Artificial Intelligence (AI) for Business

A clear look into AI and its potential for business use.

DYMENG
TECHNOLOGY SOLUTIONS

Publish Date: February 20, 2024
Version: 1.0

Author: Jack D. Leach
Editor: Danielle Smyth
Review & Collaboration: Kyle Richard,
Nicusor Serban, Hernan Marano,
Natalia Hortas, Alexandru Cioarec,
Brian Demick

Introduction to AI

Of course we've all heard the term AI. As ubiquitous a household term as "the cloud" was a decade ago, we'd have to be quite out of touch to not realize the wave we're in. But with all the hype, buzzwords and such a broad scope of discussion available, what is it really? And more importantly, what does it mean for your business?

This document aims to give business decision-makers a clear look into AI and its potential for business use. While there's any amount of material available to cover the various topics of AI, including all its buzzwords, sub-disciplines and widely varying personal perspectives, assembling this information into a practical understanding of how AI could be applied to your business operations is not as straightforward as most would wish.

In this document, we'll look at the broad scope of AI, sufficient to arm you with the insight needed to make well-defined, pragmatic decisions regarding the use of AI technology and how it may be incorporated into modern business use. Topics covered will include:

- What it is, including a brief history. This will set a timeline for past, present and future to help capture the full scope of AI and what it can mean for you.
- The service landscape: from the mundane grammar suggestions in document editors to attempts to unravel secrets of the universe with the Large Hadron Collider; what is what, what are the in-betweens and where do the practical services land for businesses?
- Various capabilities and capability categories such as pattern detection, language processing, predictive analysis and generative AI, how they work in general and what their differences and ideal use cases are.
- A top-level view of Machine Learning and how it fits into the scheme of things, including business and development processes suitable for understanding how to estimate effort
- A quick review of ethics and legislation and some practices that can help us stay on the right path while attempting to future-proof against forthcoming legislation
- Some considerations for calculating total cost of ownership, return on investment and some risk mitigation strategies
- And finally we'll cover things to be wary of and some general strategic considerations for AI adoption as a business: how to ride yet another technology wave while positioning for risk mitigation.

A last point of note before diving in: while this document is based on the heavy research and real-world experience of our AI team at Dymeng, it is also not guaranteed to be perfect, especially given the fast-paced changes of any disruptive technology like Artificial Intelligence. While the aim of this document is to be relatively agnostic to specific implementations and as "timeless" as we can reasonably make it, periodic updates to this document reflecting industry changes will be posted. In addition, reader feedback would be greatly appreciated and taken into consideration for future revisions.

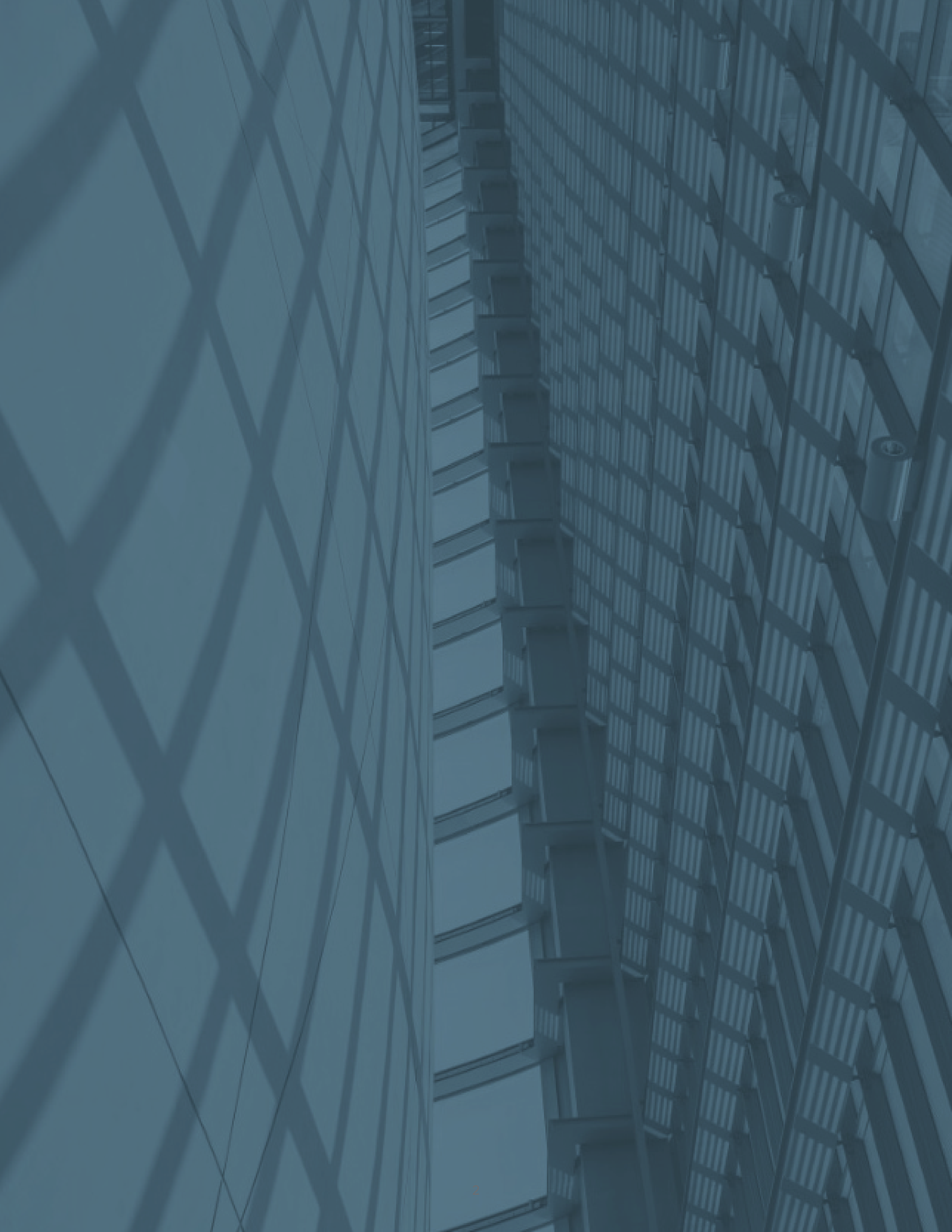


Table of Contents

AI – What Is It, Really? 5

A Brief History 6

What can it mean for you? 7

AI Service Landscape 8

Consumer AI Services 9

Citizen Developer AI Services 10

Pre-Trained AI Services for Developers 10

Ground-Up\Fully Custom AI 11

AI Capabilities 12

Pattern Recognition 12

Beyond Trend Analysis 12

Techniques & Approaches 13

Business Process 14

Predictive Analysis 14

Natural Language Processing 16

Generative AI (Text, Image, Video…) 17

General Usage & Capabilities 18

Data Models & Processing 19

Proprietary Data & GenAI 19

What About Machine Learning? 21

What Is Machine Learning? 21

Why Humans.. Isn't this "Machine" Learning? 23

Effort & Processing 23

Processing Power 23

Data and Pipelines 24

People 24

Pre-Trained vs. From Scratch Models 26

Deep Learning, Neural Networks (and others)? 26

Ethics & Legislation 27

Ethical AI 28

Legislative: a Rough Summary 28

AI Bill of Rights (USA) 28

Act (EU) 28

Costs & Risks 30

Calculating Cost 30

Time to Obsolescence 31

Mitigating Risks....31
A Healthy Dose of Pragmatism....33
No Silver Bullet....33
Watch, Watch, Watch....33
You Must Be This Tall To Ride… 33
Careful with your Data....34
Data Governance....34
Service Outages....34
Algorithmic Inertia....34
Providers Everywhere....34
Appendix A: How Much Of This Whitepaper Did AI Write?....35

AI – What Is It, Really?

Let's instead start with what it's not: AI is not intelligent robots with brains to overthrow their human masters and take over the world. Nor is it (on the opposite end of the spectrum) the computer you play a game of cards against on your phone while in a waiting room. Our definition, for general business purposes, falls somewhere in between.

IBM's one-liner definition is a good place to start: *"Artificial intelligence leverages computers and machines to mimic the problem-solving and decision-making capabilities of the human mind."* Good place to start or not, we do feel this definition is a bit over-arching in terms of practical business capability.

Let's take a slightly less abstract approach: essentially, AI is just another set of advanced algorithms, methodologies and practices. In fact, there's even a lot of overlap with common solutions not generally considered to be AI. For example, an advanced scheduling module taking into account many factors and having complex logic to generate a schedule might have been called "an advanced scheduling module" two decades ago, but labeled as AI today. Still, backed by vast data models, access to substantial amounts of computing power, highly advanced algorithms and a hotspot of innovation, today's AI certainly has capabilities that were unreachable decades ago. Ultimately though, with all its resources and capabilities, AI is still just a highly advanced version of the "if…then" conditional that all programming is built on. As another example, let's jump ahead a bit and consider a Machine Learning algorithm (which are the "brains" of many AI applications): an ML algorithm requires a set of inputs and provides an output. Reverse course by 100 years and consider a programmatic calculator function: it requires some sort of input, and provides some sort of output (e.g., ?add(2, 2); =4).

Now, there's probably any number of people heavily invested in the field that would turn purple and shoot steam from their ears to hear us liken AI and ML down to century-old if/then conditionals and a function that can add two numbers together. On the other hand, it's not exactly a false statement, either: AI and ML happen at vast scale, with vast data and computing resources, and are capable of vast calculations, but for all that are still only calculators.

We feel that this is a very important concept to keep in mind as it helps scope capabilities and expectations. At Dymeng, we've always said "we can do anything you want, if the budget's big enough," and it's true in AI as well: there may be preconceived notions of amazing things that AI can do for the business, and likely that's entirely possible, yet at the end of the day it still needs to be told what to do and how to do it. It's important to realize that there may be considerable effort involved in getting an AI system to do what you want.

Before wrapping up here, it's worth mentioning the concept of Weak vs Strong AI. Weak AI is AI that solves a specific problem. It's been designed to do one thing, and can do that one thing very well. All practical use cases of AI are built on Weak AI. Strong AI is a form of theoretical AI where such a system would have knowledge and intelligence to rival or surpass human thinking; that is, a machine with self-consciousness and learning capabilities, perhaps even some sort of sentient ambition. In other words, Weak AI is the logic in your waiting-room card game and advanced targeted algorithms, and Strong AI are those robots that will take over humankind. Luckily for us humans, Strong AI is – so far – entirely theoretical.

