

# **AI HARMONY**



Brad Flaugher

Strategies for Blending Human Expertise  
and AI for Business Success

# **AI Harmony**

**Blending Human Expertise and AI for Business Success**

Brad Flaughier



# Draft Note

Thank you for your interest in "AI Harmony," my upcoming book on the intersection of artificial intelligence and human creativity. As a valued subscriber to the Medusa Project or a reader handpicked by Brad, I am pleased to share with you this digital draft copy of the book.

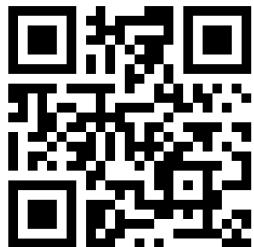
Please note that this draft is not the final version of the book and is intended for your personal use only. The final version of "AI Harmony" will be published in print and digital formats in 2023.

I kindly ask that you refrain from sharing this draft copy with others or distributing it in any way. If you wish to recommend the book to someone, please direct them to the website listed below or share my email address with them.

Thank you for your understanding and support.

Brad

[brad@bradflaughher.com](mailto:brad@bradflaughher.com)



<https://bradflaughher.com>

# Preface

Welcome to "AI Harmony: Blending Human Expertise and AI for Business Success." This book is a comprehensive guide that will help you understand how Artificial Intelligence (AI) works and how you can use it to your advantage. My goal is to provide you with a clear and unbiased understanding of AI, without the need for a technical background.

As you read through the chapters, you'll learn about how AI models are trained, with in-depth analysis of every common use case I believe is important. In chapter five, we'll delve into the specifics of AI training and explore how it can be used for business success. I've also dedicated entire chapters to the topics of autonomous weapons and self-driving cars, which I believe are important areas to consider when thinking about AI.

My hope is that this book will be useful to all readers, regardless of their role in the world of AI. Whether you're a user, developer, or investor in AI, there's something in here for you. Feel free to skip around and read the stories that interest you most - although I highly recommend the first four chapters as a must-read for everyone.

I've enjoyed writing this book and I hope you'll enjoy reading it. And if you have any notes or feedback, please don't hesitate to reach out to me at [brad@bradflaughher.com](mailto:brad@bradflaughher.com).

Remember, the goal of this book is not to be read from cover to cover, but to be used as a reference and guide for understanding AI. So take your time, read what interests you, and feel free to come back to other sections later.

Thank you for embarking on this journey with me. I hope you find this book informative, useful, and enjoyable.

*Philadelphia, PA*

*Brad Flaugher*

*March 2023*

# Contents

<b>Draft Note</b>	ii
<b>Preface</b>	iii
<b>Contents</b>	iv
<b>Chapter Summaries</b>	vii
<b>1 AI-Enabled Mass Destruction</b>	1
1.1 The AI Renaissance . . . . .	1
1.2 AI Shrinks the Market and Takes Market Share . . . . .	2
1.3 The Other Economy, the One Without AI . . . . .	3
1.4 (Human and AI) Workers of The World, Unite! . . . . .	4
1.5 An AI Sherlock Holmes? . . . . .	5
1.6 Is it The Future Yet? . . . . .	5
1.7 Key Takeaways . . . . .	6
<b>2 The End of Good Old-Fashioned Artificial Intelligence</b>	7
2.1 The AI Textbook in 1997 . . . . .	7
2.2 Explicit Rules and Codified Human Knowledge . . . . .	7
2.3 IBM Tries Every Possible Chess Move . . . . .	9
2.4 Statistical Analysis of Handwriting is the Way of the Future . . . . .	9
2.5 Less Programmer Intelligence and More Data Intelligence . . . . .	10
2.6 From Explicit Rules to a Black Box, and Beyond . . . . .	10
2.7 Key Takeaways . . . . .	11
<b>3 The Regression Theory of Everything</b>	12
3.1 Let's Avoid Knowledge Representation! . . . . .	12
3.2 A Simple Neural Network is also a Linear Regression . . . . .	13
3.3 Dummy Variables for Dummies (Wonkish) . . . . .	13
3.4 Try That Again With A Few Billion Parameters . . . . .	15
3.5 Multicolinearity and the End of Science . . . . .	16
3.6 Let's Test Some Random Inputs! Feature Importance and Explainability . . . . .	17
3.7 The Universal Machine Learning Workflow . . . . .	17
3.8 A Machine Learning Engineer is a Data Janitor . . . . .	19
3.9 Key Takeaways . . . . .	20
<b>4 Creativity and Decision Making with Deep Learning Models</b>	22
4.1 Theories of Creativity . . . . .	22
4.2 Creative Uses of Power Tools . . . . .	23
4.3 Garbage In, Garbage Out . . . . .	24
4.4 Garbage In, New Perspective Out? . . . . .	24
4.5 Concept Drift and the End of Usefulness . . . . .	25
4.6 The Impossibility of Fairness . . . . .	26
4.7 Transfer Learning Everywhere . . . . .	26
4.8 Industrial-Scale Plagiarism . . . . .	28
4.9 Working Together is Best . . . . .	28
4.10 Key Takeaways . . . . .	29

