THE CAPCO INSTITUTE
JOURNAL
OF FINANCIAL TRANSFORMATION

TECHNOLOGICAL

Overview of artificial intelligence deployment options

ALI HIRSA | SATYAN MALHOTRA

ARTIFICIAL INTELLIGENCE

#58 NOVEMBER 2023

THE CAPCO INSTITUTE

JOURNAL OF FINANCIAL TRANSFORMATION

RECIPIENT OF THE APEX AWARD FOR PUBLICATION EXCELLENCE

CONTENTS

TECHNOLOGICAL

OPERATIONAL

ORGANIZATIONAL

DEAR READER,

As the financial services industry continues to embrace transformation, advanced artificial intelligence models are already being utilized to drive superior customer experience, provide high-speed data analysis that generates meaningful insights, and to improve efficiency and cost-effectiveness.

Generative AI has made a significant early impact on the financial sector, and there is much more to come. The highly regulated nature of our industry, and the importance of data management mean that the huge potential of AI must be harnessed effectively – and safely. Solutions will need to address existing pain points – from knowledge management to software development and regulatory compliance – while also ensuring institutions can experiment and learn from GenAI.

This edition of the Capco Journal of Financial Transformation examines practical applications of AI across our industry, including banking and fintechs, asset management, investment advice, credit rating, software development and financial ecosystems. Contributions to this edition come from engineers, researchers, scientists, and business executives working at the leading edge of AI, as well as the subject matter experts here at Capco, who are developing innovative AI-powered solutions for our clients.

To realize the full benefits of artificial intelligence, business leaders need to have a robust AI governance model in place, that meets the needs of their organizations while mitigating the risks of new technology to trust, accuracy, fairness, inclusivity, and intellectual property. A new generation of software developers who place AI at the heart of their approach is also emerging. Both GenAI governance and these 'Developers 3.0' are examined in this edition.

This year Capco is celebrating its 25th anniversary, and our mission remains as clear today as a quarter century ago: to simplify complexity for our clients, leveraging disruptive thinking to deliver lasting change for our clients and their customers. By showcasing the very best industry expertise, independent thinking and strategic insight, our Journal is our commitment to bold transformation and looking beyond the status quo. I hope you find the latest edition to be timely and informative.

Thank you to all our contributors and readers.

Lance Levy, **Capco CEO**

OVERVIEW OF ARTIFICIAL INTELLIGENCE DEPLOYMENT OPTIONS

ALI HIRSA | Professor of Professional Practice, Department of Industrial Engineering and Operations Research, Columbia University, and Chief Scientific Officer, ASK2.AI

SATYAN MALHOTRA | Chief Executive Officer, ASK2.AI

ABSTRACT

Artificial intelligence is a very powerful application whose time has come. At a quick glance, it can be really seductive to believe, for example, the purveyors of xxxGPT, that its deployment is as simple as pushing a button or is a "data in, miracles out" strategy. However, harnessing it effectively requires navigating a myriad of options embedded within its critical pillars of data, models, and visuals. The complexity is accentuated by the deployer's capabilities and the organization's openness to change, as outcomes move from rules to an objective-based spectrum. In navigating these challenges lies the key to optimal deployment.

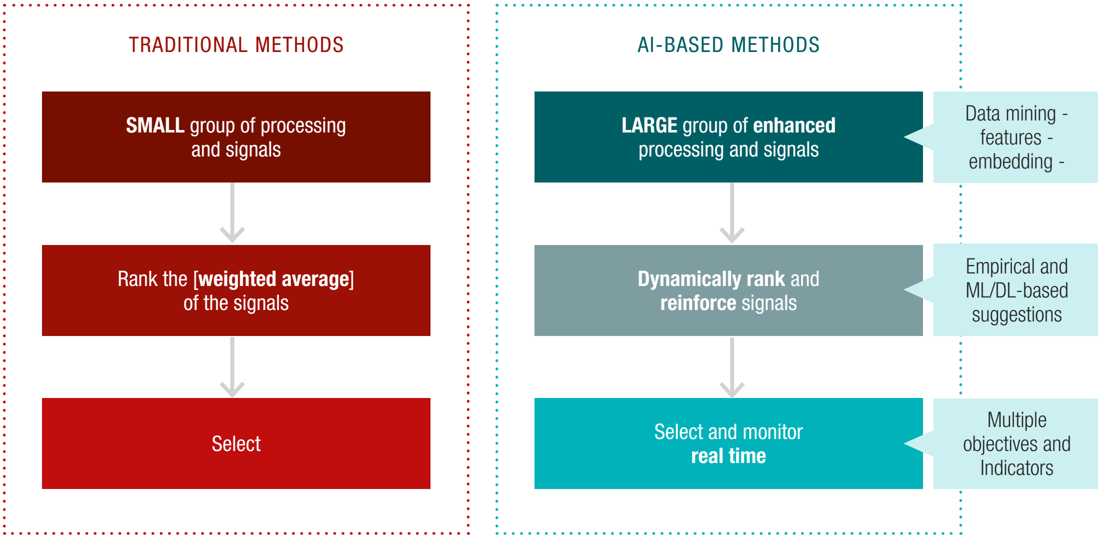
1. INTRODUCTION

In the realm of artificial intelligence (AI), some amazing things are being done by some amazing people, which is leading to some amazing results. Hopefully, this pacifies the shallow learning experts. Now for the deep(er) learning aspects of the current push of **AI everywhere and for all**. For the older engineers, most of the models being deployed (with some updates) have been there for many years, so what gives? Well, for one, we know that great strides in readily available computing power have been an excellent catalyst for the more ubiquitous push of AI. Another has been the ever-increasing money supply via the venture community and their ability to sell assets between themselves or to the next tier private or public community. This race to automate all things human is making engineering cool again with more jobs available than degrees being printed. Therein, establishing the current cycle by pointing the research engines at the opportunities at hand.

For some of us it has a hint of the Y2K[1] days when every boardroom, chatroom, money room, and classroom was pushing for solutions to the elusive double zero so the world would not come to a standstill. Since the world needs the engines running, it led to great billable hours for participants, spawning many startups, mixed solutions from developers, and money thrown at buzzwords. Pressure from time and money deployment compressed the processes for separating the grain from the chaff, which led to some not-so-pretty conclusions when the meals were finally consumed. History repeats itself, albeit maybe with some twists, tweaks, and turns. Similar compressed and expedited processes were seen during other such hyped times, including the dot com bubble. On the AI front, on February 9th, 2023, Google lost U.S.$100 billion dollars of its market value as the market punished it for its tardy Bard presentation. But was it tardy or is there an "AI catch up or lose" issue being exemplified? We found it interesting that some venture capitalists began pointing to the lack of practicing/checking the pitch before the presentation. Cannot see the forest for the trees or calculated censure given the exit plan relationship? The point remains that the pace with which all things AI are being pushed to consumers, users, funders, advisors, et al. seems to rekindle some memories for those of us who were at the table during the past frenzies. This time again, perhaps, maybe a little bit of overuse, abuse, and misuse of AI and its applications?

[1] For the younger readers, Y2K was the year 2000, when all computing was to come to an abrupt end due to the perceived poor programming of the elder generation. A lot of money was spent trying to fix the issue and we will never know if there were more issues or billable hours.

