gradient advisors

AI-Compass: A Framework for Identifying High-ROI AI Use Cases

PRAGMATIC AI FOR FOUNDERS & INDUSTRY LEADERS

JUNE 2024

YOUR JOURNEY TO PRAGMATIC AI

In the rapidly evolving Artificial Intelligence (AI) driven landscape, (Generative) AI vows to revolutionize businesses like never before. Despite presenting unparalleled opportunities, it also introduces intricate challenges in transforming this disruptive technology into successful business endeavors. The goal is not merely to navigate these challenges but also to elevate your organization's AI practices to attain the pinnacle of "Pragmatic AI."

We define **Pragmatic AI** as AI that translates AI efforts into **tangible business successes,** leading to **increased revenues** and **market dominance**, rather than focusing solely on model metrics.

This write-up serves the following purpose: *introduce a framework to select optimal use cases for (Gen) AI in the industry*. To the best of our knowledge, this is a one-of-its-kind attempt to introduce a detailed framework to identify AI use cases that are most likely to generate high ROI for businesses.

CONTENTS

Note from CEO'S DESK

Dear Reader, Greetings!

As we venture further into the realm of Artificial Intelligence (AI), this edition focuses on uncovering precise applications. It has become increasingly vital to identify the most effective applications for AI's integration. In our ongoing series, we dedicate ourselves to uncovering and elucidating a framework to identify the most suitable use cases for AI deployment. At **Gradient Advisors**, we recognize the pivotal role of accuracy in leveraging AI's transformative capabilities. Drawing on our extensive experience of 20 years in building core AI systems, we embark on a journey to identify and illuminate these precise application areas.

Our latest article focuses on a framework for identifying the right use cases of (Gen) AI, offering invaluable insights for founders, executives, and investors navigating the AI landscape. By honing in on these specific scenarios where AI can deliver maximum value, organizations can optimize their resources and propel innovation forward. For this reason, we call our framework AI-Compass.

Join us as we delve into the intricacies of **"Pragmatic AI"**, aimed at supporting founders, VCs, C-suite executives and the broader AI community; empowering decision-makers to harness the full potential of this groundbreaking technology. Together, let's navigate the evolving landscape of AI with clarity and purpose, driving meaningful progress and innovation in our respective industries.

We genuinely hope you'll find value in this initiative. Your feedback and engagement are highly appreciated.

Best Regards,

Anuj Gupta
Founder & CEO
Gradient Advisors

AI-Compass

A Comprehensive Framework for Identifying Impactful AI Use Cases to Build Billion $$$ Ventures

SUMMARY

- Software product development has many established frameworks for selecting optimal use cases and ideas
- One such popular framework is Impact x Effort x Competition
- However, a comparable framework tailored for AI use cases has been missing—until now.
- This article introduces AI Compass, a holistic framework to identify high-potential use cases for building AI ventures.

INTRODUCTION

It is a well-known fact that most AI projects fail to deliver any return on investment (ROI) [1]. The reasons behind this are multifaceted. *One fundamental reason is the pursuit of suboptimal, and at times entirely inappropriate, use cases.* To address this particular gap, we formally present **AI-Compass**, *a framework to identify the most suitable use cases for AI in industry.*

ICE FRAMEWORK FOR USE CASE SELECTION

- Impact x Competition x Effort (I.C.E) is a well-established framework that has long served as a reliable guide for identifying promising use cases and startup ideas.

- According to this framework, one should maximize their chances of success by focusing on crafting products & services for use cases that *offer high Impact, face limited Competition* and *demand low Effort.*

  A popular variant of I.C.E framework is *Impact vs Effort 2x2 matrix* that is used for prioritization. Here only Impact and Effort are considered.

The reason for its popularity is that it can be represented succinctly via a 2x2 grid, as shown below in Fig 1:

IMPACT / EFFORT

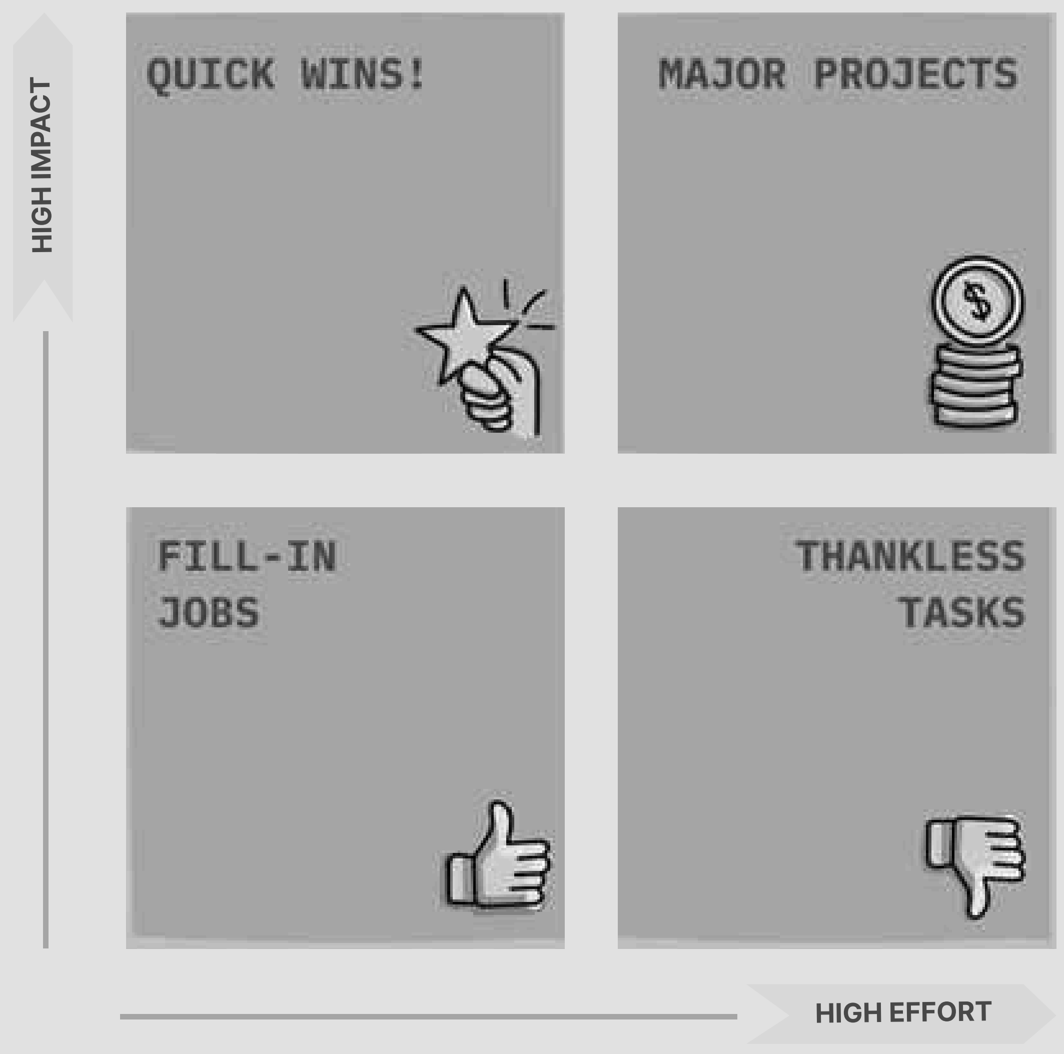


Fig 1: Impact vs Effort framework for use case selection/prioritization

WHY AI SOLUTIONS NEED A NEW FRAMEWORK

In the realm of evaluating *use cases & ideas for developing AI products/ ventures*, the I.C.E framework *proves to be grossly inadequate*. AI for industrial use cases possesses unique intricacies, necessitating the consideration of numerous additional dimensions.

While the ICE framework is adept at spotting conventional software startup ideas, it falters significantly in identifying good AI use cases.

Before moving further, it is natural for one to ask: why does the (well-established & time-tested) I.C.E framework fail to identify suitable use cases for the application of AI? Mainly for three reasons:

1. Unlike software 1.0 which is deterministic, AI solutions are stochastic. This implies AI solutions are not 100% correct and will make mistakes. This leads to bad user experience and must be handled through specialized product experience including UI, control loops & human-in-the-loop.

2. Building good AI solutions (including a very good model) is expensive process (both in terms of time and money). Most teams grossly underestimate this. Owing to this cost factor, a lot of *seemingly* straightforward use cases prove to be *economically unviable*, thereby *rendering them to be wrong choices.*

Building good AI solutions is costly and time-consuming, often grossly underestimated.

3. Owing to the specialized product experience & complexity of showing gains (ROI), the time to deploy, operationalize and test any AI system in a live environment is much longer. It is not just the simply integration of systems but also requires integrating people, processes and metrics. This often leads to drastic changes in processes & user experience along with the need for extensive instrumentation across the product for measuring the key metrics. Operationalizing AI systems requires a very careful orchestration of systems, people, processes and metrics.

Operationalizing AI systems requires a very careful orchestration of systems, people, processes & metrics

AI-Compass: AI Use Case Selection Framework

Without any further ado, the below image encapsulates the framework to identify suitable use cases for AI while building products & ventures. In addition to the three elements of Impact, Effort, and Competition from the ICE framework for software products, the AI-Compass framework includes five additional parameters. Let us understand each of these in detail:

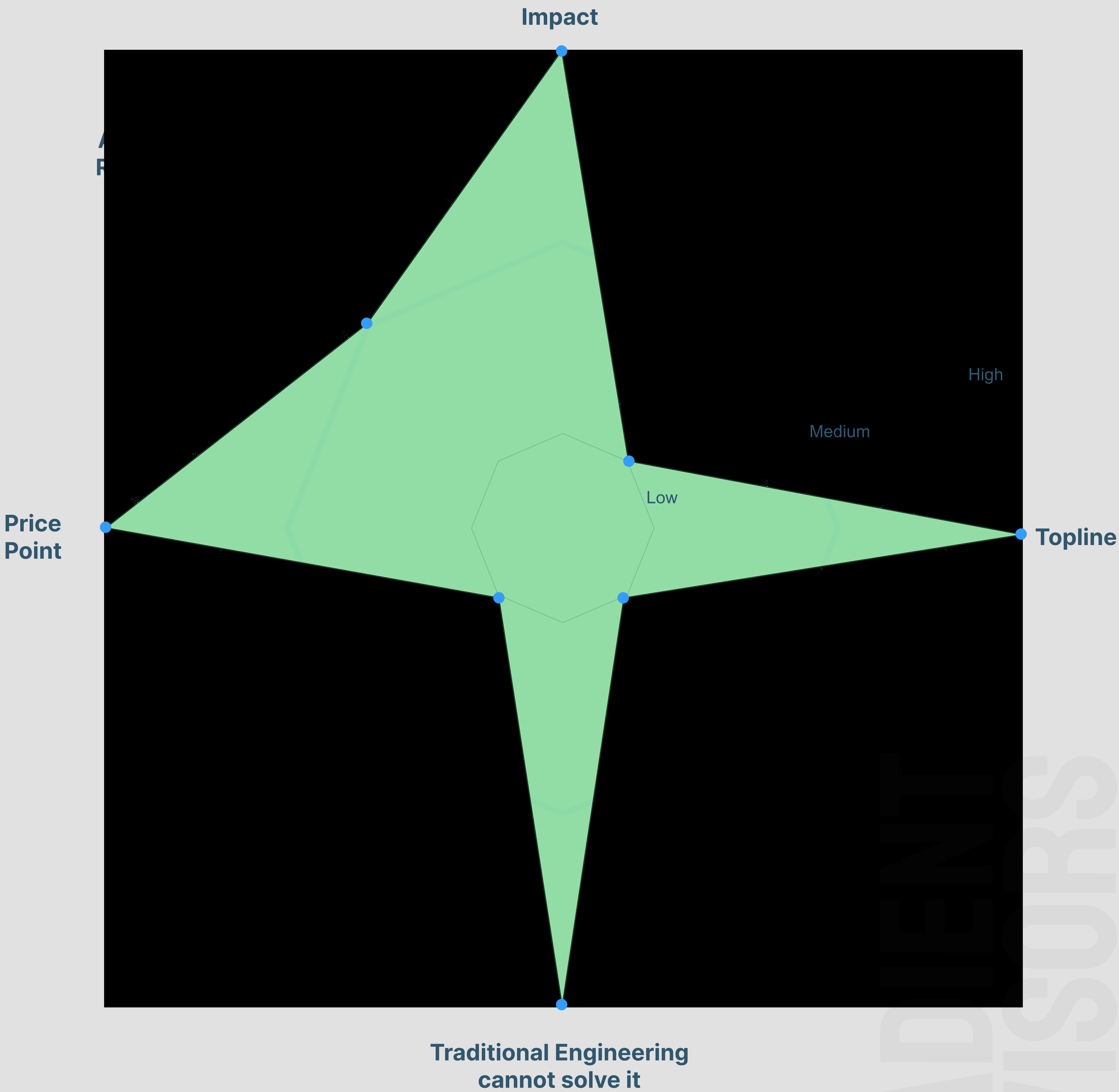


Fig 2 : AI-Compass framework

01 When Traditional Engineering Falls Short, AI as the Last Resort

In the pursuit of problem-solving, traditional engineering approaches are time-tested methodologies that have consistently delivered pragmatic and reliable solutions. However, there exist scenarios where traditional engineering methods prove inadequate, leaving a void that begs for a solution. It is in these specific circumstances AI can emerges as *the final recourse*.

AI, despite its formidable power, presents its own set of challenges. It is a costly endeavour, both in terms of resources and time, and prone to making mistakes. Therefore, it is paramount to recognize that AI should be wielded judiciously, serving as a precision tool rather than a universal sledgehammer for every challenge.

In essence, AI should be enlisted as the last resort, called upon only when all traditional engineering approaches have been exhausted.

If a problem can be solved using simple techniques, then nothing like it. AI must be used as a 'precision tool of last resort' rather than a 'universal sledgehammer for every problem'. If an NLP problem can be solved using regular expressions, one must never use LLMs! This stems from the principle of 'Occam's Razor', which says 'simple is beautiful'.

02 Availability of the right AI algorithms to effectively model & solve the problem.

Today's AI, though advanced, is far from AGI. Despite a decade of progress, AI remains an evolving field needing foundational breakthroughs. Achieving AGI could take 3, 5, or 10+ years—no one knows for sure.

For a use case to be suitable for AI, it is crucial that one must have just the right AI algorithms to model & solve the problem at hand. What do we mean by just right? It should not be under developed that it does a bad job. They should be just mature enough so that one can use them to solve the problem at hand well. For example, there is no point in trying to solve "text summarization" if one does not even have AI techniques to represent the syntax & semantics of the text. Today we are able to solve the problem because of advancement in NLP like embeddings (mathematical representation of text), Attention networks & transformer architecture. Imagine solving text summerization if the only way to represent the text had been Bag-of-Words!

At the same time, the algorithms should be mature enough that its pros n cons are well known. When building production grade AI systems, its always good to know tried n tested algorithms. Under tested algorithm can lead to severe challenges in production.

There are times when one does not have the right algorithms at hand to solve the problem. This is where one requires AI algorithm development. In the first case study that we present later on, you will see google had to develop 'Sketch-RNN' to solve the problem at hand.

We call this *AI algorithm readiness.* In the spider graph in Fig 2, this is represented on the middle layer.

Availability of *just the right* AI algorithm that solves the problem at hand very well. No more, no less is a must

03 Cost of a mistake is low

All known AI systems make mistakes, none of them is 100% correct. And they are unlikely to be 100% correct anytime soon. This is not because the AI teams that developed these systems are not up-to the mark or the training dataset is not comprehensive. It is because today's AI is stochastic. Hence making mistakes is inevitable.

Now, *if the cost of a mistake is very high, it is not the right AI problem for products & ventures.*

Applying AI systems to scenarios where the cost of mistakes is very high becomes super tricky. The (legal & financial) liability arising from one mistake could easily outweigh the (financial) gains from getting 99.9% of predictions right. Ex: self-driving cars. In 2018, Walter Huang, an apple engineer was killed when his Tesla car veered off a highway near San Francisco and ran into a crash barrier. The case dragged on for 5 years until Tesla closed it via a very hefty settlement.

On the other hand, use cases where the cost of mistakes is very low, are a great fit for applying of AI. Why? Even if you get some predictions wrong, it's ok. The end user can always retry and get the desired output! Ex: using Gen AI to create images from text descriptions. Imagine a news agency is writing an article on Mother Teresa and her work to help underprivileged kids. They use an AI system to create an image of her fighting poverty. They give the following text prompt: "create an image of Mother Teresa fighting poverty". For this text prompt, say the AI system gives them the following image as shown in Fig 3:

Fig 3: "Mother Teresa fighting poverty" as per AI

Clearly the AI system misunderstood the words "fighting" & "poverty". (The image is an actual output from a state-of-the-art AI system that creates images from text prompt). The key point is - the mistake has little or no cost. Just a bad user experience. The news agency can always go back and provide more context in the text prompt to get a suitable image.

One of the key reasons GenAI has become so popular is that for many use cases where it is applied, such as content creation, the cost of mistakes is at most "bad user experience".