<b>5 Case Studies</b>	<b>31</b>
5.1 Introduction . . . . .	31
5.1.1 Sucker Traps to Note When Analyzing Models . . . . .	31
5.1.2 Key Terms . . . . .	32
5.2 Model Cards . . . . .	32
5.2.1 Fluid Dynamics . . . . .	32
5.2.2 Shakespearean Text . . . . .	33
5.2.3 Lithium Mining . . . . .	34
5.2.4 Chess . . . . .	35
5.2.5 Go . . . . .	37
5.2.6 Large Stock Order . . . . .	38
5.2.7 Share Tender Purchase . . . . .	39
5.2.8 Stock Trading . . . . .	40
5.2.9 Horse Racing . . . . .	41
5.2.10 Sperm Counter . . . . .	41
5.2.11 Handwriting Recognizer . . . . .	42
5.2.12 Drug Discovery . . . . .	43
5.2.13 Autism . . . . .	45
5.2.14 Online Dating . . . . .	46
5.2.15 Online Advertising . . . . .	47
5.2.16 Tennis . . . . .	48
5.2.17 Hate Speech . . . . .	49
5.2.18 Fake News . . . . .	50
5.2.19 Legal Contracts . . . . .	51
5.2.20 Facial Recognition . . . . .	53
5.2.21 Smartwatches . . . . .	54
5.2.22 Threat Detection . . . . .	55
5.2.23 University Admissions . . . . .	56
5.2.24 Credit Score . . . . .	57
5.2.25 Social Credit Score . . . . .	59
5.2.26 AGI . . . . .	60
5.3 Key Takeaways . . . . .	61
<b>6 Self-Driving with Statistics</b>	<b>63</b>
6.1 Self-Driving Horses . . . . .	63
6.2 Semi-Autonomy: A False Sense of Security . . . . .	63
6.3 Autopilot and Outsourcing Responsibility . . . . .	63
6.4 Trolley Problems . . . . .	64
6.5 Model Card . . . . .	64
6.6 Concept Drift (Reprise) . . . . .	65
6.7 Multicollinearity (Reprise) . . . . .	65
6.8 Explaining The Unexplainable in Court . . . . .	66
6.9 NASA wouldn't launch it, why should we? . . . . .	66
6.10 A Train is a Self-Driving Car, Right? . . . . .	66
<b>7 Unplugging Skynet</b>	<b>68</b>
7.1 It's a Trap! . . . . .	68
7.2 Guns Don't Kill People, People Kill People . . . . .	69
7.3 Useful Incompatibility . . . . .	69
7.4 Training Data . . . . .	69
7.5 Checks and Balances . . . . .	70

<b>8 Revolutionary for Whom?</b>	<b>71</b>
8.1 The Battle of the Assistants . . . . .	71
8.2 Employees That Are Better Than You . . . . .	71
8.3 Slow on the Uptake . . . . .	71
8.4 Free-Rider Problems . . . . .	71
8.5 When You Can't Tell The Difference . . . . .	72
8.6 Thinking Fast and Slow . . . . .	72
8.7 Dead Inside . . . . .	72
8.8 This Book is a Case Study . . . . .	72
8.9 Managing Creative Destruction in Your Career . . . . .	73
<b>9 Errors and Omissions</b>	<b>74</b>

# Chapter Summaries

## AI-enabled Mass Destruction

AI has the power to disrupt various aspects of our lives, and the democratization of AI is reducing the need for human labor, driving down the cost of creation. The adoption of AI is a gradual process that may take time to fully permeate our society, but its widespread adoption is probably unstoppable. To ensure that AI serves as a catalyst for progress, we should embrace it, create an "AI-positive" culture, and rewrite our work assignments accordingly.

## 2 The End of Good Old-Fashioned Artificial Intelligence

Avoid the term "AI" and instead differentiate between rules-based programming and deep learning to prevent confusion. Good Old-Fashioned AI (rules-based AI) is challenging to create and maintain, while modern deep learning techniques use data to train models and can be as good as the data they are trained with. Good Old-Fashioned AI is not well-suited for many complex tasks, like translation and object detection.

## 3 The Regression Theory of Everything

Deep learning models are large unscientific regression systems that map input data to output data. They are complex, deterministic and exhibit chaotic behavior, making their inner workings functionally unknowable and difficult to test. Multicollinearity and feature importance can only be understood with a high level of statistical error. Getting good data to train with is crucial for machine learning engineers to train good models. While deep learning models can have impressive and useful outputs, it's important to acknowledge their failures and limitations, which can be encouraged by users, managers and investors.

## 4 Creativity and Decision Making with Deep Learning Models

Users should consider the quality and appropriateness of the data the model is trained on and ask questions about how and when the data was collected and cleaned up. Deep learning models are not sentient creative creatures and are deterministic systems that can only generate outputs based on their training data. Users should also avoid allowing models do everything and manage the models cautiously and carefully.

## 5 Case Studies

A deep analysis of many popular machine learning models, from online dating, stock trading, threat detection and more. To ensure transparency and accountability of AI models, it is important to create and use model cards that provide information on their capabilities and limitations. Avoid "sucker traps" like biased training sets, decision-making ability, and concept drift when analyzing models. Understanding the training data and allowing the model to give low-stakes advice can also be helpful. It's crucial to manage the model's performance and not give decision-making ability to all models.

## **6 Self-Driving with Statistics**

The use of deep learning models in self-driving cars has limitations that require control and regulation. Future developments in infrastructure and control systems may support fully autonomous vehicles. Statistical models play a crucial role in the future of transportation and must be thoroughly understood to ensure safe and effective self-driving cars.

## **7 Unplugging Skynet**

As we approach the artificial intelligence revolution, people's fears of machines taking over the world are largely unfounded, and the more immediate risks of economic disruption should be focused on. Intelligence does not generate a drive for domination, and survival instinct is not hardwired into AI. The real risk is AI leading to further increases in wealth and income inequalities unless new policies are implemented.

## **8 Revolutionary for Whom?**

The winners and losers (or the shaken and unaffected) in the AI economy are analyzed, and the chapter provides a framework for understanding the evolution of daily tasks and careers over time. It stresses the importance of governments, businesses, and individuals taking action to ensure that AI's benefits are distributed equitably and its potential negative consequences are addressed. The chapter's analysis goes beyond automation and highlights the need for mitigating the impact of AI on women, minorities, and low-wage workers.