Figure 1: Approach comparison

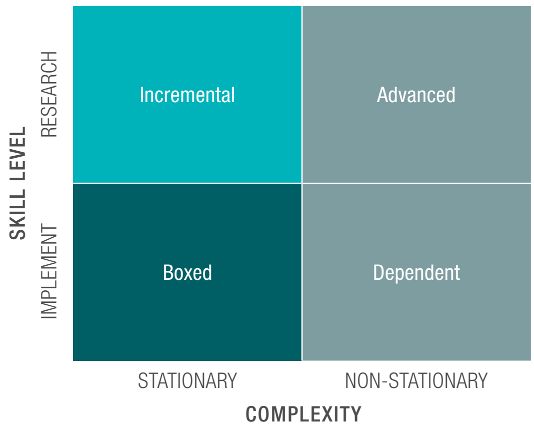


We are not saying that AI is not here to stay and note that the "rocket" train has left the station. Figure 1 visualizes the traditional versus AI-based deployments that allow parsing through numerous datasets, models, viewpoints, and visuals, etc., concurrently to continually assess historic and/or predicted performance within multiple aspects, subject to defined or suggested objective and evaluation functions.

The approach is very powerful and at a quick glance it can be really seductive to believe, for example, the purveyors of xxxGPT, that AI deployment is as simple as pushing a button or is a "data in, miracles out" strategy. Maybe true in the elusive future, but the current reality is that AI deployment has a lot of optionality and right choices need to be made to capture its immense potential. Being on the wrong frenzy driven side may entail wasted time and effort, as will be realized by some blindly following herds. The needs and level of intensity varies across use cases, regions, industries, etc. One way to think about this is in Figure 2, where for effective AI deployment we need to balance the expectations of the use case depending on the skill/experience level and the potential task complexity.

Within each of the quadrants, there is a lot of optionality and need for a lot of decisions. For the more complex cases, effective AI deployment is even more difficult and a lot of work still remains to be done. However, once any breakthrough happens then the trajectory of AI deployment engineering is relatively rapid, but we need those nudges and breakthroughs.

Figure 2: AI effort quadrants



For example, in the financial markets, given the current AI environment, one could consider using Natural Language Processing (NLP) techniques to read and file documents as somewhat of a boxed case, whereas, implementing NLP techniques to read real time data for advantageous financial trading so far remains an advanced case. Another way to look at this is that applying boxed cases in the non-stationary/research quadrant may be punitive and using advanced cases in the stationary/implement environment may not be value add. We will not get into the mathematical aspects, but where applicable, we will highlight limitations of assessed AI techniques as well as our proposed research nudges.[2]

[2] https://www.ask2.ai/research/

2. UNDERSTANDING OPTIONALITY

AI deployment is in a transitory stage and effective rollouts will depend on the knowledge bank of the deployers and decisions made by the users. As with Y2K, or the dot com bubble, or "then some", participants may burn money as AI is the new big idea whose time has come. Our suggestion is that if you have a seat at the table, it may be wise to look at the ingredients, chefs, and the dishes more closely. The more complicated dishes may need more than a naïve attempt from a cookbook and an independent mechanism for judgment. To make this discussion more pointed, we explore the AI deployment optionality around selecting a mutual fund manager. We assume those reading have some exposure to mutual funds and would ask you to parallel your current assessment processes. We look at the deployment optionality within each of AI's three pillars: data, models, and visuals.

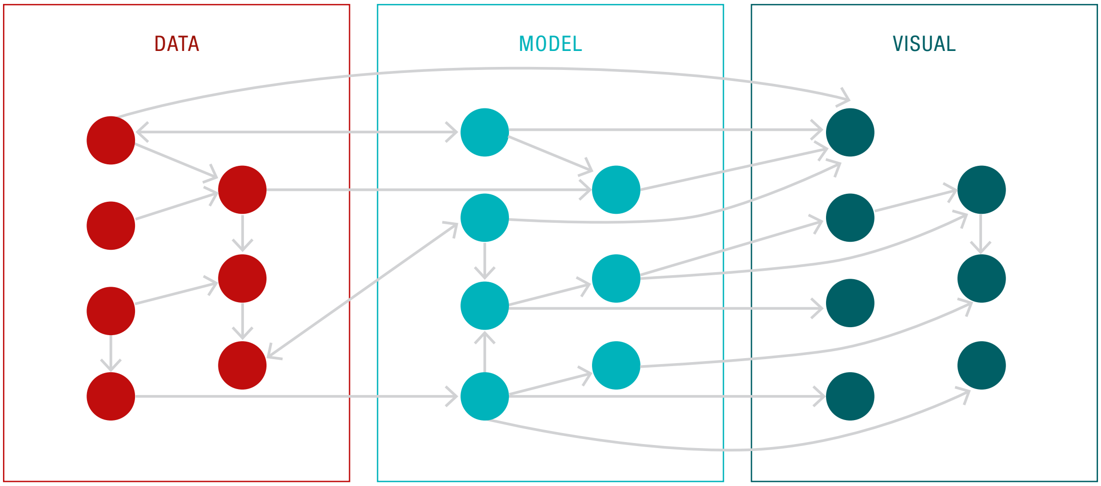
We stay away from the arguments around why these three or which pillar is more important. Some consider data to be the new oil, some modeling secrets to be the sauce, and some unique visual wrappers to be the trust builders. Additionally, judgment mechanisms have to be set that help evaluate the optionality within stability frameworks that allow evaluating the tracking error among other elements. Since this is done over time and on out-of-sample data, the judge is considered an independent unbiased framework for evaluating the results. As we will see, there is optionality there too as decisions need to be made on setting the appropriate objectives and associated evaluation criteria. This can be complicated, as judgments involve decisions on the related value system that supports the recommendations, selection, rewards, and penalties. The use cases, participants, judges, etc., are different and thus the deployment optionality needs to be understood for appropriate selection. This is because beyond considering all as vital pillars, each use case could have a very different path to "its" optimal solution. Finding that path or the tuning is possibly the key. In Figure 3, we had some fun illustrating some possible ways of connecting the dots.

Not surprisingly, there are numerous ways to connect the dots. Arguably, for boxed cases they may be established (or more or less specified for a use case), but until the advanced cases become boxed, the paths have to be tuned or are open to arbitrage. The arbitrage comes from the choice of faster deployment of any model/system (e.g., maybe untested for the use case) or cautious deployment of better models (e.g., tuned for the case). Depending on the use case the risk-rewards can be very different.

Furthermore, for binding the pillars, another layer comes into play, such as change management. As we know, without having the right people, entities, aspects, etc., be part of the AI deployment journey, or the right setup, there is a risk that all of the work may end up gathering dust somewhere. This is especially important as you begin to appreciate the level of optimality and decision points that needs to be addressed across AI deployment. Then, there are other related challenges of regulatory frameworks, local challenges, accessibility, computing power, etc., and many of them are transitionary.

Overall, for maximum effectiveness, it may be best to try not to compress the processes for separating the grain from the chaff. So, let us see what the cooking optionality entails.

Figure 3: Path choices



3. DATA

Data: the ingredients for the dish is a critical pillar. Simply put, although the xxxGPT purveyors may give an illusion, as per our assessment, models are not clairvoyants (yet). From a data perspective, it is as essential to have the data as well as to understand how to use the data. The current stage of AI deployment has put more focus on the "having part" with arguments spanning more is better, unique is better, etc. This is not unusual, as we are somewhat in the early stages of AI deployment and thus resource gathering is vital. However, the resources are not as readily, cleanly, or widely available. This opportunity has spawned providers from the software participants pitching streamlined storage or access capabilities, better processing capabilities, pure data providers pitching clean(er) or proprietary datasets, hybrids pitching better signal processing, and so on.