It is not the case that GenAI is any *more* perfect or smarter than AI 2. It is also stochastic (like AI) and makes lots of mistakes (including ChatGPT and state of the art LLMs). Just that the cost of generating a wrong output in most use cases of GenAI is super low. The mistakes made by GenAI systems often get brushed under the carpet of the model trying to be "creative"!

It is precisely for this reason, GenAI systems have seen a rapid adoption in the world of content creation. Compare this to applying AI to healthcare in the US: imagine a patient has cancer and your AI system says otherwise. Under the US legal system this is a huge mistake that is unacceptable and has grave consequences.

AI's imperfections make high-cost mistake scenarios unsuitable. Low-cost mistake scenarios are ideal for AI utilization.

SUMMARY SO FAR

Use AI as last resort, when traditional engineering fails you	Having just the right AI algorithm to solve the problem	Cost of a mistake is low

2 GenAI is no different from AI but merely a subset of AI

04 Economics of AI

AI (at least today) is still an expensive technology - datasets, talent, compute, deploying, operationalizing and maintaining AI systems does not come cheap. At the same time, unlike research labs, the whole purpose of industry using AI is to improve their earnings. Therefore, *it is crucial that one keeps economic considerations in mind from day one*. In our 20 yrs of experience of guiding AI teams, most VCs, Founders & CXO tend to completely miss this. Turns out that the ***economic considerations have huge implications on the suitability of the use case!***

(a) Top line vs Bottom line: AI systems that contribute to the top line (increasing existing revenue, creating new revenue streams) often *tend to deliver far better Return on Investment (ROI)*. This is in stark contrast to AI systems built to improve bottom line (operational efficiency). In the later scenarios ***most AI system end up delivering far lower efficiency gains as compared to the original hypothesis/expectations.***

The core reason for this disparity is two fold:

- Developing production-grade AI systems is expensive. This cost often proves much larger than the cost of human labor it aims to automate. Owing to globalization, a lot of back office work is done in countries where the human labor cost is much lower. Beating these costs with AI is very difficult.
- All AI systems make mistakes. To maintain good & consistent user experience in the face of mistakes, one often resorts to "Human-in-the-Loop". This results in significant operational overhead which eats into margins significantly.

Owing to the above two reasons, when applying AI to improve efficiency, one starts with a hypothesis of making X% gains, one typically ends up with merely X/3 or X/5, which is way less for most scenarios. In our 20 yrs of doing AI, we have seen this play out over n over.

(b) Price point: It is crucial to quickly validate what price the market will be willing to pay for your AI solution. A lot of practitioners tend to believe that a technically superior solution will win the market hands down. One cannot build viable AI business (no matter how superior is your solution) if the market pays you merely 2x or 3x of what it costs you to generate a prediction. You will not be able to break even, forget about being profitable.

A lot of Founders & VCs believe that as their models & system will improve with the consumption of more data, then they will be able to charge much more. In reality, this never happens.

The reason is that as your models become better, improving them further proves much more expensive, both in terms of time and money [2], making it a game of diminishing returns.

AI systems targeting top line growth often outperform those aiming to improve bottom line efficiency due to high development costs, lower-than-expected efficiency gains, and challenges in pricing strategies, hindering profitability expectations.

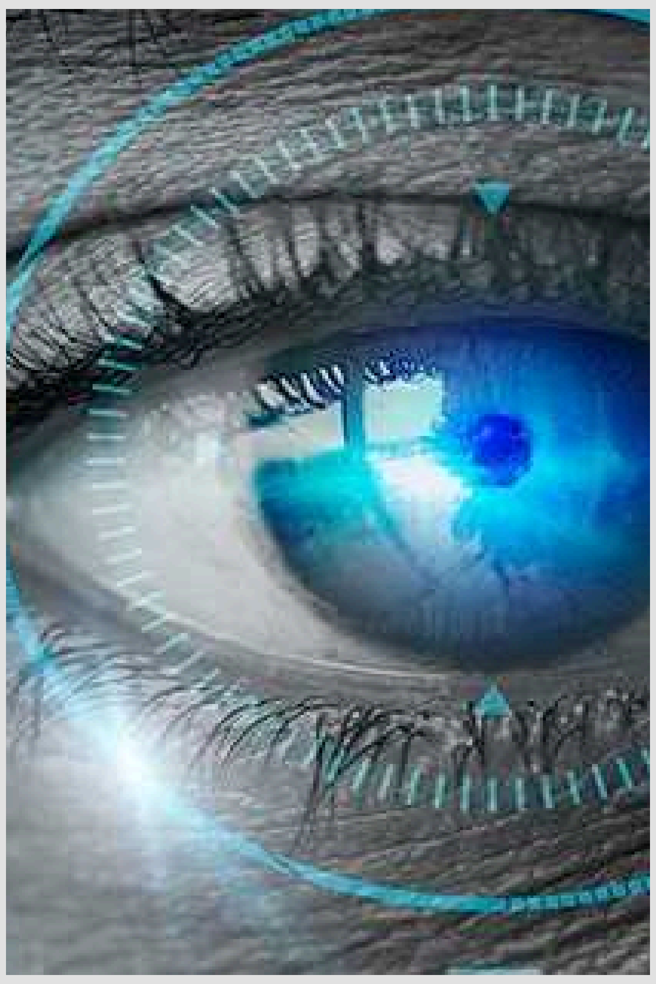
05 Ease of creating high-quality large datasets

We all know Data is the foundation of production-grade AI systems. Yet, over 85% of AI projects fail due to poor data quality [3].

One might ask, why not rely on third-party APIs or open-source models? While they're great starting points, depending solely on them is not prudent. In terms of accuracy, specialized models trained for specific problems usually outperform general-purpose ones.