# AI-Enabled Mass Destruction

*There is little doubt that AI will contribute to profound transformations over the next decades. At its best, the technology has the potential to release us from mundane work and create a utopia in which all time is leisure time. At its worst, World War III might be fought by armies of superintelligent robots. But they won't be led by HAL, Skynet or their newer AI relatives. Even in the worst case, the robots will remain under our command, and we will have only ourselves to blame.*

Anthony Zador, Yann LeCun 2019 [dontfearterminator]

## 1.1 The AI Renaissance

Technology often finds its first adopters among those who can afford it and foresee its potential. Artificial Intelligence (AI) is no exception, with its early applications dating back several decades, notably in the financial industry. Hedge funds and investment banks identified the value of AI and started using it for trading and portfolio management.

Renaissance Technologies, a hedge fund founded by mathematician James Simons, exemplifies the early adoption of AI. Since 1978 the company has used AI to predict market movements and optimize trading strategies. By employing highly skilled PhDs known as "quants" to develop algorithms and AI models, Renaissance Technologies has become one of the most successful hedge funds in history.

As AI gained traction, tech giants like Google and Microsoft began investing in their own quants. These experts were tasked with using AI to address a wide range of challenges, from refining search algorithms to enhancing language processing capabilities. Much of this research was made public, allowing the broader community to benefit from these advancements. AI-enabled products became visible to more consumers, and AI tooling became very accessible to developers. Online advertising became a highly visible industry where AI is able to analyze user behavior and target ads with remarkable precision.

The accessibility of AI technology has increased, thanks in part to open-source projects that enable individuals and smaller organizations to tap into its power. This democratization of AI has ignited a wave of innovation, with developers worldwide contributing to the AI renaissance. Developments in AI have turned a corner; large open-source datasets coupled with cheap computing power and powerful open-source development tools have taken AI out of the realm of expensive private enterprise into a new era, where companies give away large and valuable models under strong public licences (like Stable Diffusion from StabilityAI, LLaMA from facebook) and the most innovative companies operate as large non-profit or limited-profit entities (like OpenAI, the creators of ChatGPT).

1.1 The AI Renaissance . . . . .	1
1.2 AI Shrinks the Market and Takes Market Share . . . . .	2
1.3 The Other Economy, the One Without AI . . . . .	3
1.4 (Human and AI) Workers of The World, Unite! . . . . .	4
1.5 An AI Sherlock Holmes? . . . . .	5
1.6 Is it The Future Yet? . . . . .	5
1.7 Key Takeaways . . . . .	6



**Figure 1.1:** "Portrait of a beautiful young woman with purple hair looking into the camera. Associated press. Photojournalism. Street photography." made with Stable Diffusion 2.1 try it yourself at <https://stability.ai/>

The world of open AI acknowledges that training data is valuable, but realizes that especially on the internet, information wants to be free. The risks of data theft and unauthorized use have escalating, and the industry is simultaneously having its "iPhone moment" with the advent of ChatGPT, and its "Napster Moment" with the public release of large generative models trained on copyrighted content[[copyrighteconomist](#)]. Companies and individuals are striving to protect their intellectual property while others look for ways to exploit it, but if the past is a guide copyright holders are likely to lose out, as the ability to capture income from copywritten content moves from content creators and to content distributors, namely large cloud computing companies that run AI models at scale.

This situation sets the stage for an AI copyright battle looming on the horizon. As the distinction between proprietary and open-source information blurs, the question of who owns AI-generated content becomes increasingly important. Governments, corporations, and individuals are all grappling with the legal and ethical implications of AI and its widespread adoption. When one considers that AI models themselves are content that can be modeled and stolen, one can get a glimpse of the full destructive and transformative power of AI. AI doesn't just allow you to benefit from copywritten text and images and create your own that you can distribute, AI also allows you to copy the multibillion dollar models that researchers at Google, Microsoft and Facebook have made and modify and distribute them as long as you have the technical know-how and a big enough computer to run the models. The AI renaissance is breaking traditional business models and also breaking the business models of its creators almost as soon as it is released. Like many transformative technologies, AI is shrinking the market, making things cheap for everyone, and capturing market share.

[copyrighteconomist](#)



Figure 1.2: "a delicious cheeseburger made with waffles instead of a bun and covered with raspberry jam" made with Stable Diffusion 2.1

## 1.2 AI Shrinks the Market and Takes Market Share

The "Software Paradox" posits that as software becomes more open, it tends to shrink markets while capturing most of the market share. This phenomenon can be observed in the current state of AI, particularly with the advent of open-source AI. As AI technologies become more sophisticated and widespread, they are reducing the need for human labor in many markets, and bring the cost of creation down significantly across industries.[[OGrady2015](#)]

One key aspect of AI's impact on the market is its ability to automate repetitive and mundane tasks, allowing humans to focus on more complex and creative aspects of their work. This not only streamlines processes but also has the potential to improve the overall quality of work. With AI handling the less desirable tasks, human workers can concentrate on utilizing their unique skills and expertise, resulting in better outcomes and higher value-added work.

The rise of open-source AI further accelerates this trend, making advanced AI tools and algorithms accessible to a broader range of individuals and

[OGrady2015](#)

organizations. As these powerful tools become more widely available, businesses across various industries will find it increasingly difficult to justify the cost of employing human workers for tasks that can be more efficiently completed by AI. This shift will result in a smaller job market for those specific tasks, with AI taking the majority of the market share.

However, the shrinking of specific labor markets does not necessarily mean the eradication of all human work. Instead, it presents an opportunity for the workforce to adapt and focus on roles that are complementary to AI systems. This could involve tasks that require empathy, complex problem-solving, or human intuition, which are areas where AI currently struggles to excel.