It's important to realize that there may be considerable effort involved in getting an AI system to do what you want.

A Brief History

We'll make this one quick, but we do feel that a quick overview of when AI "started" and what it's evolved to today (including the various spikes and plateaus along the decades) gives a good context for realistic expectations.

We should be aware that Artificial Intelligence has been in existence for nearly 70 years. In 1956 the conference Dartmouth Summer Research Project on Artificial Intelligence (DSRPAI) was hosted and the label and idea of Artificial Intelligence was born. Since this time, the field of AI has gone through numerous ups and downs based on general interest, research funding and technical obstacles and limitations.

Through the next twenty years (to the mid-70s), AI research was in full swing. Problem solving and language translations were some early goals of AI systems with keen interest by the U.S. government. This period brought very high hopes for AI systems, but also uncovered numerous obstacles, particularly with the required versus available computing resources at the time. These high hopes and promises of AI transformations were unable to be met due to resource limitations at the time, and interest (and funding) dwindled.

In the 1980s, the idea of AI came back on an uptick with the introduction of Deep Learning algorithms (machines that could be programmed to teach itself to do a task more efficiently) and the generation of Expert Systems (a type of early AI decision making technology). Still though, expectations at large were unmet and again the heavy government interest and funding faded.

While government and corporate interest in AI wasn't comparatively strong in the 1990s and 2000s, AI did make some headway in general. In particular, Moore's Law (the estimation that computing power and storage resources double each year) had paved the way to reduce many of the early obstacles with resource requirements, and some public interest was gained through events such as IBM's Deep Blue AI computer claiming victory over world chess champion Gary Kasparov.

Currently, Moore's Law seems to be slowing down (although the probable and likely forthcoming rise of quantum computing could change this landscape immensely), but we are globally collecting more data than ever before and have the processing power to do some amazing things with that data. While we may not yet be able to simulate reactions at an atomic level efficiently enough to cure diseases

through simulation instead of trial and error, the collective advent of recent algorithms, considerable computing resources and large collections of data have provided a platform for the next wave of AI capabilities.

What can it mean for you?

Keeping in mind two primary aspects covered so far, that a) AI is essentially advanced conditional decision making, and b) modern computing resources and the collection of big data as a standard for companies in general, we can start to imagine what the next wave of AI might mean for organizations today.

While we aren't likely to be designing robots to take over the world using "Strong AI" as previously defined, we can plausibly take the general approach of: *"I have (or can get) a lot of data, and I can run that through highly advanced decision making algorithms, with some capability for those algorithms to incorporate efficiency and quality adjustments on their own over time, and therefore I can…"*.

This is a good starting point for an AI think tank in modern times and helps set the practical limits scope for what we can achieve. However, it does leave us fairly open-ended. While a true visionary can make use of this open slate, some visibility on probable paths could certainly help. Unfortunately, the answer isn't always clear, but in the remainder of this document we'll attempt to shed light on various aspects of the technology to provide a full scope view of what's what and what can be.

AI Service Landscape

In the way of AI services, we have some extremes: from playing a game of cards against an “AI computer” on your smartphone to systems that can teach themselves to better analyze patient history and identify disease patterns and risk. Another spread of extremes might be from built-in AI in word processors for helping with your grammar to teams of data scientists building incredible AI-driven solutions simulating the molecular behavior of cancer cells for cure research. Given such a wide scope of applications, how does an organization go about making an informed decision on what approaches might be needed to offer a solution to their own business problems?

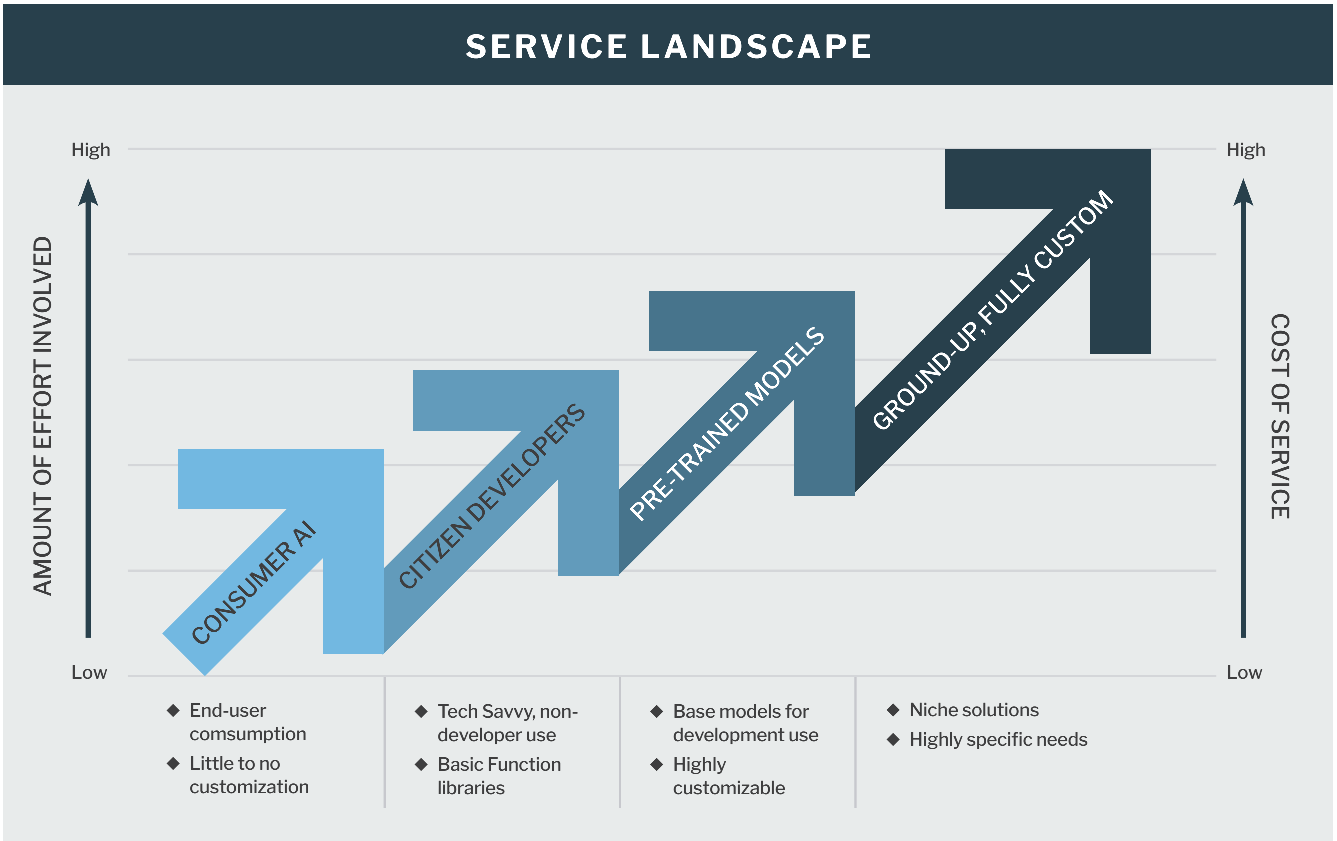
This section will help identify and categorize various levels of AI services available to us. With this understanding we should be able to recognize some rough scope of what types of solutions can be created and how much effort may be required to do so.

AI Services can generally be broken down into four primary categories:

- **Consumer AI Services:** Typically packaged into software for direct end-user consumption. Think built-in MS Word or Google Docs analysis and suggestions
- **Citizen Developer AI Services:** not directly utilized by end users, but nicely pre-packaged for inclusion by citizen developers. Think a pre-built library of functionality and data that can be drag-and-dropped into low-code app builders
- **Pre-Trained AI Services for Developers:** Powerhouse providers like Azure and AWS offer very capable, base-level AI models that development teams can utilize. Pre-trained and\or shared data models have base level functionality provide a myriad of potential solutions without having to build from scratch
- **From-Scratch AI:** Much like traditional software development, for those with highly specific needs where pre-made tools aren’t a good fit or complete solution, building from the ground up or a custom hybrid approach may be the way to go.

What is a Citizen Developer? An employee that isn’t a developer per se, but can use some basic low-code\ no-code tooling to come up with quick solutions for themselves or others.

From the top of the list to the bottom, these service levels range from the least flexible and easiest to use to the most involved. As we progress through the list, each becomes much more powerful and customizable, but also requires much more effort to see results. These could be causally defined as “brainless”, “fairly painless”, “now things are getting interesting” and “this is some serious stuff!” There’s a very, very wide scope here that could be measured monetarily as singular dollars on one end to multi-million or more at the other.



As with most things, this isn't perfectly clear cut; there may be some overlaps or some perspectives that don't quite fit exactly in one category or another. Others, like the popular ChatGPT, can feasibly fit in any or all of the categories defined, depending on how it's utilized. The key takeaway here is that we can set some sort of scale for effort involved, and upon considering a particular use case or idea, have some inkling of what kind of effort we're looking at.

Consumer AI Services

As consumers of various technologies, AI is prevalent in our day to day tasks, often in subtle ways that we don't realize. Suggestions in document editing services and scheduling assistants built into office productivity suites are good examples here. Or, perhaps, we sign up for additional services that are clearly AI-driven to help us perform our work tasks. Tools like Code Whisperer (Amazon) and Copilot (Microsoft, Github) can help mid and junior level developers write code more efficiently. In either case, these are end-user consumable tools and features backed by AI that require very little – if any – configuration, setup and internal development to utilize.

Consumer-level AI services are just that: relatively simple tools and functionality provided for end-user consumption by other companies that have done the heavy lifting. Because of their highly targeted, narrowly scoped and positioning for ease of use though, they're not very flexible and cannot generally be customized to any significant degree beyond their base use case.

While these services can offer advantages to end-users, they're not what we often consider to be "true" AI solutions in that they're typically classed as lighter weight helper tools rather than AI that solves some critical business problem. A collection of SaaS companies have popped up over the past few years offering dedicated services such as logo generation, resume parsing and others, but these are dedicated to end-user servicing rather than "buildable" AI solutions.

Citizen Developer AI Services

Citizen Developers can have access to more significant AI services than those often found at the consumer level. A citizen developer may for example use Microsoft's Power Platform to build a Power App connected to a data flow that uses a pre-made, platform solution for text translations. In this case, Microsoft would provide the platform tools and AI services for translations, and the citizen developer would "wire it up and make it work" without having to actually build the AI-driven service itself. These are generally plug-and-play services that require very little configuration or logical programming.