Eventually, you'll need to train your own models. The key lies in creating world-class datasets—comprehensive, accurate, and large—from raw data. The first step? Building the capability to collect vast amounts of high-quality relevant raw data. Putting in a comprehensive data collection strategy in place is key.

If collecting large amounts of data for a problem is challenging, it's likely not a viable AI use case. For example, building a machine translation system for low-resource languages faces a major hurdle—gathering data to create a comprehensive dataset covering dialects, social cues, emotions, cultural norms, and humor across contexts. Similarly, developing a credit system for blue-collar workers is difficult due to their limited digital and financial footprint, making data collection a significant challenge.



There are many other factors also come into play, such as ethical concerns, fairness, data governance, interpretability in finance, safety in robotics etc. However, these are not universal to every use case—they vary by industry, application, regional regulations etc. Thus, while not part of our core framework, they remain critical considerations and must be evaluated on a case-by-case basis. For instance, if you're building an AI system to predict crimes and identify suspects before a crime occurs, ensuring it is free from racial bias is essential. Likewise, when developing autonomous cars, specialized hardware is a crucial prerequisite for a meaningful implementation.

Case Studies

To better understand the proposed framework, we now take some potential use cases, apply AI-Compass framework and show how it works

01 Quickly find and insert special characters into Google Docs/Slides/Drawings

Problem Statement: *Users of Google docs/slides once in a while use special characters (α, β, Γ etc). Given the large number of possible special characters, finding 'the' special character that the end user needs among ~2000 special characters is difficult. Design a solution to help the end users in such situations - i.e. the end users can very quickly get to 'the' special character(s) that they need. The end user needs 'that' one special character only and often cannot do with some other 'similar' special character in its place.*

Now that the problem statement is clear, before reading further, we strongly suggest you independently think on this problem.

Potential Solutions: The first solution that comes to mind is to simply do what one does for lines, shapes etc - show a widget with special characters in it. Issue is, unlike lines & shapes, the number of special characters is very large - so either create a widget that big to show all of them in one go or create a scroll experience inside that widget.

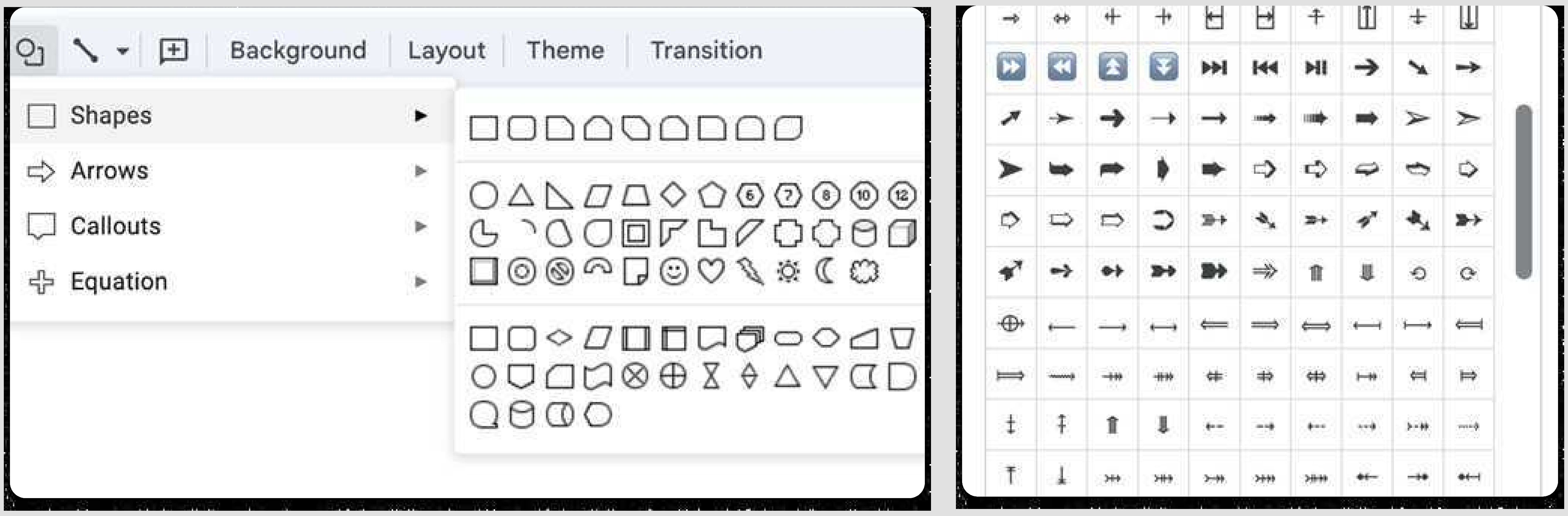


Fig 4: Widget for shapes. The one for special characters will need a scroll bar inside it

Trying to spot the special character the user needs inside a small widget with a lot of scrolling to do doesn't deliver a good user experience.

One may then think of a small widget with space only for 5-10-15 special characters, where only the most frequently/recently used ones are shown there. But what if the end user needs a character that is not in these 5-10-15 characters?

To facilitate quick accessibility, one may suggest a search bar - type the name of the special character you are looking for and we show the same/similar ones. Problem is that the traditional search fails here because *most users cannot recall the name of the special character they need*. Don't agree with this argument? Try to recall the name 20 special characters yourself! One has ~2000 special characters.

What if the end user textually describes the special character she needs? This is even more harder! (try to write the textual description of [alpha]). This renders traditional "textual search" also useless.

However, *there is a very interesting insight*: <u>users can easily "visually" describe the special character they need</u>. What do we mean by "visually' description? Users can easily draw the special character they are looking for.

So, how does this lead to a good solution? Give a scratch pad to end users, let them draw to "describe" what they are looking for. Now, the problem reduces to - can we 'visually' match the user's drawing with the shapes of various special characters. Show the 5-10 special characters that are the closest match to the user's drawing as possible suggestions.