### 1.3 The Other Economy, the One Without AI

As AI revolutionizes various sectors of the economy, there's another economy that remains untouched by AI's transformative effects. This other economy is characterized by heavily regulated industries that often resist technological innovation, resulting in stagnating growth and rising costs for consumers[andreesenunemp].

andreesenunemp

In many places, AI is practically illegal. This is partly due to consumers, professional organizations and governments struggling to keep up with the rapid pace of technological advancements, leading to a lack of understanding and appropriate regulations [lawmakersny]. Additionally, public opinion remains skeptical about the widespread adoption of AI, as evidenced by polls[morningconsult][monmouthai]. This skepticism is not helped by stories of software developers becoming emotionally attached to AI chatbots[googlenerd].

lawmakersny

morningconsult

monmouthai

googlenerd

The contrast between sectors that embrace AI and those that resist it highlights a growing divide in the economy. In industries where AI is allowed to flourish, technological innovation drives down prices and increases product quality, leading to a more efficient market. However, heavily regulated sectors that resist AI adoption experience rising costs and stagnating growth, ultimately consuming a larger share of the economy.

This phenomenon is exacerbated by the emotional interplay between production and consumption. As consumers, we tend to become frustrated with price increases in the heavily regulated sectors. On the other hand, as producers, we may feel threatened by technological disruption in industries that embrace AI. This dichotomy demonstrates the inherent conflict between the desire for stability in the economy and the need for innovation and progress.

As the regulated, non-technological sectors continue to grow as a percentage of GDP, the economy may ultimately become dominated by these stagnating industries. In such a scenario, the full potential of AI and other advanced technologies may never be realized. This highlights the importance of addressing regulatory barriers, fostering public understanding,

and promoting AI adoption in industries that have yet to embrace its potential.

Undoubtedly, individuals seeking to leverage AI for their professional tasks will do so, and it is crucial to establish appropriate controls and governance in anticipation of this shift. As high school assignments across the nation undergo transformation, so too will the nature of our work. Fortunately for humankind, many of these assignments that are being restructured were inherently flawed to begin with. In the era of virtual communication, we identified gatherings that could have been condensed into emails. Now, we are poised to uncover occupations that could have been efficiently managed through simple prompts given to an AI.

By fostering an "AI-positive" culture of collaboration and transparency, we can ensure that AI serves as a catalyst for progress, rather than a hindrance. Ultimately, the key to unlocking AI's full potential lies in our ability to navigate the delicate balance between innovation and regulation, embracing the opportunities for growth while addressing the inherent risks and concerns that come with this new era of technological advancement.

## 1.4 (Human and AI) Workers of The World, Unite!

AI's adoption can be seen as an extension of the outsourcing trend that has been prevalent for decades. The principle "if you can't beat them, join them" seems more relevant than ever, as people increasingly integrate AI into their work and lives.

Even in places where AI is technically illegal or frowned upon, individuals find creative ways to leverage AI to make their lives easier. Students may use AI to help write essays, office workers may employ Optical Character Recognition (OCR) to avoid tedious data entry tasks, and programmers might utilize GitHub Copilot regardless of whether their employers approve. These examples demonstrate that as AI tools become more accessible and affordable, they will infiltrate various industries.

Instead of resisting this trend, we should work with it and guide its development. Embracing AI and integrating it into our work processes can lead to increased productivity, innovation, and overall improvement in our quality of life[peopleusingAIhappy].

The situation with AI usage is reminiscent of self-driving cars and semi-autonomous driving modes. Despite manufacturers' intentions and guidelines, these features are often treated as fully autonomous by users, reflecting the human tendency to push boundaries and adapt technology to suit their needs.

As AI becomes more ingrained in our daily lives, it's essential to recognize the inevitability of its adoption and focus on guiding its development in a way that maximizes its benefits and minimizes potential risks. By uniting human and AI workers, we can foster a collaborative environment that leverages the strengths of both entities and paves the way for a more efficient, innovative, and prosperous future. [protopia]



Figure 1.3: "mdjrny-v4 style a propaganda poster that says 'AI and Human Workers of the World Unite' featuring some robots and humans laboring together in the fields, USSR-style 8k" made with Mann-E

peopleusingAIhappy

protopia

If we learn any lessons from early adopters of AI in finance, we'll learn that humans role has changed, but not disappeared. *"There was a time when everyone thought the quants had figured it out. That is not the perception today. When it comes to the stockmarket, at least, automation has not been the winner-takes-all event that many fear elsewhere. It is more like a tug-of-war between humans and machines. And though the machines are winning, humans have not let go just yet."* [finaieconomist]

finaieconomist

## 1.5 An AI Sherlock Holmes?

In an intimate conversation around the proverbial fireplace in 2023, NVIDIA's CEO, Jensen Huang, engaged in a spirited dialogue with Ilya Sutskever, a luminary in the realm of artificial intelligence and co-creator of ChatGPT. Sutskever likened the advancements in generative AI to handing the machine an enigmatic mystery tale, challenging it to decipher the climactic revelation: "the killer is... [blank]." He postulated that an AI adept at unearthing the final word of the whodunit may possess some semblance of comprehension and reasoning, effectively "compressing" its understanding of the world.

In this literary endeavor, we shall embark on an in-depth journey to discern the extent of AI's capacity for reasoning, the limits of its abilities, the intricacies of its creation, and the judicious application of its power. Grasping the answers to these pivotal inquiries is paramount to appreciating the role AI ought to play in our existence. Should we perceive AI as merely a "blurry JPEG of the Web," [newyorkerChatGPTBlurry] and if ChatGPT is doing any compression of its knowledge what is lost? and what should humans add back before making decisions? Should we worry? Or should we envision AI as a digital Sherlock Holmes, wielding the formidable power of boundless deduction?

newyorkerChatGPTBlurry

## 1.6 Is it The Future Yet?

AI, as a general-purpose technology, has the potential to transform various aspects of our lives. However, its widespread adoption and impact on Total Factor Productivity (TFP) might not happen overnight. History has shown that even groundbreaking general-purpose technologies can take time to reach their full potential.