These types of AI services can be very powerful in that they're relatively easy to use as a building block for Rapid Application Development scenarios. The services provided are not tied to a particular end-user use case and tend to act more as a library of potential functionality for app building. This means there's far more flexibility in what can be built with them, and the heavy players such as Microsoft are the ones doing the heavy lifting on the AI end of things. Still though, they are limited to specific functionality; the upside here is that we have much more control over that functionality than we would in consumer-level AI services, the downside being that they can't do anything more than what they were specifically designed for.

Another significant advantage to having citizen developers build out relatively simple solutions for their departmental use is that these often act as a working prototype for more enterprise-grade solutions. A development team may take the functionality built by a citizen developer and further improve and stabilize it, wrap it in disaster recovery and security layers, and extract some AI (and other) functionality into something even more customized. Often, citizen developers can be the catalyst for tightly integrated software services, and the ability to put a library of pre-built AI services in their toolset opens up many long term benefits for the organization.

Some example use cases of these types of solutions may include implementation of chatbots, advanced search capabilities, support services, translation, category classification, business card readers, etc. Internal-facing applications tend to find a sweet spot with these types of services as they're generally quick and easy to set up, but often not customizable enough or hardened enough to serve as public-facing components.

Pre-Trained AI Services for Developers

Consumer and Citizen Developer AI services are all well and good, but pre-trained, publicly available models are where things really get interesting and the true power of customizable AI starts to be realized. Consumer AI services put predefined, basic AI-driven functionality at the fingertips of users, and citizen developers have access to libraries of AI and AI-driven functionalities; enter the next level of AI services offered by major companies like AWS and Azure which expose core data models and AI/ML functionality at a much more base level, allowing development teams to utilize advanced AI algorithms.

Some examples of these services are listed below. Notice how there's often overlap with prior service tiers, depending on how they're utilized. For example virtual agents and advanced search functionality could be utilized as Citizen Developer services with a highly "railed" experience (little customization allowed), or as part of a baseline trained model that is capable of far more advanced logic.

- Image and video recognition to identify objects, people, text, scenes and activities
- Extracting and analyzing text from large stores of documents
- Converting text to AI speech and voice
- Visual scanning to identify missing components, structure damage and irregularities

- Virtual agents, support, chatbots
- Extracting health data from unstructured medical text
- Fraud detection patterns analysis
- Image, text and content generation

These providers have put a lot of effort into building data models and algorithms to address baseline needs across various industries. They provide their baseline services to business developers with a more or less open slate, meaning the developers can fully customize workflows using the AI services as a base functionality. Providers make few assumptions about how we might utilize these services beyond that they're sometimes built to serve an industry domain (e.g., medical chart interpretation). Essentially, this strips away all of the "niceties" of consumer and Citizen Developer services, leaving powerful bare bones services that would take considerably more effort to build from scratch.

These services often serve a sweet spot for business development efforts in that they're much more powerful than other services discussed thus far, highly customizable, fit a number of advanced use cases and are considerably easier to adopt than building from scratch. Also, consider that many of these services are built based on continuous algorithmic improvements with the resources of some of the world's largest companies backing them. A significant advantage of using these pretrained services is that they are pretrained, and have years of effort devoted to them and are continuously improving.

For businesses wanting to "truly" grasp the advantages of powerful AI in scenarios customizable to specific business models, pretrained AI services offer the customization flexibility that can't be found at consumer and citizen developer service levels, without necessarily having to engage large teams of data scientists and massive amounts of research and development to begin seeing results. The general corollary to this though is that it is in a large part custom development, and considerably more development time is required than that of utilizing an off-the-shelf citizen developer AI library function.

Ground-Up\Fully Custom AI

Consumer AI is a given. Citizen Developer services allow you to get some basic AI functionality into light workflows. Pretrained models unlock a plethora of possibilities for utilizing AI in custom developed solutions. Still though, our problem may require a solution that doesn't fall in the capabilities of those services. Niche use cases and highly innovative organizations may need something more.

An ambitious undertaking for many, but the ultimate in leveraging the power of AI within the organization, building solutions from scratch (or almost from scratch) can be a large undertaking requiring vast amounts of domain understanding, technical resources and data science personnel. Programming your own advanced AI algorithms, preparing your data models and training them can provide seemingly infinite amounts of functionality, but comes at high resource effort and price points.

Often, a middle ground can be found by utilizing some existing, pre-training models and standard algorithms, coupled with your own specific data sets and common practices, approaches and tooling. Utilizing the pre-made where possible and augmenting it to suit, or building from scratch and supplementing with pre-existing functionality, has proven to be a sound strategy in decades of technology adoption, and there's no reason the same can't be given consideration here.

To fully explore this topic with the depth and coverage it deserves would take far more than what's applicable within the scope of this document; the idea here is to bring general awareness to the fact that this can be done. As we continue into the next section looking at core capability categories of AI (and a subsequent section on how Machine Learning (ML) fits into the large scope of AI), consider that any or all of those capabilities and practices can be applied to a fully custom AI build.

AI Capabilities

Looking at the service landscape of AI is one perspective into how AI services can be utilized, but as with most things, there's many perspectives that could help shape a full picture. In this section, we'll take another approach and categorize AI aspects by their technical capability.

While each of the following are generally considered to be primary types of practices or techniques for implementing AI, keep in mind that the majority of AI solutions are in fact a collection of these various practices working together to solve a given problem. Much like a piece of traditional software may consist of a front end web application, underlying backend API and a database, a given AI solution may consist of layers of pattern recognition, predictive analysis and language processing.

It's also worth noting that the major capability categories we'll cover are generally stacked. Meaning, Pattern Recognition tends to form the basis for the next section, Predictive Analysis. Both of those are then used heavily in the inner workings of Language Processing, which itself is a prerequisite to many types of Generative AI.

Within each of these categories we start to brush the surface of more technical algorithmic considerations. For the purpose of this document, we'll only touch base on technical algorithms very lightly as for now it's more important to understand the lay of the land rather than specifics on how the various practices perform their tasks.

Between this Capabilities section and the prior Service Landscape section, when done we should be able to consider any given AI use case or solution and roughly determine which categories it fits. If we can do that, a large portion of this document's purpose has been served.

Pattern Recognition

Pattern recognition algorithms, as you might guess, can analyze data to pick up patterns and regularities. These algorithms can work against structured data such as that found in relational databases, or unstructured data such as what we might have collected in a data lake. Pattern recognition isn't specific to traditional, tabular or document-like data though: image and video sources, GIS data or pretty much any other type of data that you may have available can be processed. Image sharpening tools, for example, work by analyzing patterns of color layouts to form the baseline of how to sharpen it.

Pattern recognition is a key concept that sets the groundwork for many other types of AI solutions. The ability to identify patterns is critically important to many aspects of business, even outside of the AI realm. When considering potential solutions, pattern recognition should be a cornerstone practice to build around.

Beyond Trend Analysis

An important consideration in pattern recognition is that it extends well beyond the idea of picking up on trend information. While the idea of looking at traditional historic data for emerging patterns in "normal" business data is all well and good, pattern recognition in AI has multiple other uses that are fundamentally important to further AI practices:

- **Object recognition:** pattern recognition forms the basis for a computer's ability to determine objects, either in whole or partially. For example, identifying that an image or video contains people or vehicles is based on pattern recognition, and even determining body pose positions (an important step in identifying a higher level scene of an image, for example) is based on pattern recognition.
- **Textual patterns:** the ability to recognize patterns through both small and large volumes of text has many important implications within AI
- **Video recognition:** motion recognition, intrusion detection and object tracking are all forms of pattern recognition with big implications for potential solutions

While the above list are often the common ones, others such as handwriting, biometrics, voice\ speaker detection, emotion detection and many more are all examples of pattern recognition.

The profound fact of pattern recognition is that almost all other types of AI are based on it in some way or another. In many ways, it's the starting point of a strategic solution and is critically important to AI on the whole. On the plus side, it's also one of the easiest to implement and validate, which we'll discuss in more detail as we move on.

...the majority of AI solutions are in fact a collection of these various practices working together to solve a given problem.

Techniques & Approaches

A deep discussion of pattern detection algorithms is out of scope of this document, but it may be helpful to understand the higher level "how" and some of the primary means of performing this important task.

Pattern detection algorithms are typically applied to a collection of loosely related knowledge. That is, structured or semi-structured data is analyzed to identify the patterns, and that data is generally loosely related by some domain-level means. Pattern-specific algorithms tend to be classified in two ways, either Explorative or Descriptive.

- **Explorative** pattern recognition is the means of identifying patterns in general. This is often associated with *unsupervised classification*, meaning there's no predefined pattern we're attempting to fit the data set into, but rather we're looking at the data as a whole and determining what patterns we can extract from it.
- **Descriptive** pattern recognition deals with the details of classifying an identified pattern; categorizing and describing in general so downstream processes can make use of it.

Patterns can be identified through either physical or mathematical means. Physical pattern recognition is used to identify written text patterns, human faces, speech signals and things of that nature. Mathematical patterns are detected by the use of statistical algorithms, and either physical or mathematical (or a combination of both) algorithms may be more suitable to a specific need.

Business Process

The process of working with pattern detection can be fairly well generalized into a handful of standard steps.

1. **Data Gathering:** gather the required data in question
2. **Data Processing:** clean the data, reducing noise and pre-processing for performance
3. **Data Examination:** run the data through detection algorithms to identify patterns
4. **Segmenting & Classification:** group the elements found into segments and classify them to build a result structure
5. **Analysis:** analyze the result set to come away with insights
6. **Implementation:** use the gained insights to perform some action

Some final thoughts on Pattern Recognition: we must be wary of any AI solution because it's very easy to become complacent about their results, especially when it's doing a good job. However, human monitoring and intervention is always a factor, and AI results can often be misleading or have hidden traps. That said, pattern recognition is one of the more straightforward practices and is less subjective than many other types of AI. We must always be wary of "learned results" and predictive AI results for example, but pattern recognition is less susceptible to questionable output: they're pretty cut and dry. There's either a pattern or there's not, and in comparison to other AI algorithm results, it's relatively for a human to see and validate. This is not always the case with AI, especially as solutions grow in complexity.