To do this, we need an AI algorithm that can do this 'visual' matching for us. Turns out there is an algorithm that does exactly this and does it well - 'sketch-RNN'[4]. It is a neural network that understands stroke based drawings of common objects. Google used it to put together a solution:

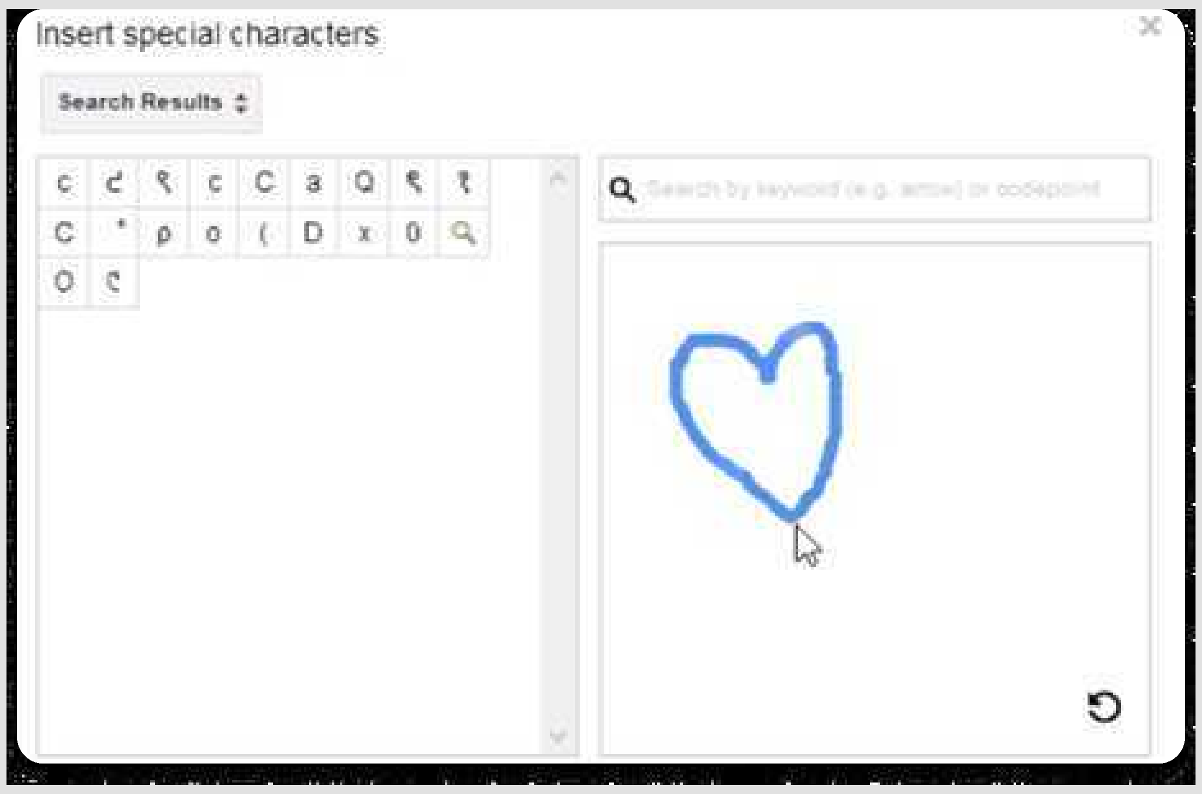


Fig 5: Widget for shapes.

Now, let us go back to our framework AI-Compass and try to use it. The impact is high -why? the lack of it spoils the experience of the end users. (We assume product metrics showed enough users getting stuck on 'finding & using' special characters. In Google's philosophy this is simply unacceptable). In this case, competition is not applicable. Having said that when this feature was launched, no one else had it including Office-365 suite by Microsoft. It only strengthens Google's image as a tech leader committed to deliver world class products & experience, giving an edge to Google slides/docs over its competition. The effort for this problem is not high, provided the core AI algorithm used is already in place.

So three out of nine points are check-mark, namely high impact, low competition, and low effort. Let us look at the remaining ones. We have already discussed why techniques like widget and textual search do not work for this problem. If you think deeply on this problem, you will realize all known traditional engineering approaches fail. Hence, *the criteria of Traditional Engineering Falling Short is also a checkmark.*

Let us now look at the point of "just the right AI algorithm for the problem at hand". At the core of the proposed solution is the ability to match the strokes of 'drawn special character' with the shape of various special characters and find the closest match(es).

Sketch- RNN [4] algorithm does precisely this. No more, no less. This is a great example of "just the right AI technology" for the problem at hand. (In Google's case, they created this algorithm).

Traditional engineering approaches fail; Sketch-RNN algorithm matches strokes of special characters, showcasing 'just the right AI technology' for the problem at hand.

Clearly the cost of mistake in this use case is super low. Further, one can easily reduce chances of wrong suggestion by suggesting 'k' closest matches rather than suggesting just the top (k=1) match. In case of failure, one can ask the user to draw slowly ensuring clear strokes.

But even then if the AI system gets it wrong, the user has a bad experience. If a lot of users have similar bad experiences multiple times, some of them might churn to MS word or other word editors. But it is not that one mistake is very expensive (compared this to self driving car driving into a barricade, or a wrong suggestion in financial markets). Both points related to economics are not applicable - Price point is not applicable since this is a free feature. Thus the question top line vs bottom line is not applicable.

In such cases, one looks to improve the end user experience - this feature is a pain killer, not a vitamin. We don't have the numbers but we are sure the product team in Google must have used product metrics to conclude a fairly high number of users trying to access special characters and struggle in the process. Without this feature, it becomes a major bottleneck for the end user to quickly find and use the right special character when they need one.