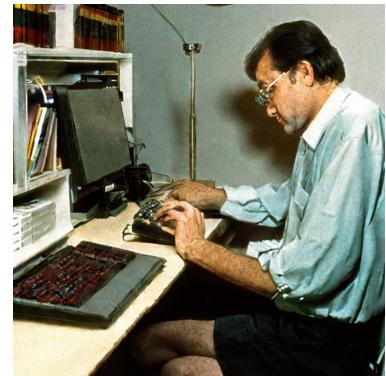
Take electricity, for example. Although Thomas Edison invented the light bulb in 1879, it took several decades for electricity to become the primary power source in industries and households. The infrastructure required to generate and distribute electricity on a large scale was built gradually, and businesses needed time to adapt their processes and machinery to leverage this new energy source. It wasn't until the early 20th century that the true impact of electricity on productivity and economic growth was realized.

Another example is the steam engine, invented by James Watt in 1775, which marked the beginning of the Industrial Revolution. The technology's full

potential was not realized until several decades later. The adoption of steam-powered machinery required significant investments in infrastructure and the reorganization of production processes. Additionally, the development of railway networks and steamships expanded the reach of this technology, leading to a profound impact on productivity and global trade.

The internet also followed a similar pattern. While the internet was conceived in the late 1960s, it wasn't until the 1990s that its commercial potential began to be explored. The widespread adoption of the internet required the development of user-friendly web browsers, the expansion of telecommunications infrastructure, and the emergence of e-commerce platforms. It took several years for the internet to become an integral part of our daily lives and contribute to increased productivity across various industries.

So, while AI has the potential to revolutionize numerous aspects of our lives, it may take time for this technology to fully permeate our society and yield its maximum benefits. Patience and continuous innovation will be essential in realizing the transformative potential of AI. By learning from the history of general-purpose technologies, we can better understand the trajectory AI might follow and work towards a future where it significantly impacts productivity and our everyday lives.



**Figure 1.4:** "a man at his computer in 1991 trying to use microsoft excel. MIT Archive." made with Stable Diffusion 2.1

## 1.7 Key Takeaways

- ▶ **AI is a general-purpose technology that possesses the potential to disrupt various aspects of our lives, including the job market and the economy.** This power to creatively destroy should not be underestimated as it can bring about significant transformations.
- ▶ **The democratization of AI has effectively reduced the cost of creation close to zero,** open-source developers worldwide are contributing to the AI revolution. The rise of open-source AI is gradually reducing the need for human labor in many markets, thereby driving down the cost of creation significantly across various industries.
- ▶ **The adoption of AI is not a new phenomenon; it is merely an extension of the outsourcing trend that has been prevalent for decades.** The shift towards automation has been a gradual process and is unlikely to happen overnight.
- ▶ **The pace of technological change is often slower than what the news may suggest.** While AI has the potential to revolutionize numerous aspects of our lives, it may take time for this technology to fully permeate our society and yield its maximum benefits.
- ▶ **AI's widespread adoption is probably unstoppable, given its affordability and its ability to operate stealthily.** Rather than resisting this shift, we must embrace it and work to determine how humans can best fit into this new era. By creating an "AI-positive" culture of collaboration and transparency, we can ensure that AI serves as a catalyst for progress. Just like in finance, where machines have "taken over," there is still room for human traders, risk managers, etc. In the same way, we should now assume that every worker and student will use AI going forward and rewrite our assignments accordingly.

# The End of Good Old-Fashioned Artificial Intelligence

2

*"Programming will be obsolete. I believe the conventional idea of "writing a program" is headed for extinction, and indeed, for all but very specialized applications, most software, as we know it, will be replaced by AI systems that are trained rather than programmed. In situations where one needs a "simple" program (after all, not everything should require a model of hundreds of billions of parameters running on a cluster of GPUs), those programs will, themselves, be generated by an AI rather than coded by hand."* Matt Welsh, 2023 [welsh2023]

## 2.1 The AI Textbook in 1997

Dr. Elaine Rich's textbook on artificial intelligence (AI), published in the 1980s, was a groundbreaking work that helped to establish many of the core concepts and techniques in the field of AI. However, the rapid advancements in AI over the past few decades have led to many of the chapters in this textbook becoming obsolete.

One of the main reasons for this is the prevalence of deep learning, big data, and large-scale statistical models in modern AI. These techniques have largely replaced the symbolic, rule-based approach to AI that was emphasized in the textbook, making many of the chapters on knowledge representation and expert systems less relevant.

Additionally, the explosion of data and the availability of powerful computing resources have made it possible to apply machine learning techniques at a scale that was previously unimaginable. This has led to the development of highly effective machine learning models that can handle complex tasks such as image and speech recognition with a high degree of accuracy, making many of the chapters on simpler machine learning techniques such as decision trees<sup>1</sup> and linear regression less relevant. [rich\_2009]<sup>2</sup>

We'll discuss this history and a few examples from the "early days" of AI to help us understand where we are headed. We'll start with machine translation, then discuss chess and finally neural networks, which will be the focus of the rest of this book.

## 2.2 Explicit Rules and Codified Human Knowledge

When we "teach" a computer to perform a task by explicitly writing down all of the rules of that task, we are really codifying human understanding.<sup>3</sup> When we codify human understanding we write down every rule that we know explicitly. For small tasks we can do this with 100 percent accuracy, and only minor headache on the part of the software developer.

2.1 The AI Textbook in 1997 . . . . .	7
2.2 Explicit Rules and Codified Human Knowledge . . . . .	7
2.3 IBM Tries Every Possible Chess Move . . . . .	9
2.4 Statistical Analysis of Handwriting is the Way of the Future . . . . .	9
2.5 Less Programmer Intelligence and More Data Intelligence . . . . .	10
2.6 From Explicit Rules to a Black Box, and Beyond . . . . .	10
2.7 Key Takeaways . . . . .	11

1: Although mathematically neural networks are decision trees <https://arxiv.org/abs/2210.05189>

rich\_2009

2: the book is now in its third edition and unlikely to be updated as Dr. Rich as retired <https://www.cs.utexas.edu/~ear/>

3: Programming this way makes some software development totally boring, I almost switched my major in college to math after considering what a life would look like manually writing rules for handling "edge cases" for the rest of my natural life.