And finally, don't forget that the detection of patterns also has a reverse: what it doesn't tell you and lack of patterns can also be highly valuable information (remember the example of bullet holes in airplanes, i.e., Survivorship Bias?).

Predictive Analysis

Predictive analysis is the practice of using historical data, patterns and contextual factors to make predictions about future events. Predictive analysis is heavily based on data, and the more of it that's available, the more accurate the predictions are likely to be. Predictive analysis results are difficult to qualify by their very nature; they provide predictions and hypotheses. In production systems, the only way to verify the algorithm is after the fact. For pre-production, we can time-box historic data to a known relative-future output for testing model accuracy, but by its nature, it is a difficult one to fully verify. We should always be careful to monitor and be ready to adjust and have a fail-safe plan in place. Risk management in critical predictive analysis use cases cannot be overstated.

Predictive analysis uses statistical algorithms to generate predictions and has uses in a wide variety of scenarios. Making use of trends, correlations and statistical patterns, predictive AI is heavily based on previously mentioned Pattern Recognition practices, taking them steps further to provide insight on statistically probable future outcomes. When based heavily on domain and company-specific data, predictive results can shed light on organizational needs (recall that modern AI is used to solve a specific task, and generally, the more specific the better).

There are many business use cases for predictive analysis:

- Financial Services, especially in forecasting accuracy
- Marketing applications, predicting market trends and campaign performance, point of sale traffic predictions and similar
- End User Services, for recommendation systems, user interest and navigation predictions, etc.
- Inventory management in manufacturing and other inventory-critical industries
- Medical applications both large scale (disease outbreak predictions) and smaller scope (predicting patient risks for a particular disease or complications)
- Energy Consumption predictions that can be used to automate environment and resource usage controls
- Mechanical Analysis for preventative maintenance of machines, specialized engines, structures and other tangible objects

Two particularly important aspects not mentioned above are simulation scenarios and automations. Simulation scenarios can provide insight as to expected promotion performances, pricing adjustments or product support in sales, or the simulation of weather patterns for better weather forecasting. Simulation does require heavy input from subject matter experts to tune the simulation for better results.

> Predictive analysis can be a powerful tool capable of providing highly accurate results.

Automation results are another important aspect of predictive analysis, allowing for many tactical advantages. For example, we might make careful, fine-tuned adjustments to jet engine operation based on emerging trends and their future predictions, or we may aggressively trigger data recovery or increased disaster recovery alert levels when predictions indicate abnormal data center operations.

Prescriptive Analysis and Descriptive Analysis are two closely related practices dealing with events and outcomes, both of which predictive analysis is a prerequisite to. Prescriptive Analysis deals with describing why an event is likely to occur, and what could be done to implement an effective solution or workaround to a predicted event. Descriptive Analysis deals with describing events that are currently happening: collecting and analyzing information about the event, often taking into account contextual factors to present a “this is what is” result.

Predictive analysis can be a powerful tool capable of providing highly accurate results. Powerful though it may be, this one should always be taken with a grain of salt. Risk identification and management are of critical importance in implementing solutions that rely heavily on predictive analysis, and the lower in the logical chain such analysis is implemented, the more potential for things going wrong. Especially, be wary of any unexpected events that may factor the area prediction: the predictive outputs are only as good as the models they are trained on, and no model can capture every factor that exists, or possibly be trained on things that may happen in the future outside of what we expect. For this, we prefer to be very cautious by default, always watching.

Natural Language Processing

Natural Language Processing (NLP) is an area of AI that aims to give computers the ability to understand text and spoken words in much the same way that humans do. An excellent candidate for utilizing pre-trained models and further building on pattern recognition and even predictive analysis, language processing has the ability to either understand text or speech or to generate text or speech.

While text processing algorithms have been in existence for a long time in the form of various word and sentence parsing and construction, more recent advances and models are able to take this much further into the realm of contextual processing of text, interpreting sentiment, sarcasm and other more subtle aspects of natural human languages. Modern language processing can go so far as to understand a writer's or speaker's intent and sentiment based on various contextual factors within the passage and potential outlying factors.

Natural Language Processing is split into two primary categories: Natural Language Understanding (NLU), which is the ability to interpret given text with human-like understanding, and Natural Language Generation (NLG) which is the ability to generate human-like language. Of course, the generation portion of this has been made very popular in early 2023 by the ChatGPT service from OpenAI, but there's a much broader scope of capabilities:

- Text translation, such as converting from English to Spanish, or converting text to speech and vise versa
- Processing of spoken commands: the ability for Siri or Alexa to receive a spoken command, translate it to a textual command and interpret the result to extract meaningful commands from it
- Sentiment analysis has proven highly valuable to many organizations in the ability to process reviews, feedback, social media and other user-generated content (even moving into areas of identifying risk in violent or self-harm behaviors)
- Text summarization is the ability to process large volumes of text and interpret it as to provide a summary of the content, or aggregated materials within the content, etc.
- Scanning resumes to extract pertinent information without requiring a standardized structure as we would need with OCR-only solutions

Voice activated GPS systems, assistants like Siri and Alexa, customer service chatbots and voice-processing phone navigation systems are all examples of language processing. Requesting an AI service to process thousands of resumes for a particular role to extract generalities and identify outliers is another lesser known but powerful example of text processing as well.

In many cases, language processing crosses the line into the category of Generative AI (to be discussed in the next section), but before moving into that area, let's consider some of the interesting challenges that NLP has to contend with. With this in mind, it becomes readily apparent that pre-trained models in this offer significant advantages…

Human language has ever been a difficult one for computers to handle as there are so many quirks and variations outside of a common ruleset. Homonyms, homophones, metaphors, idioms, sarcasm, grammar, usage exceptions are some examples of language-level factors to contend with. Add further human elements such as slurred words, fast speaking, lingo, accents and incorrect grammar and the problem becomes even more difficult. These and many other considerations must be factored into reliable processing and generation algorithms. NLP's primary tasks are interpreting all of these factors, generally by using a set of well defined practices and techniques:

- **Part of Speech Tagging**, also known as grammatical tagging, determines the part of speech of a particular work or piece of text based on its usage and context. For example the word “make” could be interpreted as either a verb or noun depending on its context: *I’m going to make a bet* and *The make and model of that car*, for example.
- **Word Sense Disambiguation** takes this a level further, in being able to determine the sense of a word based on context: “make a bet” and “make the grade” for example to “place” or “achieve,” respectively.
- **Named Entity Recognition** is the ability to identify that a particular word or group of words is specifying a named entity, such as the state of “Florida” or the person “Mary”
- **Coreference Resolution** is used to determine when two words refer to the same thing, which could be grammatical (“she” and “Mary” both relating to the same entity) or metaphoric (“bear” as in a particularly hairy man). These are key piece of understanding the larger intent of a passage
- **Sentiment Analysis** deals with the ability to extract subjective qualities from text, including attitudes, emotions, sarcasm, confusion, suspicion and others.

Together, along with various other factors, NLP AI models are able to interpret and generate text at increasingly advanced levels of accuracy. Language models continue to grow and evolve (now being named as such: Large Language Models (LLMs) for example) and algorithms have reached a point over the last few years to be considered accurate, stable and reliable in their ability to process human languages.

Generative AI (Text, Image, Video…)

Generative AI is all the rage these days. In particular, ChatGPT has become a household name, bringing AI to the forefront of even non-technical minds, all in a matter of months. Its ability to answer questions and generate content is astounding, from writing technical essays on a topic under a particular style to generating victory song lyrics as if you were a viking that just plundered a village, there's seemingly no end to the amount of things it can do.

Impressive as it is, ChatGPT is more of an end-user playground that showcases the abilities of Generative AI, particularly those abilities of working with text. However, GenAI has many serious business use cases and goes well beyond mere advanced text generation. Let's first take a look at how it works and how we got here, then we can review some general capabilities based on that.

General Usage & Capabilities

GenAI, as the name implies, deals with generating content. Content can come in many forms, but the most common are text, image, video and audio. Prior to current days, the last big push in GenAI was in the form of image generation. Remember a decade or so ago when we had tools popping up that could sharpen old and blurry images and add aging to a face? These are examples of generative AI at work: the ability to interpret an input such as an image, process it against some algorithm, and produce a desired output.

Popular current-day generative algorithms work by accepting a number of input parameters (generally a very large number, ranging in the billions), processing them against a large trained data model, and producing some sort of output.

Recent advances in language processing and the sheer size of trained data models have given way to current abilities. The more data it has to draw from, the more capable the algorithms can be. ChatGPT for example can use modern day language processing to interpret a request, convert it into various parameters for its generative algorithm, consult its vast models of information and produce credible results. The same general process goes for other types of content, such as imagery and videos.

Let's take a quick look at some general capabilities of GenAI algorithms:

- **Text and Language:**
  - ***Marketing Content:*** the ability to write blog posts, site content, copy for fliers or brochures
  - ***Note Taking:*** the ability to interpret raw data or voice notes to output structured format or summary notes, particularly useful in medical fields
  - ***Code Development:*** the ability to have basic code written on your behalf, which can be dropped into a larger codebase and tested
  - ***Essay Writing:*** the ability to provide topic information and an outline, to which GenAI can output an essay of a particular type based on your request
  - ***Documentation:*** the ability to generate technical documentation (excellent for systems development) or legal documentation are some core uses of GenAI
- **Visual:**
  - ***Image and Video:*** the ability to create images and video based on input prompts is useful in many places, from marketing to signage to logo design and beyond
  - ***3-D Models:*** the ability to take a 2-D picture of an object and have it generated into a 3-D computer model has numerous implications in gaming and simulation development, CAD modeling, manufacturing and more
  - ***Design Work:*** GenAI can have a place a

tool to help boost inspiration for company branding, marketing design materials, interface look and feel, logos, cover art and even fashion

- **Auditory**
  - ***Voice Generation:*** we're starting to see response algorithms and processing powers fast enough to have voice chats over the phone with a bot…
  - ***Music Generation:*** the ability to create music or "jingles" per various prompts and context can be a creative boost to artists and can help with audio advertising and ambience setting
- **Simulations**
  - An interesting and powerful use of GenAI within organizations is its potential to create business simulations which can then be utilized to gain insight as part of and along with predictive analysis work
- **Data Generation**
  - Another useful case for GenAI is in data generation, sometimes referred to as synthetic data. Often we have partial sets that are missing data, or have restricted data, and given enough of a base to work with and the size of the gaps, GenAI is capable of producing fills or completed sets that serve further purposes.