Until we reach the elusive utopian data stage, we find that data management straddles all the boxes of the AI effort quadrants. This becomes especially true as processes evolve from we "have" data to "how best to use" the data stage. Even within the "we have" data part, you will note that you need to be careful and know what you have/get as not everything is as simple as a pitched boxed case.

3.1 Nature of data

Data itself can be classified as "structured" or "unstructured". Structured data is tailored and generally stored in designated formats, while unstructured data is an amalgamation of different data types stored in their native formats. For example, your hard drive may be structured, but we can assume that various types of files are stored in the sub directories. This analogy can be extended to the task in hand and, as you will start to note below, the ability to manage both data types is generally a value add. Definitionally, processing unstructured data is more of a challenge and the key resides in effective and accurate extraction. Not surprisingly, a lot of effort is being expended in streamlining unstructured data so more and more can be part of boxed cases.

3.2 Types of data

The types of data sourced are topical. For our mutual fund example, we can source processed or unprocessed price data, holdings data, alternative data, news data, social data, proprietary data, and so on. Within each of these data types, there are various fields with varying frequency that all add to the data management complexity (e.g., multidimensional information can be tick level, minute, hourly, monthly, semiannual, etc.).

3.3 Storage of data

The traditional usage and familiarity are generally around "relational" databases, where tables are used to store data (think Excel). As relationships become more complex, "graph" databases may be better suited (think trees, branches, and leaves). Each branch or leaf can store various types of information, and since the types are somewhat grouped (e.g., within the branch or leaf) the number of connections is reduced versus a relational table where data is in a tabular form. For example, this can improve the response and management time associated with the queries. For our mutual fund example, the price and related information may be in a tabular form in the relational database, whereas connected information, such as alternative or social information, is in the graph database.

Data volume also has to be balanced with concerns around control and security, where fragmented data is harder to protect consistently. For example, the large data needs of LLM/GPT are understood but it is undecided whether to store data internally or use open-source solutions. The E.U.'s GDPR dictates data privacy norms and this puts an increased burden on data walls and mirrors, navigating global versus local datasets, inherent biases, etc. Basel regulatory pressures include making data auditable and reproducible for third parties. And so on.

3.4 Pre-processing data

The objective here is to have the data ready for analysis. The data can be sourced from a single or multiple sources, be in different formats, have different information, stored in a variety of ways and so on. For our mutual fund example, we also have to deal with multidimensionality and time series that are continually updated. To get the data ready for analysis, varying degrees of pre-processing may be required.

3.4.1 APPENDING AND CHECKING DATA

Most participants take the source data as a given. Unfortunately, there is usually no one true source of data. For the mutual fund example, data can be received from multiple sources, can have different identifier codes/symbols (e.g., CUSIP, ISIN, SEDOL, TICKER, etc.), and could be subject to very different taxonomies and protocols. It is imperative to know what you are working with, the rationale for the difference, where the pitfalls are, and so on. For example, a comparison of data for the same mutual fund from two reputable sources can show different (a) alternative data – expense ratio compositions, b) price data – total return on how they capture capital gains and dividend days, or c) holdings data – sector exposures. It is worth noting that this is for financial products, where the reporting is more or less structured – as in regulation heavy, legalized via prospectuses, and reported via electronic exchanges. Now imagine these pitfalls where the data is unstructured and all of it could be driving exposure, sentiments, signals, and so on.

3.4.2 FORMATTING AND CLEANING DATA

As the baseline data needs are set, additional steps are format cleaning and data standardization. Format cleaning requires streamlining the data, where some features may be stored as strings, could be nested, not have values, different frequencies, and so on. Once general formats are set, the dataset could further require "imputation" (e.g., filling the missing data with substituted values, where imputing time-series data should avoid data-leakage), NLP (e.g., aligning nomenclature that points to the same), and model specific engineering (e.g., standardizations). For example, filling missing data utilizes techniques from the simplest to the most sophisticated, including (a) forward autofill, (b) linear interpolation, (c) cubic spline interpolation, (d) cubic B-spline interpolation, (e) Brownian bridge, (f) variance gamma bridge, (g) Fourier transform techniques, and so on.

For the mutual fund example, the data is received at discrete points in time. We have to keep track of manager history across accounts rather than just continuity in the fund. When a manager leaves or joins another fund, the system has to account for the adjustment in expertise. Similarly for illiquid assets, the performance is self-reported as there are no central clearing systems. The challenge extends to managing incomplete data, incorrect data, reported biases, and so on.

3.4.3 TESTING DATA

We have to ensure that the datasets are robust enough to deal with AI models, e.g., raw or processed with cleaned values. Testing includes ensuring perturbations, different signal-to-noise ratios, adversarial attacks, and such, do not drastically distort results.

3.5 Processing data

The assumption at this point is that the data is clean and readily accessible for analysis. The objective of this part is then to make the data ready for modeling purposes.

3.5.1 DEALING WITH LARGE DATASETS

Generally, models are able to deal with more data better than with less data. At the same time, feeding similar data would lead to overfitting, auto-correlation, and other not so pleasant issues. Dimensionality reduction, such as "principal component analysis" (PCA), is one such method that can transform and reduce the number of measures or times so a single series can represent a set (without losing any information). However, if the datasets have time and multidimensionality aspects, then the standard PCA techniques may not give stable results. Here, we propose the robust rolling PCA (R2-PCA) that mitigates commonly found obstacles, including eigenvector sign flipping, temporal instability, and managing multiple dataset dimensions. If the objective is to identify some latent relationships or interrelationships among variables, then "factor analysis" (FA) can be the preferred method.

3.5.2 DEALING WITH SPARSE DATASETS

This can be a real challenge for AI models, as nothing can be done without data. However, if there is some level of data, then that can be augmented with synthetic data. Techniques such as Bayesian sampling and adversarial generative modeling can help create data that closely mimics existing datasets. Here, generative and hierarchical models are used to sustain statistical properties and stylized facts for different frequencies in both the time domain and frequency domain. These are high-risk areas, as care must be taken to ensure that the augmented datasets do not contribute to an alternative reality. We refer to our work on "temporal attention" and "temporal transformer generative adversarial networks" (TAGAN & TTGAN), where images inspired the original work and now the work is being extended to account for various datasets including financial products.

3.5.3 ASSESSING DATASETS

These techniques can augment the analysis and make for easier explainability when reduction or performance assessment techniques are applied across categories versus across the whole dataset.

- **Categorizing/classifying:** this is a simple form of grouping datasets where items can be bunched within predefined categories. This can be done using basic definitions or some manual structure of commonalities. In our mutual fund example, the industry has grouped funds using predefined classes, e.g., Large Cap, Small Cap, Fixed Income, and so on.
- **Segmenting:** this is a way to divide the data into parts or segments based on motivation. In our mutual fund example, it can be those funds that perform well during a regime.