For example, let's write a boring function to tell you the number of days for a given month.

```
def days_in_month(year, month):
    if month in [1, 3, 5, 7, 8, 10, 12]:
        return 31
    elif month in [4, 6, 9, 11]:
        return 30
    elif month == 2:
        if (year % 4 == 0 and year % 100 != 0) or year % 400 == 0:
            return 29
        else:
            return 28
    else:
        return "Invalid month"
```

Writing code can be a tedious and repetitive task, especially when it comes to debugging and testing. It can be especially frustrating when you're working on a large project and you're trying to track down a specific bug that's causing the program to crash. Testing code can also be boring, as it often involves running the same tests over and over again to ensure that the code is working correctly.

Additionally, writing code can be boring because it requires a lot of concentration and focus. It can be easy to get lost in the details and lose track of time, especially if you're working on a complex problem. It can also be challenging to come up with creative solutions to problems, and it can be frustrating when your code doesn't work as expected.

While writing and testing code can be rewarding and fulfilling, it can also be a tedious and boring process. It requires a lot of patience, persistence, and attention to detail, and it can be easy to get frustrated and lose motivation. However, with practice and perseverance, it is possible to overcome these challenges and find enjoyment in the process of writing and testing code.

AI has traditionally operated by explicitly codifying human knowledge into machine-readable formats by doing the boring job of coding. This approach, which I'm calling "codified human knowledge" relies on humans to carefully structure and organize information in a way that can be understood by the AI system. The AI system then uses this structured knowledge to make decisions and perform tasks.<sup>4</sup>

However, recent advances in AI have largely ignored the knowledge representation problem and instead have focused on using statistical techniques and neural networks to automatically learn patterns and relationships in data. This approach, known as "deep learning," involves training large neural networks on vast amounts of data, allowing the AI system to make educated classifications and transformations of data without explicit human guidance.

Deep learning has proven to be highly effective in a variety of applications, such as image and speech recognition, and has contributed to the rapid progress we have seen in AI in recent years. However, the reliance on large amounts of data and the lack of transparency in these models can make



Figure 2.1: "a frustrated programmer writing boring rules on his computer" made with Stable Diffusion 2.1

4: Some smart people think that AI spells the "End of Programming" <https://cacm.acm.org/magazines/2023/1/267976-the-end-of-programming/fulltext>, I think the headline is clickbait but the general idea that programming is changing along with advances in AI is true. We are all data scientists now!

it difficult to understand how they are making decisions, which can be a concern in certain applications (hence the title of this book).

## 2.3 IBM Tries Every Possible Chess Move

Deep Blue was a revolutionary computer developed by IBM that was specifically designed to play chess at the highest level. It was programmed with a vast database of chess knowledge and was able to analyze millions of positions per second.

Garry Kasparov was the reigning world chess champion at the time, and he was considered to be one of the greatest players in history. He was beaten by Deep Blue, which used rules based GOFAI (Good Old Fashioned AI) to essentially calculate every possible move and project that move to the end of the game, then choose the best position by brute force. Despite Kasparov's best efforts, he was no match for Deep Blue's computational power. In the end, Deep Blue emerged victorious, defeating Kasparov in a historic match that changed the world of chess forever.

Deep Blue was a turning point in the development of AI, but Deep Blue's methods (namely calculating every possible outcome of a Chess game to determine the best move) was not suitable for many of the world's problems. It turns out that Chess is fun, but the world is not like chess.<sup>5</sup> The "real"<sup>6</sup> future of AI was being developed elsewhere, using statistics and a toy model of the brain to solve a very practical problem for banks.

## 2.4 Statistical Analysis of Handwriting is the Way of the Future

It was the early 1990s and Yann LeCun was a researcher at Bell Labs in New Jersey. At the time, the process of reading and processing checks was a tedious and time-consuming task that was done manually by bank employees. LeCun saw the potential for using artificial intelligence to automate this process, and he began experimenting with using convolutional neural networks (CNNs) to recognize patterns in images of checks.

At the time, CNNs were a relatively new type of neural network that had been developed in the 1980s for image recognition tasks. They were inspired by the structure of the human visual system, and were able to process images in a way that was similar to how the human brain does.

LeCun's work was groundbreaking, and he was able to achieve impressive results using CNNs to process checks. By 1993, he had developed a system that was able to read and process checks with a high degree of accuracy, significantly reducing the amount of time and effort that was required to process checks manually.<sup>7</sup>

LeCun's work on using CNNs for check processing was a major milestone in the field of artificial intelligence, and it laid the foundation for the



Figure 2.2: "Garry Kasparov setting a computer on fire" made with Stable Diffusion 2.1

5: if you would like to read an example of the simplest chess engine that I could imagine that is written in a similar "codified" way as Deep Blue check out <https://github.com/thomasahle/sunfish/blob/master/sunfish.py>

6: We might go back, and try again to more explicitly code everything up, and in some cases we still need to, but from the author's perspective, we live in a deep learning/neural network world.