Last but not least - ease of building a dataset. sketch-RNN was introduced by Ha & Eck [4] from Google. They created QuickDraw [6], a dataset of vector drawings obtained from Quick Draw [5], an online game where the players are asked to draw objects belonging to a particular object class in less than 20 seconds. QuickDraw dataset consists of a collection of 50 million drawings across 345 classes of common objects. Each class of QuickDraw is a dataset of 70K training samples, in addition to 2500 samples in validation & test sets.

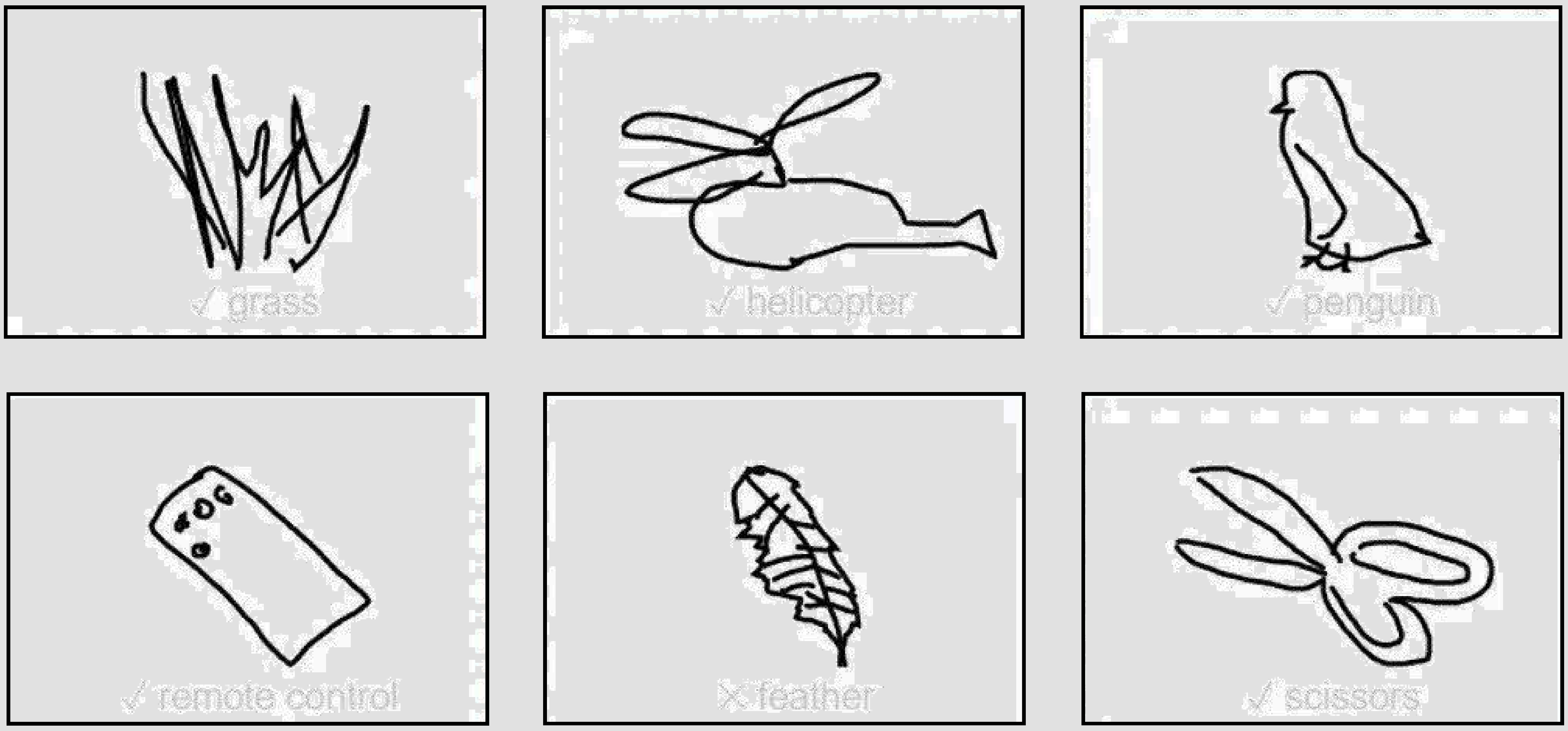


Fig 6: Widget for shapes.

The team [6] very innovatively created a secondary system (a gaming system) to collect the precise data they needed. Data collection started in early 2016, and it is only in 2017 sketch-RNN came in. The two key points to be noted:

1. They found a super innovative way to collect data from end users.
2. They started collecting data and built a very good dataset much before they started working on the modeling part.

***

02 Customer Support ChatBots

Let us now look at one of most favorite AI use cases in industry - chatbots. *It is a well known fact that no chatbot solution provider has been able to get the kind of revenues and profit margins they originally anticipated.* ***In our experience, building chatbot solution provider company was not and even today, is not the best use case for AI when building billion dollar venture.*** One of the key reasons for this is *'economics of AI' falls flat*. Let us understand this in detail.

Problem Statement: A key aspect of back office teams in any organization is customer support team. This consists of humans agents who respond to user queries & complaints over chat and phone. Since most customer support teams use predefined answers/templates to respond to user queries, it is widely believed that automating human agents is a great use case of AI to build a strong venture. Our framework does not agree with this popular belief.

One of the key realizations that has happened in last 5-7 years in the business of chatbot is that *in this space the revenue & margins are much lower as compared to the initial expectations*. We will now show the same using the above framework.

At the heart of any chatbot system is goal oriented dialogue system which tries to make sense of user query and respond accordingly. In the chatbot world, the key business metric used is *the number of queries/request the system is able to handle/intercept without going to human agents* [1].

Lets apply our AI-Compass framework to this use case. The impact is very high. Most organizations see customer support teams as a necessary evil to address customers issues and end up putting in sizable resources in it. So clearly a good solution will be a major pain killer. Competition is high, chatbot market is flooded with providers but all solutions are equally bad (good). So for a far superior solution, there is hardly any competition. On the effort front, most teams believe thats building a chatbot platform is not rocket science and can be done easily. We will revisit this point soon....