The power of Generative AI is immense, especially when coupled with other practices and techniques as part of a larger solution. As one of the most advanced and capable types of AI available today, it also carries with it some of the highest costs of effort and responsibility to utilize.

Data Models & Processing

Let's take a bit of time to discuss the data models at work here. Generative AI (and indeed, most sorts of AI) utilize what we can generally assume to be "trained models," meaning that the data model and algorithm has been reviewed and structured in some way, either by human or machine. Generally speaking, the thought is that the larger and more inclusive these models are, the more capability and accuracy the algorithmic results can be.

If you train a model based on one or two paintings by Salvador Dali then ask your GenAI to produce a Dali-like image, the results aren't going to be as good as if you train your model on the entirety of Dali's work, which could potentially include some personal characteristics of the artist himself. Then take a much larger and more generalized example, such as compiling the entirety of the information contained in Wikipedia into a model, along with other general information sources, and we start building much larger data sets and models that can be trained and re-trained for further accuracy.

> **A key takeaway here is that the publicly available GenAI models are generally trained on massive data sets, far beyond what most individual organizations have readily available in their own stores. These models tend to back popular services that utilize pre-trained models, which can be a huge boon for organizations that have neither the resources nor knowledge to train their own. In addition, many of these models are collaborative efforts by some of the largest technology firms in the world.**

Proprietary Data & GenAI

Access to large, publicly available models as the result of someone else's effort is a fine thing, but what happens when you want to use GenAI against your company's own proprietary data? Whatever information you provide to these types of models has a strong tendency to become a part of that model. Training our own internal models to match those of the big giants isn't exactly feasible, and there are definite concerns about handing over sensitive data about our company: ethical and legal constraints, proprietary data, inadvertently making sensitive information available to a wider audience, etc. How, then, do we go about utilizing the power of GenAI while keeping our data close to home?

Data governance is always something an enterprise should be concerned with, and those concerns become much more pronounced when dealing with AI, and Generative AI in particular.

One method for dealing with proprietary data in GenAI is to utilize the larger public models against what would essentially be sanitized proprietary data. This is referred to as Model Distillation, and some creative mixtures of public large-model processing and internal adapters and data distillation are emerging as viable means of getting the best of both worlds; or at least a bit of both worlds. Consider that large GenAI algorithms have billions of input parameters, and it becomes easier to imagine that we can clean some proprietary data into something suitable to be passed into such algorithms. Another example of this involves a Stanford group that asked GPT-3.5 to generate thousands of paired instructions and responses, then put that into their own model to produce an "internal" version of GPT-like conversations. Since, other mix-mode models like Vicuna and Dolly have done the same.

Another consideration is that many public models may be suitable and allowed to be forked or instanced into a private copy, which would imply that you can then control both the model and your data that you'll use the model to process. However, a catch here is that currently, many of these carry concerns about what the algorithm may be doing, even if it's hosted within your own organization's infrastructure. For example, regulated industries must be extremely careful of open source solutions, taking steps to verify that there's no code that's "phoning home" to some third party. With AI, even open source algorithms are for practical purposes still a black box as the complexity of the algorithms is such that they can be extremely difficult to verify.

On another note, there has been evidence in recent studies that smaller models with domain-specific data can outperform large, generalized models, if the questions being asked of it are indeed specific to that domain. Consider the example given regarding Salvador Dali's collection of works, which would be very domain specific, and quite small when compared to large models. Processing times are faster and for domain-specific questions, results can be better as well. For a less theoretical example, PubMedGPT has its own data model specific to the medical domain and can answer medical questions with higher accuracy and faster response time than the more general ChatGPT service. Thus, there seems to be a case for smaller, domain specific models, and as the ChatGPT hysteria continues to wane, perhaps we'll see more pragmatic usages of these smaller model practices emerge.

Without doubt, Generative AI is very powerful and is very much at the forefront of the current wave of AI. While the technical aspects of this may present its own technical difficulties, this is also an area that's in the focus of many legislative efforts. Data governance is always something an enterprise should be concerned with, and those concerns become much more pronounced when dealing with AI, and Generative AI in particular.

What About Machine Learning?

We've covered a lot of material about services, pre-trained models and general capabilities under the context of AI, but you're probably wondering why we haven't yet mentioned Machine Learning (ML), or Deep Learning (DL) or neural networks or many of the other terms that tend to accompany AI. The reason is twofold: first, there's really been no need yet as Machine Learning is a subset of AI, and we've only covered AI in general, and second is explaining ML after having some general AI context is a lot easier. Deep Learning (and a few other common terms) we'll touch base on toward the end of this section.

The only two we really need to know about are Artificial Intelligence (AI) and Machine Learning (ML). The rest can generally be considered an implementation detail and not highly relevant to gaining an executive level understanding of AI. However, we do want to define Machine Learning in general and why it's important.

What Is Machine Learning?

AI, at its most abstract definition, is simply computers that attempt to understand things in the way that humans do. That has a pretty wide scope. Machine Learning is an area of AI that deals with how computers are able to "learn" on their own. Machine Learning is a core practice within the realm of AI. It's part of AI, but it is not all of AI. Machine Learning is also much less abstract than AI. While AI has a very generalized high level definition, Machine Learning is a far more concrete set of practices, methods, algorithms and other technical aspects that contribute to AI on the whole. In many ways, AI is an idealistic thing, where ML is a scientific, process-bound thing.

Machine Learning deals with the ability for a computer program to "learn" to do things better. At the technical level, this goes very deep with some of the most impressive and algorithms ever devised working at the tasks. At a more general level, this can be easily scoped and considered in our larger ponderings of AI on the whole.

Before we get into an example, let's recall some statements made in the *AI – What Is It, Really?* intro to this document. AI comes in two primary forms: weak and strong. Weak being AI that is designed to do a specific task, and Strong being a theoretical form of AI that would seek its own knowledge, learn without human direction and do a vast array of non-instructed things of its own accord. In much the same sense, Machine Learning is the same in that for all practical purposes, we are targeting a specific task to be improved by learning process. What we're not doing is throwing a massive amount of data into an algorithm that will make its own decisions outside the scope of what it's been told to do. A machine, for example, will not "learn" to overthrow the human race using its advanced computing powers that it built in secret from its once-master humans. Rather, ML has the capability to do something along the lines of: *here is a desired end goal; here is what we have now; here is the input parameters, and with this, iterate over some process we'll guide on, and determine a more efficient or accurate way to reach the end goal through trial and error.*

It could be argued that the vast majority of progress in humanity is attributed to trial and error. Civilization doesn't advance by just one day understanding the secrets of some domain and applying it, but rather a gradual and often methodological exercising of some knowledge and feedback loop. Doing something, finding out what works and what doesn't, incorporating that feedback and doing it again, and continuing to do so until the process is very well refined.

In many ways, ML is the same. Essentially, it's the process of taking a pass at something, learning what worked and what didn't, adjusting for it, and repeating. In some cases, as the model that's being "trained" becomes more near to the desired output, some further automation can be put into the ML algorithms to have it do much of this on its own. In earlier stages of training, more human intervention is required to make adjustments and incorporate their "real world knowledge" into the process. In that case, we can't simply throw some mass of data at an ML tool and say "ok, sort through all that and let me know what it really means." Rather, we do have to instruct it on how to do so, what to look for, what the results may mean, etc. Despite the name "machine" learning, the process still relies heavily on input from humans in order to fully train a model in the complete sense.

Let's consider a real world example of a manufacturing company that makes use of an "intelligent" robot whose job is to select a piece of raw metal stock from an adjacent rack and load it into a cutting machine. In its first iteration, the robot is fairly simple in that it expects each of the pieces of raw stock to be aligned in a neat array, oriented the same way, all of the same length. The robot does its job well, as long as the raw stock is aligned the way it's supposed to. On the occasion that a piece of raw stock is slightly misaligned though, the system ceases to function. Perhaps we can improve the stock selection algorithm to be a bit more intelligent and handle these cases?

There's two potential ways to go about this. Let's call it "old school" and "fancy new way":

- ◆ The old-school method would involve a human writing into the algorithm the expected ways that a piece of raw stock might be aligned, and improving the algorithm to recognize a misaligned piece of stock, determine the "new" orientation of the stock, then turn its robot hand at a n-degree angle, *then* reach in and grasp, upright, and extract the stock from the shelf.
- ◆ The fancy-new-way would be to tell the ML algorithm something along these lines
  - We expect that there will be misalignment of raw stock on the material rack
  - Success is the ability to grasp and extract the raw stock from the shelf so the end result of that particular task is the same as if it were a properly aligned piece of raw stock
  - Given a detection of misalignment, determine the ideal method to reach the desired outcome. To do so, you'll use a grasping-arm-rotation attempt method.

At this point, the ML algorithm would have 1) a goal, and 2) some instructions as to how to achieve that goal. Now, the ML algorithm has the capability to, on its own, attempt various methods at the solution. Given many trials and errors, it can consult its history of what's been successful, correlate its pattern recognition, run some predictive analysis and generate a new action plan for the next round. Continued cycles feed back into this array of information, and human monitoring allows us to make sure the machine is "learning" on the right track with the ability to intervene and re-adjust the algorithm as necessary.

Even the above is a fairly simplified example in that we as humans did provide a good "hint" to the algorithm on how to approach the solution. Using more advanced pattern recognition, predictive analysis and generative AI to create new potential workflows and even simulated scenarios, we could make the machine even more "intelligent."

While the above scenario is somewhat tangible in the fact that the intelligence is driving a robotic arm, the same general process can be applied across the board for a great many applications. The ability to program a machine to create and interpret its own feedback, with human adjustments to keep it on track, is a concept with massive implications across any sort of industry.

Why Humans.. Isn't this "Machine" Learning?

You may notice that we keep mentioning human input to the process. This is very important, and raises an interesting point. Process, in general, is pretty easy when approached at the basic level. Take an example of putting a work order through a manufacturing shop, start to finish. Despite that it may be a fairly complex process, will take some time and may at times be tedious to map out, it's not a particularly difficult task. Even capturing unknown exceptions to the process isn't bad once they make themselves known; it's a relatively straightforward matter of refining the affected area of the process. The real difficulty in modeling a process like this is capturing senior level intuition and wisdom about it. Most organizations have their handful of "wizards" that have been around long enough and done enough that they "just know" that if you tweak this a little bit here, that over there comes out as needed, despite the fact that "this little bit here" isn't part of the written process, or that if things start to go sideways, you can get them back on track by making a little nudge over there. There's a very subtle gem in here...