7: CNN digit OCR models are frequently featured in beginner training tutorials for deep learning libraries like PyTorch and Tensorflow, check one out [https://github.com/jonkrohn/DLTFpT/blob/master/notebooks/alexnet\\_in\\_tensorflow.ipynb](https://github.com/jonkrohn/DLTFpT/blob/master/notebooks/alexnet_in_tensorflow.ipynb)

development of many other applications of CNNs in the years that followed. Today, CNNs are widely used in a variety of applications, including facial recognition, image classification, and natural language processing.<sup>8</sup>

## 2.5 Less Programmer Intelligence and More Data Intelligence

I think it's useful to separate the knowledge in the AI problem-space into two groups. The data and the programmer together make the programs that we use every day, and for the rest of this book I'll try and separate the discussion of the smarts of each to help us better understand the world.<sup>9</sup>

Early AI relied heavily on a human programmer to design, write, and debug computer programs. Good programmers needed domain expertise, problem-solving skills, logical thinking, and the ability to learn and adapt to new programming languages and technologies.

We are now in the age of big data, and everyone knows that "data is gold". Statistical AI methods that are now most prevalent rely on extracting meaningful insights and knowledge from large datasets. This involves using statistical and analytical methods to discover patterns and trends in data, and using this information to inform business decisions or solve problems.

Throughout this book I'll discuss the interaction between programmer and data, and what can go wrong. Working with big data and statistics at a large scale has given AI tremendous ability, but has made understanding and testing models infinitely more difficult. It is your authors belief that understanding the nuance of this interaction between programmer and data is essential to understanding modern AI.

## 2.6 From Explicit Rules to a Black Box, and Beyond

Artificial intelligence (AI) has come a long way since its inception, and the way that it makes decisions has changed significantly over time. In the early days of AI, explicit rules were used to tell the AI system what to do in certain situations. These rules were often written by humans and encoded into the system, and the AI would follow them to make decisions.

However, with the advent of deep learning, we have started to rely on a statistical understanding of the truth for AI to make decisions. Deep learning is a type of machine learning that involves training artificial neural networks on large datasets. These neural networks are able to learn patterns and relationships in the data, and can use this knowledge to make decisions.

The use of deep learning has led to the development of powerful AI that is able to perform tasks that were previously thought to be impossible for a

8: check out Yann LeCun demonstrating a convolutional neural network in 1993 at [https://www.youtube.com/watch?v=FwFduRA\\_L6Q](https://www.youtube.com/watch?v=FwFduRA_L6Q).



Figure 2.3: "a group of computer programmers striking outside of Microsoft's offices with placards saying 'rule-based programming is boring'" made with Dall-E 2

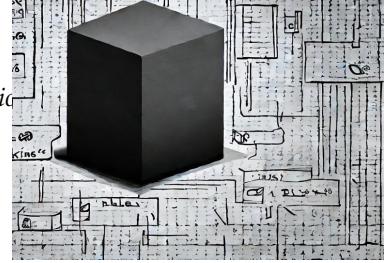
machine to handle. For example, deep learning has led to the development of AI systems that are able to recognize faces, translate languages, and even beat humans at complex games like chess and Go.

While deep learning has led to significant advances in AI, it has also made it harder to debug and understand how the AI system is making its decisions. With explicit rules, it was relatively easy to understand why the AI made a particular decision. However, with deep learning, it is often difficult to understand exactly how the AI arrived at its decision. This can make it challenging to troubleshoot problems with the AI system and to ensure that it is making decisions that are fair and unbiased.

AI has come a long way since its early days, and the way that it makes decisions has changed significantly over time. While explicit rules were once used to tell the AI what to do, we now rely on a statistical understanding of the truth for AI to make decisions. This has led to the development of powerful AI that is able to perform a wide range of tasks, but it has also made it harder to debug and understand how the AI is making its decisions.<sup>10</sup>

## 2.7 Key Takeaways

- ▶ **AI is a misleading term**, we should instead talk about rules-based programming (GOFAI) and deep learning so we don't confuse ourselves, our partners and our users.<sup>11</sup>
- ▶ **Good Old-Fashioned AI (rules-based AI) is hard to create and maintain.** We used to use it for chess engines (like Deep Blue) and translation, but now programmers favor using statistical machine learning techniques.
- ▶ **Good Old-Fashioned AI is not well suited to many problems**, like machine translation and handwriting detection.
- ▶ **Modern Deep Learning techniques use data to train models instead of humans explicitly writing rules**, and the **deep learning models are often as good as the data they are trained with**.<sup>12</sup>



**Figure 2.4:** "from explicit rules, to a black box and beyond" made with Stable Diffusion 2.1

10: From here on out I'll try and use "GOFAI" (Good Old-Fashioned Artificial Intelligence) to describe the rules-based approach, and use "Deep Learning" or "Machine Learning" to talk about modern neural network and statistics-based techniques

11: Avoid the imprecise words "Algorithms" and "AI" and instead consider the more precise "rules-based AI" AKA "GOFAI" to describe situations where programmers explicitly write rules that they design and "deep learning" or "machine learning" where programmers create models using a dataset and let the machine figure out the rules via a neural network.

12: More *intelligent* data and maybe less intelligent programmers, or maybe programmers who are better trained in statistics and complex systems theory instead of "classical" computer science.

# The Regression Theory of Everything

# 3

*"AI Scientists disagree as to whether these language networks possess true knowledge or are just mimicking humans by remembering the statistics of millions of words. I don't believe any kind of deep learning network will achieve the goal of AGI [Artificial General Intelligence] if the network doesn't model the world the way the brain does. Deep learning networks work well, but not because they solved the knowledge representation problem. They work well because they avoided it completely, relying on statistics and lots of data instead. How deep learning networks work is clever, their performance impressive, and they are commercially valuable. I am only pointing out that they don't possess knowledge and, therefore, are not on the path to having the ability of a five-year-old child." Jeff Hawkins, 2022 [hawkins2022]*

## 3.1 Let's Avoid Knowledge Representation!

The knowledge representation problem in AI is the challenge of how to formally represent knowledge in a way that a computer can understand and reason about. This typically involves creating a set of symbols, rules, and structures that can be used to represent concepts, relationships, and other types of information. The goal is to create a representation that is both expressive enough to capture all relevant aspects of the domain, and computationally tractable enough to allow for efficient reasoning and inference. There are many different approaches to knowledge representation, including logic-based, semantic networks, frames, and ontologies, each with their own strengths and weaknesses.

