AI Model to run by Federated Learning.

Deepak Karuppiah, MicroStrategy

Senior Architect with MicroStrategy's Gateways Group with over two decades of experience in software engineering and data analytics area. He is currently responsible for the connectivity layer of MicroStrategy Enterprise platform focusing on secure high performant data ingestion from a variety of data sources from relational databases to modern cloud and application sources. In a past life, he has published peer reviewed research in machine vision and machine learning and his recent work with BI & AI committee has brought him full circle to his previous interests. He supports a local non-profit with his technical expertise in his spare time.

Joe Madden, SAS

Sr. Product Manager within the Analytics and AI Product Strategy division at SAS. He joined SAS in February 2022 and his primary focus is enabling customers with open-source solutions and leading enhancements to Model Studio within the SAS Viya ecosystem. Joe received an MBA from Boston University's Questrom School of Business in 2017. Joe received a BS in Industrial & Systems Engineering and a Certificate in Computer Science from the University of Wisconsin-Madison in 2010.He resides in Cambridge, MA and spends his free time running along the Charles River

Dalton Ruer, Qlik

Dalton Ruer is a Data Scientist Storyteller and Analytics Evangelist. He is a seasoned author, speaker, blogger and YouTube video creator who is best known for dynamically sharing inconvenient truths and observations in a humorous manner. The passion which Dalton shares thru all mediums, moves and motivates others to action. Most recently he has become addicted to artificial intelligence generated artwork platforms

BI & AI Committee Members (alphabetical)

Yi Shao, IBM

An architect in IBM's Data and AI group. He is the tech lead for IBM SPSS Modeler, a visual data science and machine learning solution with low-code/no-code workbench. His primary job is to design and code features for modernizing IBM SPSS Modeler with trending open-source technologies from data, algorithm, and platform ends. He is passionate about making a better world with AI. He teaches AI to students from primary school to post-graduate. He is also an active inventor, hold 12 US patents in the field of data science at the time of this publication

Sachin Sinha, Microsoft

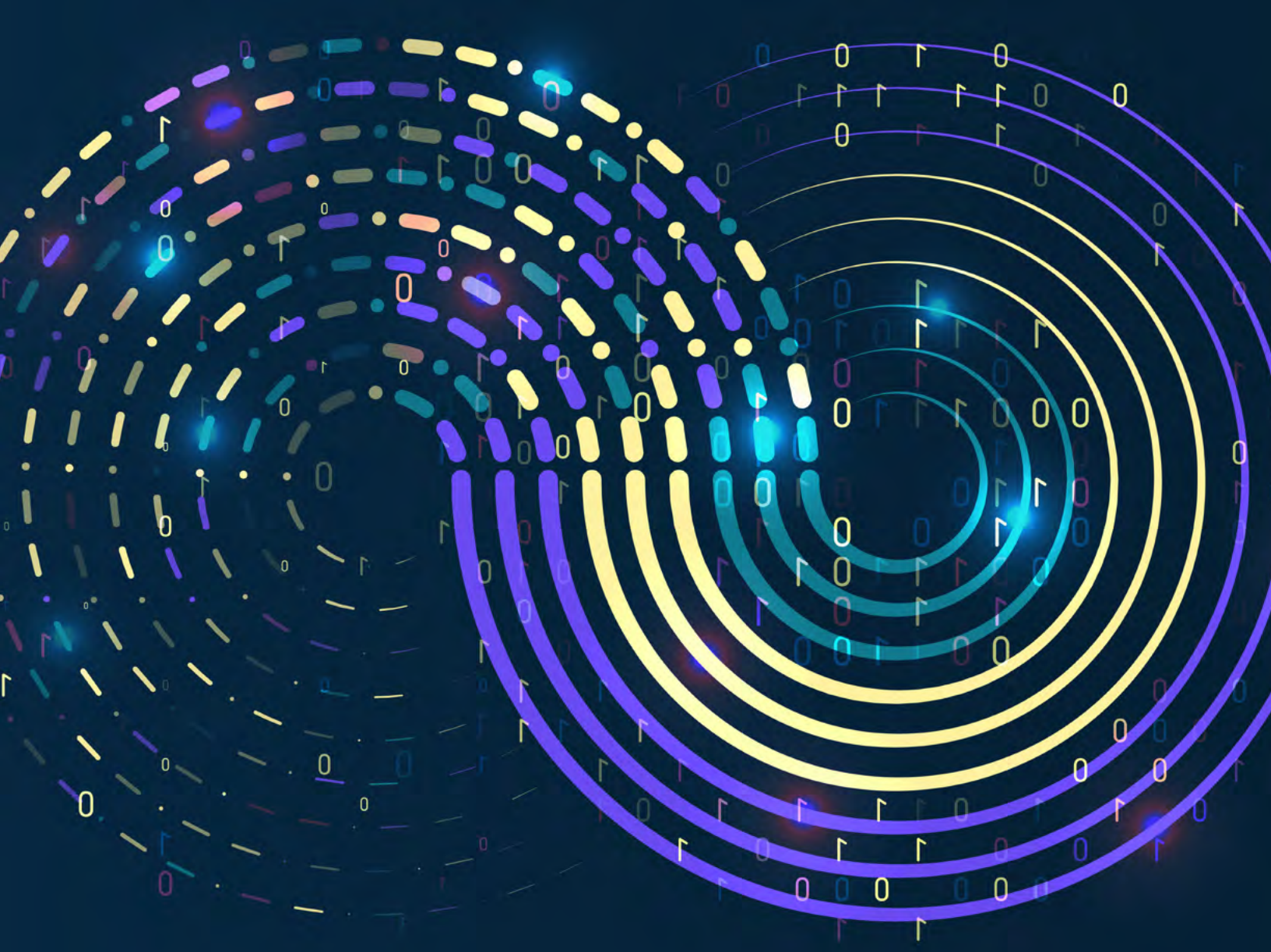
Director of Technology Strategy at Microsoft. After graduating from the University of Maryland, he continued his information management research as a data engineering leader and designed systems that helped his customers make decisions based on data. During this time, he helped startups in the healthcare space get off the ground by building a business on data and helped organizations in public sector achieve their mission by enabling them for decisions based on data. He lives in Fairfax, VA, with his wife and two sons, and remains a fervent supporter of the Terps and Ravens.

Stu Sztukowski, SAS

Product manager at SAS in the InsightOps Division driving approachable analytics through SAS Visual Analytics. He received his bachelor's in Statistics in 2012 from North Carolina State University and his master's in Advanced Analytics in 2013 from the Institute for Advanced Analytics. Prior to product management, Stu specialized in forecasting, statistical analysis and business intelligence with a vision to advance low-code/no-code high-performance analytics solutions that can be used and understood by all. He is a mentor and well-rounded leader with a passion for public speaking who helps make complex analytics friendlier for data scientists and business analysts.

Your Name, Anywhere around the World

This person is passionate about technology, especially in analytics. S/he joins this committee after reading a whitepaper about how AI, BI and Data can work together and start hanging out with a group of industry leaders sharing the same vision and passion. If you read the whole paper to this point, you know who you are. So, don't wait and contact us to join the force!



FOR MORE INFORMATION

https://lfaidata.foundation/projects/bi-ai

LF AI & DATA

OBAIC