The key business metric used is the number of queries/request the system is able to handle/intercept without going to human agents

It is a well known fact that using traditional engineering approaches one cannot solve this problem. Also the cost of mistake is not very high - of course it spoils the end user experience but it is not the case that a wrong answer can cost the very existence of the company or the end user. Do we have just the right AI algorithms to solve it well - with time there have been major advances in NLP. From word embeddings, RNNs, attention networks, Transformers and now LLMs. While none of these are perfect still the tech might be good enough to get the job done.

In this case, where the narrative takes a drastic turn is when one comes to economics of AI. Let us first understand the price point. Its a well known fact that most organizations have outsourced their back office operations to APAC or South American markets for cost savings. A human agent in these markets costs atmost $300 per month. For simplicity, lets say this is $500 per month. Clearly no CFO/CIO of any organization will pay more than $500 per virtual agent unless the agent does lot more than just respond to user queries and complaints. So say, you are a chatbot company 'Bots & Company' and deploy your proprietary foundational model powered chatbot solution for client Y that has a 100 people team. So the max you can extract from Y is $500 x 100 per month.

1. In our experience, this metric does not paint the correct picture and is often misleading

Now building & deploying Deep learning solutions does not come cheap - more complex the model, more it costs. LLMs being super complex, cost a bomb. Despite there complexity, they make mistakes. To ensure a decent end user experience, you will have to bring humans-in-the-loop as well. This adds significant operational overhead and costs.

Bots & Company at the same time wishes to further improve its models. No matter how good a model your team builds, you will always have a long tail of edge cases your system will keep getting wrong. AI lives in long tail of these edge cases. More advanced the model, greater its accuracy; necessitating higher-quality and more accurate data for further enhancements. ***Instead of exponential improvement in performance, paradoxically one sees exponential increase in the expenses and efforts required for further.***

Bots & Company now faces triple whammy:

1. For their value proposition of "Replacing human agents for low end work" (improving bottom line), the price point the market is willing to pay is very low (<= $500 per agent per month)
2. High cost of deploying & maintaining deep learning models in production and cost of human-in-the-loop eats into margins significantly.
3. Improving the models further increases cost development and this is never ending because today's AI is far from 100% correct. This further dents profit margins.

Bots & Company last hope is getting to a *single superior model, with which they can serve thousands of large clients* (SaaS play). To improve models, a common technique is fine tuning. Now, it does not make sense to fine tune a model for food delivery client on the data of airline clients [1]. This means Bots & Company needs to build customized models per vertical if not per client. This completely destroys already wafer thin margins.

To get to better AI, Bots & Company will have to build & maintain specific models skyrocketing costs further. This proves to be final nail in the coffin.

It is for these reasons the chatbot market is reduced to the market of discount pricing rather than emerging market most likely to produce unicorns or decacorns.

Now imagine you were to build a solution that help your clients do better sales (improve top line). If your solution is really good, you can charge a much higher price. Hope this example clearly illustrates the point of 'top line vs bottom-line' and 'price point'.

It's important to revisit the central question: How can we quickly identify AI opportunities most likely to create unicorns? a good use case is necessary but not sufficient to create significant value.

References

[1] S. Ransbotham, S. Khodabandeh, R. Fehling, B. LaFountain, D. Kiron, "Winning With AI", MIT Sloan Management Review and Boston Consulting Group, October 2019.

[2] Anuj Gupta (Gradient Advisors), "AI CAPABILITY CONTINUUM: A Three Step Framework to Understand Return On Investment (ROI) in AI", Feb 2024

[3] New survey shows AI and ML are still nascent. https://content.alegion.com/blog/new-survey-shows-ai-and-ml-are-still-nascent

[4] David Ha and Douglas Eck, "A Neural Representation of Sketch Drawings", 2017. https://arxiv.org/abs/1704.03477

[5] J. Jongejan, H. Rowley, T. Kawashima, J. Kim, and N. Fox-Gieg. The Quick, Draw! - A.I. Experiment. https://quickdraw.withgoogle.com/, 2016.

[6] A. Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/, 2015

ABOUT US

We are the world's leading AI company, offering on-demand Chief AI Officer leadership to accelerate AI adoption and drive transformative business outcomes.

For example, a YC company partnered with us to build a critical AI system, showcased to Sam Altman (OpenAI) and Vinod Khosla (Khosla Ventures), leading to their Series B funding from Khosla Ventures. This AI system was also featured at OpenAI's recent flagship event.

With over two decades of pioneering AI expertise across startups and Fortune 50 companies, we co-create end-to-end AI solutions—spanning vision, strategy, use case identification, roadmapping, data strategy, model development, deployment, and seamless product integration. Beyond technology, we help build world-class AI teams and establish AI Centers of Excellence (CoE) for sustained success.

Don't let the future pass you by. Embrace AI with confidence under the expert guidance of Gradient Advisors—your trusted partner in unlocking AI's full potential.

For more information, do reach out to us :

hello@gradient-advisors.ai

https://gradient-advisors.ai

Gradient Advisors
904 Regency Court,
San Ramon,
California, 94582, US

Gradient Advisors
109 Wilow Road,
Enfield, EN13BP, London,
UK

Gradient Advisors
11 Charlotte Street
ON M5V 0M6
Toronto
Canada

Gradient Advisors
203, Wing E
Dubai Silicon Oasis
Industrial Area, Dubai
UAE

Gradient Advisors
W5/902, Ahad Euphoria,
Sarjapur Road, Carmelaram
Bengaluru 560035, India

Gradient Advisors
11 Els Court, Berwick
Melbourne,
Victoria 3806, Australia

GRADIENT ADVISORS

Want to see more such knowledge nuggets? Follow our social media handles