- **Clustering:** this is a more advanced form of categorization/classification, where the groupings are based on similarities and data characteristics. For example, clustering mutual funds based on holding data exposure (equity, sector, etc.), performance measures, factor sensitivity, macro/market conditions, alternative data, etc. This can be done in a parallel or in a sequential manner. If datasets have time and multidimensionality aspects, standard "clustering" methods (K-Means, Hierarchical, etc.) may not give stable results. We propose CPK-Means and SIK-Means methods for producing stable and deterministic clusters over time.
- **Regimes:** this classifies the dataset into periods of similar behavior or events. The classification includes the defining characteristics of the regimes as well as the transition probabilities as movements between regimes. This can effectively assess the anticipated behaviors at similar points in time. For example, an advanced use case assesses how the fund clusters behave within and/or across regimes. Given the complexity of managing times series and multidimensionality, we propose an AI-based methodology for classifying regimes that produce stable financial regimes with transition probabilities.
- **Measuring performance:** these are constructed by manipulation of the same underlier in the form of a time series. For our mutual fund manager selection, the underlier is price, and the performance measures range from primitive to those requiring advanced financial engineering. A survey and a taxonomy of portfolio performance measures reveals that there are over a hundred such performance measures, and there is an assessment choice of time horizons (e.g., monthly, quarterly, annual, three years, five years, etc.) and roll/look back windows (e.g., daily, monthly, quarterly, etc.). Some of these performance measures are relative and thus need a designated benchmark. To quantify the manager's risk attitude, we propose an additional golf inspired "advisor assessment framework" with a scorecard, fairway average, and handicap.
- **Indexing/labeling:** this is a way of naming the grouped data to be easily referable. In our mutual fund example, Cluster A can be funds with high returns and Cluster B can be funds with low volatility. Note that since the data within the group takes on the implied meaning, this can lead to biased results and potentially amplify issues.

> *"Great experiences blend the exterior with calibrated power under the hood."*

Given the inherent probabilistic nature of AI models for making the suggestions, recommendations, selections, and so on, it should not be surprising that a large number of the data processing techniques are statistical in nature. The key is choosing the proper technique and understanding that many of the boxed solutions may not work for the learning-based models.

4. MODELS

Models: the equipment for the dish are a critical pillar. They encapsulate the analytical part of the task and objective. This is a complex part of the deployment process, yet a lot is taken for granted or assumed to work, potentially as black boxes. Depending on the task, model dependency or deployment can easily straddle all the boxes of the AI effort quadrants. Attempting to naively transplant models across use cases with differing nuances, datasets, temporal considerations, dimensionality, and so on, can be punitive (depending on the appetite for the error rate). There are many models to choose from and some are better for the task, some easier to comprehend, some easier to explain, some easier to implement, and some less computationally expensive, etc. We need to be able to choose the "right" models and have mechanisms to know when they are working and when they are not working.

4.1 Setting the framework

The model deployment framework consists of setting the objective, measuring the results, evaluating the results, accepting or rejecting the results (or the penalty-reward functions for the more advanced models), and refining the models – then repeating the loop. In actuality, this is done rapidly and concurrently by running numerous models under various scenarios, parameters, assumptions, targets, etc. All are obviously subject to the deployer's knowledge, data depth, and available computational power.

4.1.1 OBJECTIVE FUNCTION

The objective acts as the desired result for the model. This target cannot be abstract and has to be set as a quantifiable objective function. For example, if the results of the target are to drive a decision, then the target can be one or many steps towards "suggesting" the optimal decision. Consequently, the objectives can be interim or final, near or longer term, whole or components, sequential or nonsequential, single or multiple, and so on. Additionally, a subtle difference between an objective and a constraint is worth noting, where constraints are guard rails that drive the model towards the target.

4.1.2 MEASURING RESULTS

This entails assessing if the objective or the target has been met. The selection, ranking, or recommendation is usually based on the results closest to the target. The more precisely the objective function is defined, the easier it is to measure the results. It is also essential to assess details around the results, for example, which models were performed, what the error rate was, under what circumstances or scenarios it was met, what features drove them, were there any outliers, and so on, as all this comes into play via the refinement loop.

4.1.3 EVALUATING RESULTS

This entails accepting or rejecting the results. If this includes potentially rewarding or penalizing the results, then it also allows for setting the degrees of reward-penalty functions.

- **Back-testing:** this technique involves splitting the dataset into a training set and a test set. The model is trained on the training set and then evaluated on the unseen test set to assess its generalization performance. This evaluation helps determine how well the model performs on new, unseen data. This can also include various types of "scenario analysis", "stress testing", and "simulations".
- **Validation set and early stopping:** in cases where models have hyperparameters to be tuned, a validation set is often used. It is separate from the training and test sets and is used to evaluate different hyperparameter configurations. Early stopping is a technique that monitors the model's performance on the validation set and stops training when performance deteriorates, thus preventing overfitting.
- **Robustness testing:** this involves testing the trained model over various different data with known and similar characteristics to see how the trained model behaves. This could include having different degrees of noise, perturbations, and adversarial attacks.
- **Deployment:** once an AI model is deployed in real time applications, ongoing monitoring and evaluation are essential. This involves tracking the model's performance, detecting anomalies or drifts, and ensuring it continues to perform as expected. This also serves as input for the refinement loop.

4.2 Selecting the models

There are many types of models, including simplistic ones, complex ones, and those that auto choose between models. One more characteristic has to do with the representation of data or input, where if the data is presented in multiple levels and a different model is used at each level and gets combined for final decision making, then the models are hierarchical.

4.2.1 RULE-BASED MODELS

These are the simplest form of models that operate based on predefined rules. They follow "if-then" logic to make decisions that are essentially fixed equations to represent relationships between inputs and outputs. These models are straightforward to implement and suitable for simple problems but are less effective for complex tasks and have limited flexibility and adaptability. A mutual fund selection example would be to select a fund if the total return is more than a certain percentage.

4.2.2 REGRESSION-BASED MODELS

These models are suitable where there is a need to identify some form of a relationship between the inputs and outputs.

- **Linear regression:** these utilize a linear equation to model the relationship between input features and the target variable.
- **Lasso regression:** these perform feature selection and regularization by adding an L1 penalty term (the sum of the absolute values) to the loss function.
- **Ridge regression:** this incorporates an L2 penalty term (the square root of the sum of the squared values) into the loss function, encouraging smaller coefficient values.
- **Elastic net:** these combine L1 and L2 penalties, offering a balance between feature selection and regularization.

Regression-based models offer advantages such as interpretability and flexibility and allow for the assessment of the relative importance of different input features in determining the outcome. Regression models provide a statistical framework for inference and hypothesis testing and explicitly define assumptions about the relationships between variables, which helps guide the modeling process. Regularization techniques, such as ridge regression or lasso regression, can be applied to mitigate issues like multicollinearity or overfitting. They are computationally inexpensive and provide a baseline for comparing the performance of other complex models. They are also widely understood and used in various fields, making them accessible to researchers and practitioners, e.g., credit scoring, demand forecasting, econometrics, marketing analytics, risk assessment, and in general predictive analytics. A mutual fund selection example would be to select a fund based on the regression coefficients of measures, where it can be expected that the coefficients would adjust for the changing performance.

4.2.3 BAYESIAN MODELS

These, also known as belief networks or probabilistic graphical models, are graphical representations of probabilistic relationships between variables. They use directed acyclic graphs to depict dependencies and conditional probabilities. Bayesian networks are used for reasoning under uncertainty, probabilistic inference, and decision making.