In some ways, the heart of AI and ML is about capturing those subtle pieces of wisdom and uncharted knowledge that the masters of their craft have been able to cultivate over the course of their career. For domain-specific intelligence, these are often the pivot points upon which "smart" applications are developed. After all, training models on the easy stuff is, well... easy. Innovation and competitive advantage doesn't come from the norm.

Effort & Processing

It's worth mentioning resources required for ML projects, particularly in terms of processing power, infrastructure and qualified personnel. An understanding of the general resources required will be a primary consideration in AI\ML project feasibility, and tends to underline the value of pre-trained models.

Processing Power

In the early days of AI, processing power was a primary issue that prevented AI from progressing through the ideas that were logically feasible. Quite simply, there wasn't enough processing power available to handle the heavy computations required, even though AI pioneers knew what such algorithms could be capable of. Since then, Moore's Law – which estimates that available computing resources doubles each year – has brought us much further along and into modern-day AI\ML capabilities.

Even so, training models in ML requires a massive amount of compute resources. Cost, environmental impact and time to process are all considerable factors. Larger cloud service providers (AWS, Azure, GCP) have provided compute services aligned for the task, but the larger the models to be trained and the more complex the training to be performed, the more significant the computing resources required. Even in current times, with Moore's Law generally slowing, there are a great many ideas for advanced AI\ML that are not computationally feasible or possible. Perhaps the expected rise of quantum computing with its orders of magnitude more power will open further doors and bring the next wave of AI and ML capabilities with it. In any case, here and now it's worth recognizing that training ML models is among the most resource intensive processes of modern days, and our potential adoption of the technology should bear that in mind.

Data and Pipelines

Processing aside, ML (and even AI in general) is somewhat of a "you need to be this tall to ride" type of scenario. Years ago while researching microservices and studying various material on it, Martin Fowler said "you need to be this tall to ride microservices," meaning that if a company did not first have in place a sufficient infrastructure for automated monitoring and reaction to infrastructure events within a microservice environment, it would be disastrous. Much the same with AI and ML, except that it's a prerequisite to not just safely handle it, but to even start it at all.

AI is fundamentally based on data models, and ML is trained against data models. That is, data is the lifeblood of everything happening in AI and ML. In order to do any serious work with customized AI\ML, a company must have 1) data, and 2) pipelines sufficient to process that data. A good indicator of the ease at which a company can adopt serious AI and ML initiatives is by looking at the maturity of a) their Analytics & BI platforms, and b) their infrastructure pipelines.

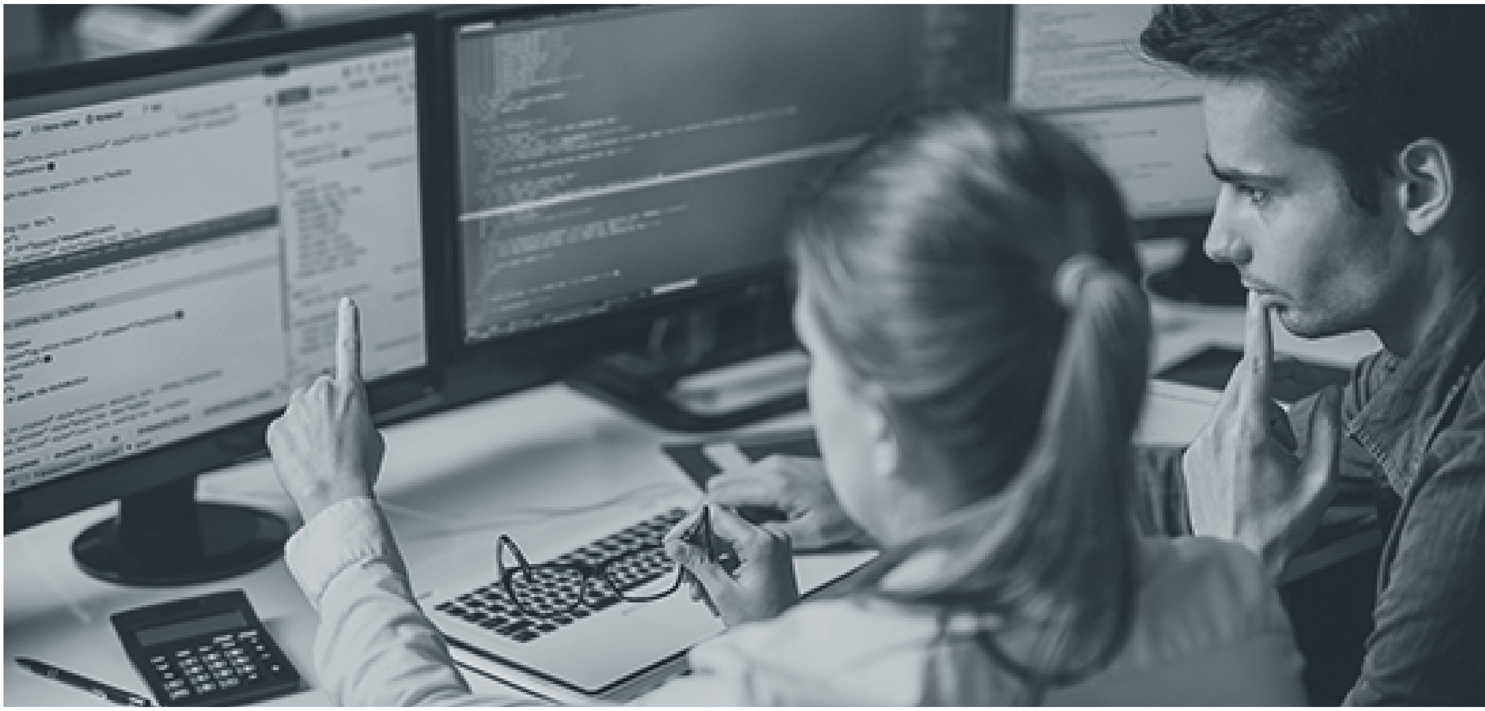
An established and well functioning analytics stack sets the groundwork and good data hygiene required to work in the AI realm. Perhaps not so much in the presentation end of things, but surely a solid and high integrity backend process of data ingestion and warehousing is a key discipline to do much of anything with a custom AI build. In many ways, analytics stacks are the baby brothers of AI, in that the data requirements for AI are that of analytics, taken to much more stringent levels. In particular Data Governance practices (the process of managing availability, usability, integrity and security of data) are of utmost importance, and poor implementation of this could even land a company in legal trouble.

Automation pipelines are a second pillar required to hold up AI\ML initiatives. Automation pipelines are most commonly known for their role as DevOps, particularly within CI\CD systems (Continuous Integration\Continuous Delivery). In the past decade DevOps has become a standard even for small development efforts, so that's a plus, but there's also a big difference between a personal GitLab project with some basic CI in place and enterprise-grade automation pipelines built by a solid platform engineering team. Enter MLOps: the ability to automate at the platform level in order to move large datasets and perform any number of other required tasks that simply can't be done manually.

The bottom line? If you currently flounder with half-baked data management and automation platforms, implementing an AI and ML solution is going to be extremely difficult. While it doesn't necessarily mean any existing implementations need to be fixed first, it should stand as a strong warning that movement into the AI\ML realm will require serious investment and strict discipline.

People

Training and advancing a machine learning model can be a complex and resource-intensive task that often requires a diverse team with various roles and expertise. Here are some key personnel and team members that might be required for training and advancing a machine learning model:



- **Data Scientist/Machine Learning Engineer:**
  - Responsible for designing, implementing, and fine-tuning machine learning models.
  - Develops data preprocessing pipelines and feature engineering techniques.
  - Selects appropriate algorithms and hyperparameters for the model.
  - Manages training, validation, and testing phases.
  - Monitors and evaluates model performance.
- **Data Engineer:**
  - Collects, cleans, and preprocesses the data required for training.
  - Sets up data storage and retrieval systems.
  - Designs and implements data pipelines to feed data into the model.
  - Ensures data quality, integrity, and security.
- **Domain Expert/Subject Matter Specialist:**
  - Provides domain-specific knowledge to guide model development.
  - Assists in feature selection and validation of results.
  - Helps in interpreting and applying the model's output in real-world scenarios.
- **Research Scientist (for advanced research projects):**
  - Contributes to novel algorithm development and research.
  - Stays updated with the latest advancements in the field.
  - Collaborates on pushing the boundaries of the model's capabilities.
- **Software Engineer/Developer:**
  - Integrates the machine learning model into existing software infrastructure.
  - Creates APIs and interfaces for model deployment and usage.
  - Implements real-time monitoring and error handling mechanisms.
- **DevOps Engineer:**
  - Manages deployment, scaling, and monitoring of the machine learning model in production.
  - Ensures high availability and reliability of the deployed model.

  - Implements version control and rollback strategies.
- **Quality Assurance/Testers:**
  - Conducts rigorous testing of the model's behavior in various scenarios.
  - Identifies and reports issues related to model performance or behavior.
- **Ethics and Compliance Specialist:**
  - Ensures that the project adheres to ethical guidelines and legal regulations.
  - Addresses potential biases and fairness issues in the model's predictions.

The exact composition of the team can vary depending on the scope and complexity of the machine learning project. Some roles might be combined or shared among team members, especially in smaller teams or startups. Effective collaboration, clear communication, and a diverse set of skills are essential for successfully training and advancing a machine learning model.

Pre-Trained vs. Ground-Up\Fully Custom Models

When talking about Machine Learning, we're not suggesting that every AI solution will require a host of resources, personnel and advanced automation pipelines to see valuable results. In this, we're certainly veering well into the "from scratch" realm of the AI\ML landscape. There is a wide variety of options utilizing pre-trained models and utilizing other people's effort, and many opportunities to utilize pre-trained models paired with lighter-side proprietary implementations. As with most things, mixing and matching to find a good balance of customized needs coupled with readily available services tends to make the most sense.

Deep Learning, Neural Networks (and others)?

All this about Machine Learning is well and good, but what about Deep Learning (DL), Neural Networks, Large Language Models and many of these other terms I come across? Where do they fit in?