- **Bayesian networks:** these extend traditional linear regression by incorporating prior distributions over the regression coefficients. It provides a probabilistic framework to estimate the uncertainty associated with the regression parameters and make predictions.
- **Bayesian linear regression:** these extend traditional linear regression by incorporating prior distributions over the regression coefficients. It provides a probabilistic framework to estimate the uncertainty associated with the regression parameters and make predictions.
- **Gaussian processes:** these are flexible probabilistic models that define a distribution over functions. They can be used for regression, classification, and uncertainty estimation. Gaussian processes capture prior assumptions about the smoothness and correlations in the data.
- **Variational autoencoders (VAEs):** these combine the concepts of autoencoders and Bayesian inference. They use deep neural networks to learn a low-dimensional representation of the data and model the underlying distribution in a probabilistic manner. VAEs enable the generation of new samples and provide uncertainty estimates.
- **Bayesian neural networks (BNNs):** these integrate Bayesian inference with neural networks. They assign probability distributions to the network weights, allowing for uncertainty estimation and more robust predictions. BNNs can be trained using techniques like variational inference or Markov Chain Monte Carlo (MCMC) sampling.
- **Sequential Monte Carlo methods:** these are also known as particle filters, Bayesian-based models used for state estimation and tracking in dynamic systems. They represent the probability distribution using a set of particles and update the distribution as new observations arrive.
- **Bayesian reinforcement learning:** these combine reinforcement learning techniques with Bayesian inference. It allows for incorporating prior knowledge about the environment and policies, enables uncertainty estimation, and provides a principled approach to exploration-exploitation trade offs.

Bayesian-based models offer advantages such as the ability to handle uncertainty, incorporate prior knowledge, update beliefs with new evidence, and provide probabilistic interpretations. They find applications in various domains, including natural language processing, computer vision, and decision making under uncertainty. A mutual fund selection example would be to select a fund based on sector preference by examining sector rotations and their impact on holdings in mutual funds. This could help to pick funds that are resilient to some macro shocks.

4.2.4 MACHINE LEARNING-BASED MODELS

Machine learning-based models can learn complex patterns and relationships in the data that cannot be captured by linear regression models. These models can handle non-linear relationships between variables and adapt to complex decision boundaries, and can automatically learn relevant features from raw data, reducing the need for manual feature engineering. They are designed to handle large datasets with high computational efficiency and can scale well. They often employ ensemble methods, such as random forests or gradient boosting, to combine multiple models and improve overall performance. Their strength is in automatically selecting relevant features and identifying the most informative variables for the task at hand.

- **Decision trees:** these are hierarchical structures that recursively split the data based on input features to make predictions. They are easy to interpret and widely used in various applications.
- **Random forests:** this is an ensemble learning method that constructs multiple decision trees to make predictions.
- **XGBoost:** this is a "gradient" boosting algorithm that combines weak learners into a strong predictive model.
- **Support vector machines (SVM):** these are supervised learning models that classify data by finding optimal hyperplanes.
- **(e) Naive Bayes:** these are probability-based classifiers assuming independence between features given the class.
- **(f) Gaussian mixture:** these are probabilistic models that assume the data is generated by a mixture of Gaussian distributions. They are often used for clustering and density estimation.
- **(g) Hidden Markov:** these are statistical models that can capture temporal dependencies in sequential data. They are commonly used in speech recognition, natural language processing, and bioinformatics.
- **(h) Logistic regression:** this is a statistical machine-learning model used for binary classification problems. It estimates the probability of a binary outcome based on input features using a logistic function.
- **Principal component analysis (PCA):** this is a dimensionality reduction technique that identifies the most important features or patterns in data. It transforms the data into a lower-dimensional space while retaining as much information as possible.

Machine learning techniques encompass unsupervised and semi-supervised learning approaches, which can discover patterns and structures in the data without relying on explicit labels. These methods can be valuable for exploratory analysis, clustering, anomaly detection, and identifying hidden patterns. They are often designed to adapt and learn from new data, allowing them to handle changing environments. A mutual fund selection example would be utilizing a supervised machine learning algorithm like logistic regression to classify mutual funds. For example, finding the probability that a group of mutual funds with good historical performance would continue to have good future performance.

4.2.5 DEEP LEARNING-BASED MODELS

These models have a human-like ability to learn based on non-linear and more complex relationships embedded in the data. Deep neural networks can automatically learn hierarchical representations of the data. They consist of multiple layers of interconnected nodes (neurons) that learn increasingly complex features at each layer. They can model non-linear relationships and capture complex patterns in the data. They can scale effectively to large datasets and are designed to handle big data scenarios, and can benefit from parallel computing on GPUs or distributed systems. This scalability allows for training models on vast amounts of data, which can improve performance and generalization. They often benefit from transfer learning, where models trained on large datasets or related tasks can be utilized as a starting point for new tasks. For example, pre-trained models, such as those trained on ImageNet for image recognition, offer a head start by leveraging prior knowledge and learned representations, reducing the need for extensive training on new datasets. They can extract relevant features from raw data automatically. Through multiple layers of abstraction, they learn representations that are useful for the given task. This feature extraction and representation learning make deep learning models effective in tasks such as image classification, speech recognition, and NLP. The disadvantages of "deep learning models" are that interpretability is challenging, and in general, they are data hungry, which means they require much more data for learning and to avoid overfitting.

4.2.5.1 Early generation models

Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are particularly effective in capturing spatial and temporal patterns in these domains. CNNs are designed to capture spatial patterns and structures, while RNNs are effective in modeling sequential or time-series data. This makes them well-suited for tasks like object recognition, sentiment analysis, speech recognition, and machine translation. They can learn directly from raw input to output without relying on manual feature engineering or intermediate representations. This end-to-end learning simplifies the modeling pipeline and reduces the need for domain-specific knowledge and handcrafted features. They have demonstrated state-of-the-art performance on various tasks, surpassing traditional machine-learning approaches. They can achieve higher accuracy and better generalization, especially when trained on large-scale datasets.

- **Feedforward neural networks (FNNs):** these are inspired by the structure of the human brain; FNNs consist of interconnected nodes (neurons) organized into layers. They are usually used for reverse engineering or one-to-one mapping.
- **Convolutional neural networks (CNNs):** these employ convolutional layers to extract features from input data and are used for image recognition tasks.
- **Autoencoders (AEs):** these models are used for embedding and dimensionality reduction.
- **Recurrent neural networks (RNNs):** these are suitable for sequential data; RNNs utilize recurrent connections to capture temporal dependencies. Long short-term memory (LSTM) and GRU (gated recurrent unit) are also variations of recurrent neural networks that were introduced to help with vanishing gradients to avoid premature optimization.
- **Transformer models:** this is a type of architecture used for various tasks, especially natural language processing, due to their attention mechanism's ability to handle long-range dependencies effectively (particularly effective for natural language processing tasks, transformer models leverage attention mechanisms for sequence modeling).

4.2.5.2 New generation models

Deep learning models benefit from ongoing research and advancements in the field. With the growing popularity of deep learning, new architectures, regularization techniques, optimization algorithms, and network designs continue to emerge, pushing the boundaries of what is possible. Recent advances in deep learning models are to work on human languages.

- **Generative models:** even though they may classify as part of deep learning models, we set them under their own due to their architecture and training. These models aim to generate new data instances that resemble the training data. Examples include: "generative adversarial networks" (GANs), which is a type of autoencoder that learns a probabilistic representation of data, enabling the generation of new samples; and "variational autoencoders" (VAEs), which is a type of autoencoder that learns a probabilistic representation of data, enabling the generation of new samples.
- **Transformer learning models:** these models leverage knowledge learned from one task to improve performance on a different but related task. Pretrained models like BERT for NLP or ImageNet-pretrained CNNs are common examples. Other examples include: "large language models" (LLMs), which are recent advances in deep learning models to work on human languages (the transformer architecture is the fundamental building block of all LLMs); and "generative pre-training transformer" (GPT), which is a language model that is pre-trained on sample data (tokens) to understand and then create language results, for example, for sentiment analysis.