Generally speaking, they're not too much to worry about at a high level. Neural Networks, for example, are a specific classification of model that Machine Learning might utilize for its task, but also are a bit beyond the scope of laying out the general AI landscape. While it does downplay it to a large extent, for practical purposes they can be considered somewhat of an implementation detail, to be explored more deeply at need. Deep Learning is much the same, being a sub-categorization of Machine Learning itself and a means by which ML can reach a desired outcome. Large Language Models (LLMs) are pre-trained models vast in scope and size used to power many of the generative AI services available today. Small Language Models (SLMs) get less lip service outside the techies, but are likewise an implementation detail for our purposes in this document.

A general understanding of Artificial Intelligence and Machine Learning, without diving into specifics of the various model types and or science and mathematics of algorithms is enough to provide a lay of the land and arm one with enough information to get a start in thinking about how to adopt these technologies into the organization.

Ethics & Legislation

It's important to understand the ethical responsibilities of using AI, and legislation is something that we'll want to keep an eye on as well. As with most leading edge technologies, legislation is often well behind the technology itself, and is also often the result of using (or potential to use) the technology in an unethical manner. It therefore stands to reason that by adopting AI technologies in an ethical manner, we can – to a certain degree at least – somewhat future-proof ourselves against inevitable legislation surrounding the technology.

At the time of writing, there is relatively little legislation in place governing the use of AI technologies, though there are a number of pieces in the works. For example, the United States is in process of working up an AI Bill of Rights, which explains primary principles that should be followed, and the EU has its AI Act that defines levels of risk with types of AI and regulation around those risk levels. Aside from those, there are numerous targeted regulatory actions being taken almost on a daily basis, and many US States are putting regulations around how AI can be used, what needs to be disclosed and so on.

While the purpose of this document is not to inventory legislative concerns about AI technologies, we can briefly review the above to set some basic expectations. Before doing so though, let's first look at some of the ethical concerns regarding AI, as these are the baseline concerns for much of the legislative efforts.

Ethical AI

Ethics in AI involves the moral principles, values and guidelines that lay the groundwork for development and use of AI technologies. The goal is to ensure that AI systems are designed and used in ways that align with human values and promote the well-being of individuals and society as a whole. Some key ethical considerations are:

- **Transparency & Explainability:** AI systems should be designed to provide understandable explanations for their decisions and actions
- **Fairness & Bias:** AI systems should be developed and trained to avoid biases and discrimination in their decisions and outcomes
- **Accountability:** Designers of AI systems should be held accountable for the impact of those systems
- **Privacy:** AI system should respect user data privacy and handle sensitive data accordingly
- **Safety:** AI systems should be designed to operate safely and minimize potential impact to environments and people. This is especially true in applications of Autonomous AI, including autonomous vehicles and robotics.
- **Social Impact:** AI systems should be designed with positive social impact considerations in mind; its effect on jobs, economic equality, social structures, etc.

The above list is not fully-inclusive, but provides some basic groundwork for the types of considerations that should go into building AI solutions. One way to help ensure the responsible use of AI is to implement an AI Ethical Risk Framework tailored for your organization. This would generally include:

- Articulation of ethical standards (both good and bad) of the organization
- Identification of relevant internal and external stakeholders
- Generation of a recommended governance structure
- Articulation of how the structure will be maintained during changes in personnel and circumstances
- Establishment of KPIs and quality assurance programs to monitor the effectiveness of the framework

The goal is to ensure that AI systems are designed and used in ways that align with human values and promote the well-being of individuals and society as a whole.

Legislative: a Rough Summary

Having covered some of the basic ethical considerations of AI systems, let's quickly review the two primary pieces of legislation: the United State's Blueprint for an AI Bill of Rights, and the European Union's AI Act.

AI Bill of Rights (USA)

Reference: https://www.whitehouse.gov/ostp/ai-bill-of-rights/

The following is a brief, paraphrased summary of the primary rights of users identified in the reference link above:

- Users should be protected from unsafe and ineffective systems
- Users should not face discrimination by algorithms and systems should be designed in an equitable way
- Users should be protected from abusive data practices via built-in protections and users should have agency over how data about them is used
- Users should be notified that automated systems are being used and understand how and why it contributes to outcomes about the user
- Users should have the ability to opt out when appropriate and have access to personnel who can consider and remedy issues they encounter

AI Act (EU)

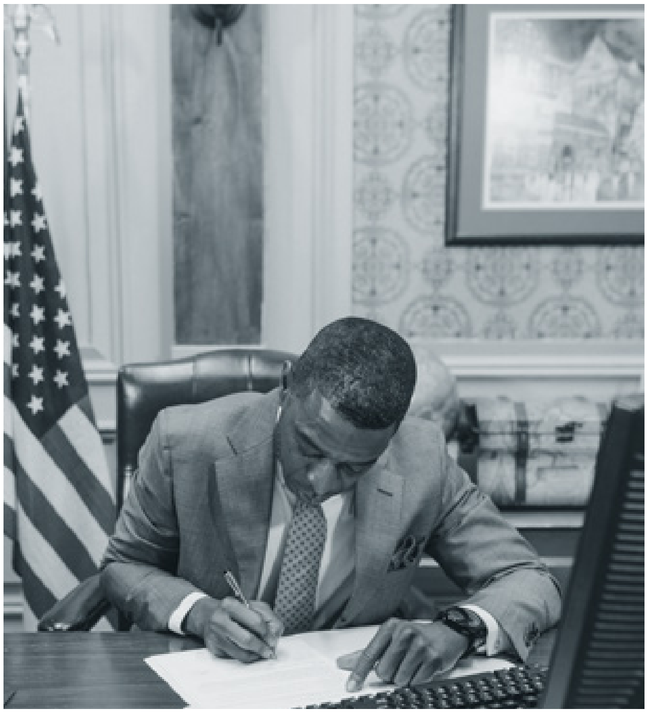
Reference Links:
https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021PC0206

https://www.europarl.europa.eu/news/en/headlines/society/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence

EU's AI Act is paraphrased below as we saw fit for this document. please see the reference link above for further details.

The AI Act defines varying rules for varying risk levels of AI. Included is the identification of those risk levels, thus implying that AI solutions we may build should be categorized accordingly.

- **Unacceptable Risk:** AI systems considered a threat to people and will be banned, including:
    - Cognitive behavioral manipulation of people or specific vulnerable groups (e.g., voice activated toys that encourage dangerous behavior in children)
    - Social Scoring: classifying people based on behavior, socio-economic status or personal characteristics
    - Real-time and remote biometric identification systems, such as facial recognition
- **High Risk:** AI systems that negatively affect safety or fundamental rights are considered high risk and are divided into two areas. Each will require review and approval before being put on the market
    - AI systems used in products falling under the EU's product safety legislation (toys, aviation, vehicles, etc.)
    - AI systems falling into specific areas that will have to be registered to EU databases (e.g., biometric categorization, critical infrastructure management, educational training, etc. – see the reference links for further details)
- **Generative AI:** GenAI systems would have to comply with transparency requirements:
    - Disclosing that the content was generated by AI
    - Designing the model to prevent it from generating illegal content
    - Publishing summaries of copyrighted data used for training



- **Limited Risk:** Should comply with minimal transparency requirements allowing users to make informed decisions, including that of which they want to continue using it. Users should be aware when they're interfacing with AI.

By no means does this portray the full coverage of regulations around AI. There are many facets to this that are left unsaid, and the playing field is changing on a daily basis. Naturally, the legislation of AI systems in general will continue to evolve, and while we will update the whole of this document periodically to capture changes to the field in general, do note that the above information is provided merely as a means of awareness and not as a definitive framework in which we should base our systems.

If there were three primary factors we could keep in mind for this, we would consider 1) ensuring our AI solutions are not biased, 2) disclosing how AI is used and what decisions it is making and 3) data governance: ensuring complete control of the data in use.

Costs & Risks

Measuring the ROI of an AI project is much like that of any other technology development effort, though due to the wide scope AI consists of and some heavier unknowns in the modeling and training phases it may require a bit more diligence to gain confidence. With that said, we're not here to explain how to calculate the total cost of ownership (TCO) and return on investment (ROI), but a brief look at the primary factors regarding AI-specific projects can provide some guidelines for incorporating it into otherwise standard practices.

Calculating Cost

Let's approach this under the context of a custom built, "significant" AI solution, meaning that it's a fair bit more involved than a citizen developer plugging into off the shelf services. A considerable factor is whether we're able to use pre-trained models or whether we have to train our own (or often, some mix of the two). The general list items below apply to either or, but will tend to have heavier impact for anything built from the ground up.

- **Business Understanding:** naturally, we can't do much without understanding what and why. Clearly defined expectations help firm up ROI calculations.
- **Data Exploration & Preparation:** we usually know roughly what data we have or can get, but what kind of state it's in and how much it will take to get it into good general shape is a considerable portion of working with AI. Clean-structured warehouse data is a great starting point, and messy lake data probably has data we'll need, and we'll probably have some initiatives to pipe further data in from external sources as well. Don't forget that filling missing data may be a required step here also. This phase doesn't include modeling the data as needed for algorithmic processing, it's just a matter of getting everything into a clean base structure and ready to be modeled. This area should also include efforts for addressing data governance and security.
- **Modeling:** working with data at this level is a big shift from the exploration and preparation phase. Here, we're actively designing the models and algorithms, processing them, training them, iterating over them until we have something satisfactory.
- **Evaluation:** this primarily involves reviewing the results of the model and making decisions on whether it's satisfactory or needs further processing, and additionally includes reviewing the processes in use and common risks to see if we need further adjustments.
- **Deployment:** deployment of a satisfactory model into production can be a heavy operation with many dependencies and disruptions. Of course we've considered this due diligence already in terms of scoping the project itself, but we'll want to look at it again for costing as it

generally involves change to existing processes, technologies, workflows and the like.

- **Operating and Maintenance:** let's not forget the infrastructure and resource costs of both the pre-deployment phases as well as the production operations. Computing power, data storage, pipelines, monitoring, reliability engineering, business continuity and disaster recovery efforts all are aspects that should be considered
- **Continuous Improvement:** AI systems are rarely a once and done type of solution, but tend to offer ongoing value and are high impact candidates for continuous improvement.
- **Forced Improvement:** it's worth noting that third party services and models, or even internal dependencies, may evolve quickly (or not quickly enough), thus forcing us into having to update and adjust. Consider the system dependencies and their risks in order to determine the cost of keeping up with it when needed. New legislation may also force our hand, so that's a risk to factor in as well.