In mutual fund selection, the methods and models often struggle to capture the complex patterns and stylized facts, potentially leading to suboptimal decisions. Generative adversarial models (GANs) or variational autoencoders (VAEs) can generate synthetic data that closely mimics the characteristics of real mutual fund data for better and more robust training of models.

4.2.6 HYBRID MODELS

Hybrid models refer to the combination of two or more different AI techniques or algorithms to create a single, more powerful, and effective model. For instance, reinforcement learning with deep neural networks (deep reinforcement learning) has been used in various applications. These models leverage the strengths of each individual technique, compensating for their weaknesses and improving overall performance. Hybrid models are often used to solve complex problems that may be challenging for a single AI approach to handle on its own. They can help make more informed decisions by combining different types of data, models, or strategies.

In mutual fund selection, they can (a) combine structured data (e.g., financial statements, price history) with unstructured data to gain a more comprehensive view of the fund's potential. By integrating various data sources, the model can identify patterns and relationships that individual models might miss and (b) help evaluate risks by combining traditional statistical models with machine learning algorithms. The statistical models may provide a solid foundation for risk estimation, while machine learning models can add the capability to analyze complex patterns and market dynamics, (c) use NLP techniques to analyze sentiment from news articles, social media, and financial reports, then combine the sentiment scores with other financial indicators to make more informed investment decisions, and (d) be designed to learn and adapt over time by combining reinforcement learning with other algorithms to continuously improve their decision-making abilities as market conditions change.

4.3 Training the models

Models need to base their decisions on some form of prior behavior that is set as an objective, and those suggestions can be accepted or rejected (including rewards or penalties for the more advanced models) at evaluation. Training techniques center around how to make the models learn the logic for making the suggestions. Feature engineering to extract meaningful patterns or relationships from raw data, which can help the model better understand the underlying patterns that can assure accurate classifications or predictions. This involves baseline training, testing (on out of sample data), and validation (during training to make sure there is no overfitting).

4.3.1 SUPERVISED TRAINING

The models learn from labeled training data to make predictions or classifications. They are provided with input-output pairs during training and aim to generalize patterns in the data to make accurate predictions on new, unseen data. Models learn to make predictions by minimizing the discrepancy between predicted and true labels. As such, labeled data means that for any input, the corresponding output is called a label, where input features are paired with corresponding target labels.

4.3.2 UNSUPERVISED LEARNING

The models learn by finding patterns and relationships in unlabeled data. They do not have explicit target labels during training with an aim to discover patterns, structures, or representations without explicit target labels. Unsupervised techniques often perform tasks like clustering, anomaly detection, dimensionality reduction, and generative modeling.

4.3.3 SEMI-SUPERVISED LEARNING

The models learn by a combination of supervised and unsupervised learning, where the model is trained on a small amount of labeled data and a larger amount of unlabeled data.

4.3.4 REINFORCEMENT LEARNING

The models learn by utilizing an agent interacting with an environment, learning optimal actions through trial and error. They receive feedback in the form of rewards or penalties for their actions and aim to maximize the cumulative reward over time. Reinforcement signals (rewards) guide the agent toward desired behavior.

4.4 Tuning the models

Training a model refers to the process of feeding labeled data into a model and adjusting its (internal) parameters so that it can learn to make accurate classifications or predictions on new (unseen) data. Tuning a model is the process of optimizing the hyperparameters of the trained model to improve its performance. These are parameters that are not learned during training but affect the learning process and the model parameters that result from it.

4.4.1 HYPERPARAMETER SEARCH

Hyperparameter search plays a vital role in fine-tuning machine learning models in order to do optimal performance. Grid search, random search, and Bayesian optimization are three common methods used for this purpose, each offering unique advantages. The choice of the hyperparameter search method depends on the complexity of the model, available computational resources, and the size of the hyperparameter space. By selecting the most suitable hyperparameters, machine learning models can deliver more accurate and reliable predictions for a wide range of real-world applications.

4.4.2 HYPERPARAMETER TUNING

Hyperparameter tuning involves optimizing model performance by fine-tuning hyperparameters, such as learning rate, regularization strength, batch size, and more. Unlike model parameters that are learned during training, hyperparameters are set before training and could significantly influence how the model learns and generalizes from data. Proper hyperparameter tuning is essential for achieving optimal model performance and preventing issues like overfitting or underfitting. By systematically adjusting these hyperparameters, learning models can better adapt to complex datasets and deliver more accurate and reliable predictions.

4.4.3 ONLINE LEARNING

Models are trained incrementally on streaming data, adapting to new information in real time. Particularly useful when data arrives sequentially or when computational resources are limited. Instead of waiting to accumulate a large batch of data and then retraining the model periodically, the online learning approach processes the data as it arrives. This approach enhances the ability to optimize outcomes that need real time assessments.

4.4.4 AUTOML (AUTOMATED MACHINE LEARNING)

Automates the process of model selection (architecture), hyperparameter tuning, and feature engineering. Reduces the need for manual intervention, making AI more accessible to non-experts. For example, AutoML can automatically generate features (e.g., technical indicators, fundamental analysis metrics) that are relevant to predicting the performance of the mutual funds. By doing this, the system can sift through numerous holdings, analyzing different features, and can identify patterns and relationships that humans might overlook.

4.5 Assessing the models

Assessment techniques for AI models involve evaluating their performance, accuracy, and generalization capabilities. These assessment techniques provide insights into an AI model's performance, help identify areas for improvement, and ensure its suitability for the intended task or application. The choice of assessment techniques depends on the specific problem, type of model, and available data.

4.5.1 ACCURACY AND LOSS METRICS

These metrics measure the model's performance on a specific task. For classification problems, metrics like accuracy, precision, recall, and F1-score are used. For regression problems, metrics such as mean squared error (MSE) or mean absolute error (MAE) are commonly used. These metrics provide quantitative measures of how well the model is performing.

4.5.2 CROSS-VALIDATION

Cross-validation is a technique used to assess a model's performance by splitting the dataset into multiple subsets or folds. The model is trained and evaluated on different subsets, allowing for a more robust evaluation of its performance. Common cross-validation methods include k-fold cross-validation and stratified cross-validation.

4.5.3 CONFUSION MATRIX

A confusion matrix is a table that summarizes the performance of a classification model by displaying the counts of true positive, true negative, false positive, and false negative predictions. It provides insights into the model's ability to correctly classify different classes and identify errors or misclassifications.

4.5.4 Receiver operating characteristic (ROC) curve and area under the curve (AUC)

These techniques are primarily used for binary classification problems. The ROC curve plots the true positive rate against the false positive rate at various classification thresholds. The AUC represents the area under the ROC curve and provides a measure of the model's ability to distinguish between classes.

4.5.5 PRECISION-RECALL CURVE

The precision-recall curve illustrates the trade off between precision (the proportion of true positives among predicted positives) and recall (the proportion of true positives identified). It is particularly useful when dealing with imbalanced datasets or problems where one class is of greater interest than the other.

Overall, simply saying there is "AI" or transplanting solutions may not work as they have to be selected, tuned, trained, and refined for the tasks. To put this in perspective of our mutual fund selection example, assume we want to select five mutual funds (from a choice of thousands) given an objective (and evaluation criteria) of generating three-year excess return over the S&P500. Much like the multiple mutual funds, depending on the deployer's knowledge bank, there can be multiple models capable of dealing with multidimensional and temporal financial market datasets. These could include simplistic ones based on a single measure, a fixed equation, regression based, machine learning based, deep learning based, AutoML models, and so on. Furthermore, the circumstances themselves need to be modeled, including the interplay of measures, regimes, events, signals, sentiments, factors, etc. Another set of models could be for back-testing, where a model stability framework needs to be set up to continually assess if the chosen model is behaving the way it is supposed to and the triggers to note if/when the model is misbehaving and what to do (hopefully as a leading indicator). Simulation models can give color on the behavior of the selection under different scenarios. All this is in a continual loop of selection, evaluation of the variance from the objective, and refinement. It should also be noted that as we go beyond selection, other sets of models come into play, such as for asset allocation, portfolio management, risk management, asset planning, and so on.

On the change management side, this also highlights the need for robust "model risk management" (MRM) frameworks, especially for high-risk decisions, including:

- model change policies addressing periodic recalibration, data acquisition, algorithm decision overrides, dataset shifts, and replacement criteria
- using multiple shadow AI models, as recommended by regulators, to challenge and monitor the performance of the primary model
- establishing validation and audit standards.

5. VISUALS

Visuals: the appeal/taste of the dish remains a critical pillar. We believe visuals serve an important purpose in helping build trust around the analyses (however simple or complex). Without going on a psychology tangent, let us assume that human acceptance of results requires some degree of comfort around what the recommendation is for, when it is being made, and why it is being made. From a human-and-human perspective, this resides in the form of trust built around direct or implied relationships, needs, experiences, and so on. If we were to assume that human-and-AI interaction is also loosely based on a similar setup, then there needs to be a similar trust system. AI deployments attempt to build that trust by a) being accretive to some expectations (e.g., reduce time/effort, be profitable, etc.), and b) presenting them, at least initially, in a humanly digestible way (e.g., numerical and graphical representations that are appropriate, pertinent, experiential, etc.). This engagement is likely the key to accepted deployment, and as they say, a picture should speak a thousand words, or, in this case, become the face of the computational engines. We will not delve into the myriad of visual/presentation choices; simply put, if the visuals are not meaningful, intuitive, and easily explainable, then no matter how good the results, they may not be "useful" and will possibly be put in a drawer somewhere. In our opinion, for AI development to be trusted, it needs to be able to clearly represent the "what, why, and when" in a transparent and simple manner.

5.1 The what

The telos or the purpose of the AI application deployment. Holistically and locally, what is the purpose of the deployment? Is it accuracy, personalization, removing biases, eliminating emotions, supplementing information, expanding knowledge, automation, scaling analysis, remote or distant delivery, increasing solution points, increasing speed, reducing costs, increasing profitability, removing blind spots, identifying embedded relationships, recognizing patterns, detecting anomalies, faster execution, etc.? Yes, the choices and objectives can be very diverse and multiple, but they need to be articulated, understood, and set. The what, or the objective, is the key and is managed via accept-reject (or with reinforcement methods leveraging penalties and rewards) decisions in the training of the models. Herein, unless the telos or the overall objective is agreed to clearly, it may be a difficult deployment as the AI decision systems can be geared towards very different answers. AI deployment allows users to move from rule-based to decision-based ecosystems, but we note that these decision-based systems reside somewhat within rule-based ecosystems as critical decisions on objectives and judgments are arguably disguised rules with levels of granularity. And these need to be set.

In our mutual fund example, the objective can be to maximize excess return, and the evaluation can be to have a high Sharpe ratio. If the objective is to have a high return and low volatility, then you can set one as the primary objective (high return) and the other as a constraint (lower volatility), or use the "explainability index" approach to accommodate both. From a visuals perspective, setting the objective allows for easier visual representation for deployment, with numerical or graphical representations ranging from simplistic two-dimensional ones, such as line charts, to complex multidimensional ones, such as heatmaps, bar charts, stacked area charts, bar plots, parallel coordinate plots, etc. All with the motivation of providing data points to instill confidence and trust.

5.2 The why

This can be viewed as supporting representations for the decisions being "suggested". Any form of a model (whether complex or naive) can be a black box, depending on the user's sophistication. To build trust, we need to know why the decision is being made. However, given the accept-reject frameworks (or the penalty-reward functions for the more advanced models) embedded in the AI designs, it is easier to know if/when the outcomes are reliable than to know why (or how) the models are making them. And, as expected, increasing model complexity makes it exponentially more difficult to identify decision rationales. As a result, a lot of focus is on back-testing and simulations to test the models, but "the why" remains possibly the least structured part of the AI deployment processes. With increasing AI deployment, some methods are being suggested that try to explain the model's workings, including "partial dependence plots" (PDP), "permutation importance", "global surrogate models", and "anchors". Here we discuss the more common ones.

5.2.1 INTERPRETABILITY

This is to understand the relationship between elements in terms of the cause and effect (e.g., inputs and outputs); the drivers within the relationship for understanding the causality.

- **LIME (local interpretable model-agnostic explanations):** a technique for explaining individual predictions of black-box models. Generates locally interpretable explanations by perturbing input data and observing the impact on model predictions. Visualization tools can display the explanations, such as highlighting important regions in an image or showing word importance in text data.
- **SHAP (SHapley Additive exPLanation):** a method to attribute the contribution of each feature to the prediction outcome based on cooperative game theory.

5.2.2 EXPLAINABILITY

Understanding what is implied by the elements (e.g., inputs or outputs) in terms of what they all represent as part or whole. This facilitates data comprehension.

- **Attributions/contributions:** visualizing the attribution or contribution of input features to the model's predictions.
- **Feature visualization:** technique that is used to understand what features or patterns in the input data activate specific neurons in an artificial neural network.
- **Explainability index (EI) and "risk of target" (RoT):** technique that explicitly balances hundreds of input categories of performance measures according to default or specified preferences for a composite bounded score between 0 and 1 for each and the aggregate of the measures. RoT leverages the EI for comparing individual performance against benchmarks (as targets). The composite and component analyses explain the drivers of divergence of the target/objective (as a point-in-time, trend, or relative assessment). In the mutual fund example, this can also be used for managing multiple objectives and for reinforcements.