In particular, the modeling and evaluation phases tend to be the largest unknowns and hardest ones to accurately estimate. AI systems are iterative in their nature, and these two phases could cycle many times before reaching a satisfactory result that can be considered for production deployment. Consider the complexity of the problem being solved and use that to gauge the effort required in this area.

From this we should be able to pull together a reasonable TCO estimate. ROI calculations after that are rather typical and heavily dependent on the nature of the problem and expected solution. We can work up estimated timelines and returns for the effort and proceed from there.

Time to Obsolescence

One thing not mentioned explicitly above, and we feel worthy of its own heading, is keeping in mind the time-to-obsolescence (TTO) for AI systems. This is a difficult one to estimate, and a significant downside of cutting edge technology adoption. In an early-curve, fast paced technology like AI, the TTO is short; much shorter than we may like. This isn't set in stone and can depend on many factors (the Risk Mitigation section touches on this a bit more), but a 12-mo TTO for AI is a decent average as things stand at the time of writing. This doesn't necessarily mean that any AI system we build will be obsolete in a year, but we do want to give this due diligence consideration when designing our systems.

Mitigating Risks

Risk is ever a factor, especially in higher-end investments and when dealing with cutting edge technologies. While a complete review of risk mitigation is something specific to each project and system, it's probably worth touching base on some generalities as they pertain to AI. Here's a few things we can keep in mind:

- **Check in early and often, and be ready to call off.** Especially during the modeling and evaluation phases, which is where we're really proofing the "brains" of the system, be especially diligent while reviewing the outputs of the model. They should generally be trending in a direction that would indicate success, and don't be afraid to put the brakes on and back up a few steps to re-assess on the whole if they aren't trending expectedly.
- **Aim for a proof of concept:** it may be a little more difficult to build a POC for an AI system given that so much of the success of the project is tied up in highly technical details of the training aspects, but it is possible. Don't forget also that expected results can be mocked much like any other data for integration and testing purposes.
- **Consider business continuity and disaster recovery:** building AI systems that pose a centralized risk to your business may be difficult, and BC\DR can be significantly more challenging with AI systems. This can cover

a number of areas, from infrastructure DR (generally straightforward), to how to react to models gone awry (emergency re-training plans), to what happens if a third party service comes offline. A comprehensive risk assessment of the system and its impact throughout the organization is nothing short of critical.

- **Plan for service outages** and determine how long your system can survive, and\or how long dependent systems can survive without it. This is common technology risk assessment practice, but being that many of the third party services available – even those of major players – are still new and have intensive resource requirements, it's likely to happen much more often than working with service providers that are comfortably on the plateau of a mature technology lifetime.
- **Consider Time to Obsolescence in all corners of the solution:** be particularly wary of grabbing the latest and greatest shiny new toy in the realm. As much as possible, try to stick to the basics, as they're the ones likely to pass the test of time. Venture off the beaten path only at significant need, and do so cautiously. Play the thought game of considering what the field may look like in terms of services and methodologies a few years from now, and see if your strategy still fits that vision… it doesn't matter so much how accurate the vision is (a few years ahead in AI? hah!), but how our strategy might hold up to that vision is often a good indicator of its overall endurance.
- **Adopt and implement solid development methodologies:** from dealing with data platforms to running compute and infrastructure to automation pipelines and even integrations with existing systems when deploying, these should be well-known, highly controlled practices running at the pinnacle of quality control. Having a mountain of AI effort come crashing down because of slipshod development practices would be a disgrace to the integrity of our amazing field of practice. These are the foundations of our AI systems; let's treat them as such.

Any serious business endeavor should be given an in-depth risk assessment, and AI systems are no different. Typical risk assessment models and practices can be employed, and giving extra consideration to the items above along with a comprehensive risk assessment can provide the confidence needed to make sound decisions on whether an AI system can feasibly be brought to the table and how it can be handled once it is part of your organization.

A Healthy Dose of Pragmatism

We've covered a lot of ground so far, and hopefully the information provided has been able to give a broad view of the field of AI, and how it may be used. Before closing, there's some things worth consideration in what not to do, what to expect, what to be careful of, etc. Let's cover some of those things here and leave them as closing thoughts.

No Silver Bullet

Fred Brooks wrote in The Mythical Man-Month (ISBN 978-0-201-00650-6, an excellent set of essays that anyone involved in software development should at least skim through) that "there's no silver bullet," essentially meaning that despite decades of technological innovations, there is no one tool or technology that will solve all of your problems. Written nearly fifty years ago (1975), this still holds true today, and has had years of proof since it was written.

Artificial Intelligence is a tool. It is "A" tool, and not necessarily "the" tool. We may know that AI offers incredible technological power, but there are certainly significant costs and considerations involved in attaining that power. Be careful of being complacent in your ideas: research them thoroughly and understand the true limits and technical envelope of your work.

Watch, Watch, Watch

AI systems need a guiding hand. Constantly. Often with custom software implementations, it's the case that an organization implements them and then more or less leaves it alone once it's in place and working adequately. With AI, we're implementing highly advanced, cutting edge technology that is designed to essentially do its own thing. It's imperative that we don't lose sight of this fact and continually monitor and redirect what it is we've asked the system to do.

You Must Be This Tall To Ride...

To implement solid AI driven solutions, there needs to be a strong framework in place for managing it. Organizations that are unfamiliar or undisciplined with data management and automation pipelines will have trouble implementing AI solutions that go deeper than the citizen developer and pre-made service levels. The algorithmic power of AI is exponentially more advanced than that of traditional software, and as such requires strict discipline to handle the need for good, clean data and processes that enable it, lest the garbage in turn into that much more garbage out.

Careful with your Data

Without doubt, AI is data, data and more data. We should be especially diligent about how we're using the data, particularly with a) ethical concerns, and b) proprietary concerns. When designing AI systems, we should be ever-cognizant of where our data is, what services or people we're giving it to, and ensuring we're following strict security protocols to avoid data breaches.

Data Governance

Did we mention data yet? Let's say it again… Seriously, we've mentioned data a lot, and the risks of misusing data, especially in AI and ML cannot be overstated. We must be serious about how we're handling our data, and solid data governance practices are by far the best safety net we have. A discussion of the topic is out of the scope of this paper, but suffice to say that this is important.
Data Governance

Did we mention data yet? Let's say it again… Seriously, yes, I know. Please read the preceding entry again though… pretty please? With a cherry on top? Thanks.

Service Outages

Most AI systems are implemented using third party services. Given the vast amount of computing power required to offer these services, and the fast-paced rate of changes, updates, fixes and other stability considerations, it's not at all uncommon to see services being temporarily taken offline or be down for various reasons. We should not lose sight of what a service outage would mean in terms of our critical operations. Risk factors in this area are much more difficult to mitigate as secondary and backup services that are often implemented as part of traditional disaster recovery and business continuity are not nearly as likely to be available in the AI service realm.

Algorithmic Inertia

While not an "official" term, we think it ought to be. The idea of algorithmic inertia is that the algorithm is going to continue doing what it was told to do (including "learning") regardless of outside factors. Herein lies the danger: if you develop a predictive analysis system trained on known material, new and unknown factors could present itself to the domain that essentially renders your predictive analysis unusable, or worse, downright dangerous. A famous example here is the fall of Zillow Offers, which used AI to predict the housing markets for flipping houses. The problem emerged when the market changed unexpectedly, but the algorithm wasn't told to incorporate that change, and it kept right on predicting the market based on its "own knowledge," ultimately driving that particular business right into the ground. We should be ever-vigilant in what our AI is doing, and ready to step in at need.

Providers Everywhere!

With AI being as popular as it is and promised to be the biggest thing in technology since the creation of the internet, everyone wants a piece of the pie. As a result, there are a great many startup AI companies offering services across the spectrum: from bulk resume scanning to facial recognition to generative solutions… It seems there's any number of companies offering something. Therefore, be especially diligent when looking at these companies for potential inclusion as part of your solution. There's a very good chance these could be the weakest link in your dependency chain, and they should be closely scrutinized for risk assessment. In a pop-up minefield of lesser-funded startups attempting to reach the market quickly, we would go so far as to say that most of these companies are not likely to be enterprise-grade. Attempt to make sure the company is legit, stable, cognizant of ethics and legislation, has a good service track record, etc. All the usual, but with extreme diligence.

Appendix B: How Much Of This Whitepaper Did AI Write?

All of it? Some? None? Do you really want to know?

Actually, relatively little. However, it was in the toolset. The core of this paper is written based on research and experience, but Generative AI was a helper. For example, ChatGPT can make for an advanced search and query tool. “Give me a list of industries that use AI” and “what are some risks associated with building an AI system.”

The answers given don’t really tell us anything we didn’t already know, but can help make a nice bullet list of things and remind us of something that we might not have had explicit notes on (“oh yea, transportation logistics, I should toss that into the use case appendix”). Interestingly, the actual content returned in itself was hardly ever used directly, but in some cases serves as good inspiration for further (manually handled) topics such as the case of articulating thoughts on risk management in AI systems.

Somewhere – and I wish I remember where – I was reading about various levels of competence within a practice, and the author described a master chef as being able to perform acts of greatness, but having some difficulty articulating how they do it. A pinch of this in there, a splash of that in once it reaches a consistency kind of like this… they know, but incorporate so much subtle knowledge and decades of experience in there that it becomes difficult to really spell things out for someone at a lower level of understanding. In many ways, ChatGPT served as an excellent tool for helping to articulate some of these thoughts and experiences regarding the topic I’ve been covering. We know this and this and this. We can write a general whitepaper outline and set about filling it out. ChatGPT came into play somewhere in between, in helping to make sure the various bases were covered and providing some ideas of what might be added. As a tool for collecting thoughts and outlining various sub-topics, it saved some time. Beyond that though, quality would have suffered immensely. No silver bullet…

All that said, there are two sections that were written solely by ChatGPT. The Business Use Case Appendix is the obvious one, but in the main paper there’s a small section that was likewise generated via ChatGPT (and heavily reviewed and culled). I wonder if someone can find it? I wonder if AI could read this document and tell me which section was written by a machine…

DYMENG
TECHNOLOGY SOLUTIONS

917-336-9006
266 Paseo Reyes Drive, St. Augustine, FL 32095
info@dymeng.com | dymeng.com