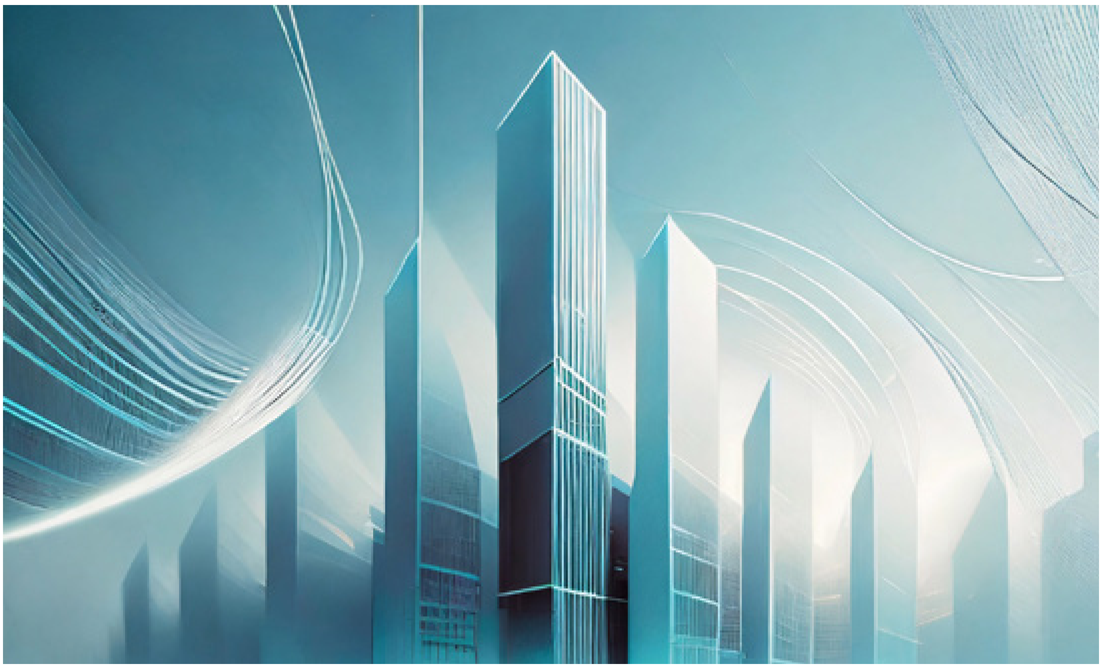


Image generated by Adobe Firefly

Regulatory concerns have added impetus to explainability and interpretability research. For example, under the E.U. GDPR, consumers are entitled to explanations for algorithm-driven decisions, a right not explicitly confirmed in the U.S. While AI may excel in credit scoring statistically, few E.U. banks seemed to have sought licenses for AI in internal credit evaluations due to regulatory concerns. In contrast, unregulated credit rating agencies heavily rely on AI.

5.3 The when

This can be viewed as the representations for the time period(s) being assessed. They can be absolute or relative and assess results as point-in-time or trends.

5.3.1 Historical

Visualizing past data and trends to gain insights into historical patterns and relationships. Time series plots, line charts, and heat maps are commonly used for visualizing historical data.

5.3.2 Prediction

Visualizing model predictions to understand patterns, trends, and potential future outcomes. Scatter plots, bar charts, and interactive visualizations are used to represent prediction results.

5.3.3 Scenario analysis

Creating visual representations of hypothetical scenarios to explore the potential impact of different variables or events. Helps in decision making, risk assessment, and planning by visualizing various outcomes.

5.3.4 Simulation

Visualizing simulations of complex systems or processes. Allowing users to observe and analyze the behavior of the simulated results. Graphs, animations, and 3D visualizations are common techniques used in simulation visualization.

5.3.5 Back-testing

Visualizing the results during discrete points in time on out of sample or historic datasets. Helps in visualizing results during similar periods.

6. CONCLUSION

Our aim with the paper is to give the reader an appreciation of the multitude of ways to connect the dots, choices within use case deployments, possible variations in results, need for localized knowledge, dangers in oversimplification, need for cross sectional expertise, and so on. For the boxed cases, we may be more comfortable in pushing the proverbial deployment button (e.g., via a xxxGPT), but as the risks associated with the decisions increase, the deployment need and analysis may move across the quadrants, where understanding the nuances becomes critical in enabling optimum outcomes. As you read AI publications, you will note that the AI deployment itself is no different. The preference is somewhat in the eye of the beholder and pitched around the deployer's knowledge (including searchable methods) that is influenced by their backgrounds and agenda, e.g., economist, mathematician, philosopher, politician, etc. For example, economists tend to lean on the cost or value wrappers, and philosophers on the choice wrappers.

One way to think of the deployment optionality spectrum is as a range from acceptable imperfection (i.e., with lower accuracy, higher error tolerance, low efficiency per training data, weak models, weak infrastructure, etc.) to assumed perfection (i.e., with generally reduced choices with hyper personalization). We note that a) perfection itself is transitory, as most methods are based on available knowledge banks that are rapidly evolving, and b) current AI deployments can largely only handle accept-reject functions (with degrees of reward-penalty functions for more advanced ones); they are weak in managing the grayer human aspects such as implied meaning, emotions, evolving expectations, intentions, gut, valuing collateral damage, etc. The question becomes how is the telos (or, for that matter, your thinking) placed on the spectrum? Going to our food analogy, just because we know certain food types are not good, do you entirely stop eating them? Do you only go to the very "best" restaurants? How much of the freedom of choice can you give up? How quickly do you cede control to the "suggestions"? These decisions are easier for some tasks than others. As you frame answers to these choices, you start forming your deployment spectrum placement and path.

Since AI deployment is technical, the question regarding whether you need supplemental expertise and from whom arises. Experts are putting stakes in the ground with publications and packaged solutions. Incumbents with legacy infrastructure and capital investments have varying degrees of inertia and appetite for discovery. New entrants' nimbleness allows for speedy delivery but generally comes with a higher focus on beauty and experience, so the cut/paste of models becomes risky as the results can be very questionable. Either way, not everyone can engage in the advanced quadrants, as that requires knowledge, time, and capital. Herein is our word of caution, the race to AI everywhere that is now being accentuated with the xxxGPT claimants across verticals has dangerous elements, especially when combined with the traditional tech industry mindset of accepted risk of failure in getting the minimal viable product out. Maybe herd decisions will make some use cases subject to self-fulfilling prophecies, but where the risks associated with deployment are high you need to be cautious.

In navigating these elements lies the key to mitigating adverse surprises akin to the Y2K and the then some money burning adventures. We believe effective AI deployment lies in the knowledge intersection of subject matter, computer science, data science, and machine learning expertise. Advanced users understand the importance of what is under the hood and casual users base the usage on trust, which is earned. Either way, we find that meaningful AI deployments demand more than a simplistic "data in, miracles out" strategy. They require meticulous tuning, enhancements, and occasionally rethinking approaches. As such, great experiences blend the exterior with calibrated power under the hood.

ABOUT CAPCO

Capco, a Wipro company, is a global technology and management consultancy focused in the financial services industry. Capco operates at the intersection of business and technology by combining innovative thinking with unrivalled industry knowledge to fast-track digital initiatives for banking and payments, capital markets, wealth and asset management, insurance, and the energy sector. Capco's cutting-edge ingenuity is brought to life through its award-winning Be Yourself At Work culture and diverse talent.

To learn more, visit www.capco.com or follow us on Facebook, YouTube, LinkedIn and Instagram.

WORLDWIDE OFFICES

APAC
Bangalore – Electronic City
Bangalore – Sarjapur Road
Bangkok
Chennai
Dubai
Gurgaon
Hong Kong
Hyderabad
Kuala Lumpur
Mumbai
Pune
Singapore

EUROPE
Berlin
Bratislava
Brussels
Dusseldorf
Edinburgh
Frankfurt
Geneva
London
Milan
Munich
Paris
Vienna
Warsaw
Zurich

NORTH AMERICA
Charlotte
Chicago
Dallas
Hartford
Houston
New York
Orlando
Toronto
Washington, DC

SOUTH AMERICA
Alphaville
São Paulo

THE COVER IMAGE WAS CREATED USING JASPER AI, AN AI ART GENERATOR

FSC
www.fsc.org
MIX
Paper from responsible sources
FSC® C004309

WWW.CAPCO.COM

CAPCO 25
a wipro company