Deep learning techniques handle knowledge representation differently than traditional symbolic AI methods. Unlike symbolic AI, which relies on explicit and hand-coded representations of knowledge, deep learning techniques learn to represent knowledge implicitly through the use of neural networks.

In deep learning, knowledge is represented in the form of the weights of the neural network. These weights are learned through training on a large dataset and they capture the underlying relationships and patterns in the data. The neural network can then use these learned weights to make predictions, classifications or generate new data.

Deep learning models can handle large and complex datasets, and can automatically extract features from the data without the need for manual feature engineering. This makes them particularly well-suited for tasks such as image and speech recognition, natural language processing, and other areas where large amounts of data are available. However, they are not as good at explicating how they arrived at a decision, which can be a disadvantage.

3.1 Let's Avoid Knowledge Representation! . . . . .	12
3.2 A Simple Neural Network is also a Linear Regression . . . . .	13
3.3 Dummy Variables for Dummies (Wonkish) . . . . .	13
3.4 Try That Again With A Few Billion Parameters . . . . .	15
3.5 Multicollinearity and the End of Science . . . . .	16
3.6 Let's Test Some Random Inputs! Feature Importance and Explainability . . . . .	17
3.7 The Universal Machine Learning Workflow . . . . .	17
3.8 A Machine Learning Engineer is a Data Janitor . . . . .	19
3.9 Key Takeaways . . . . .	20



Figure 3.1: "mdjrn-v4 a disembodied computer being force-fed data like a duck looking somewhat like a sci-fi character 8k" made with Mann-E

In summary, deep learning techniques handle knowledge representation by learning the underlying patterns and relationships in the data through the use of neural networks, which can then be used for prediction, classification, and generation tasks. In GOFAI, knowledge is held by the programmer and explicitly coded into rules, deep learning methods instead use data to guess the best rules from the relationships present in the dataset.

## 3.2 A Simple Neural Network is also a Linear Regression

A neural network can be mathematically equivalent to a regression or a decision tree under certain conditions.

A neural network is a machine learning model composed of layers of interconnected artificial neurons, which are designed to process and analyze data. They can be used for a wide range of tasks, such as image and speech recognition, natural language processing, and prediction.

A regression is a statistical method used to predict a continuous variable based on one or more input features. A linear regression, for example, is a simple neural network with one input layer, one output layer and no hidden layers. In this case, the weights of the network are the coefficients of the linear equation and the network is equivalent to a linear regression model.

A decision tree is a tree-based model used for classification and prediction tasks. It consists of a series of if-then rules that are used to make decisions based on the input data. A neural network with one input layer, one output layer and one hidden layer<sup>1</sup> is equivalent to a decision tree. The network implements a piecewise linear function which can represent the decision boundaries of a decision tree. So, under certain conditions, a neural network can be mathematically equivalent to a linear regression or a decision tree. These conditions include having one input and one output layers, and having a specific activation function in the case of a decision tree.[Aytekin]

## 3.3 Dummy Variables for Dummies (Wonkish)

Chapter Summary: It's all numbers, man. Machine learning techniques require that we turn everything (images, text, sound) into numbers and shove them into the model in the same way we use dummy variables in a simple regression. If you are satisfied with this, please skip this section. If you would like to learn a bit about the details and see some code examples, please keep reading. This section is necessarily technical, but should be approachable for anyone who has taken a college statistics class.<sup>2</sup>

Dummy variables are used in regression analysis to include categorical variables in a model. Categorical variables are variables that take on a finite number of distinct values, such as "red", "green", "blue" or "yes", "no". Since

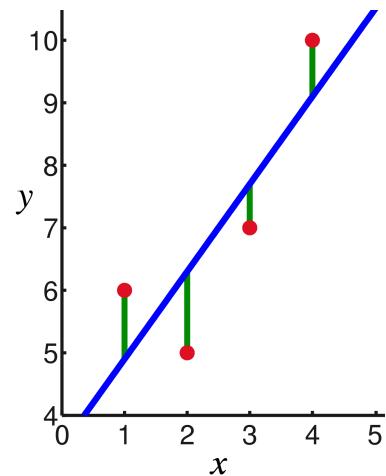


Figure 3.2: A simple linear regression, the red points are the training data, and the blue line is the regression line. If you don't understand this please read [https://en.wikipedia.org/wiki/Regression\\_analysis](https://en.wikipedia.org/wiki/Regression_analysis)

1: One with a ReLU activation, Neurons in a neural network can have different activation functions, if you don't know what it is and don't want to read about it on wikipedia, that's OK. [https://en.wikipedia.org/wiki/Activation\\_function](https://en.wikipedia.org/wiki/Activation_function)

Aytekin

2: Thanks to Paul Krugman for popularizing (to me at least) the *wonkish* classifier, I just mean it's a little nerdy. But it's important.

these variables cannot be directly included in a regression model, as they are not numerical, they need to be transformed into numerical variables.

The process of creating dummy variables is also known as one-hot encoding. It involves creating a new binary variable for each category of the original variable. For example, if you have a categorical variable "color" with three categories: "red", "green", "blue", you would create three binary variables: "color\_red", "color\_green", "color\_blue". Each binary variable would take a value of 1 if the original variable is equal to the category, and 0 otherwise.

When using dummy variables in a regression, it is important to remember to include only  $n-1$  binary variables, where  $n$  is the number of categories in the original variable. This is because including all  $n$  binary variables would result in perfect multicollinearity, which is when two or more independent variables are perfectly correlated. One of the binary variables can be dropped to avoid this problem.

Dummy variables are used in regression analysis to include categorical variables in a model. The process of creating dummy variables involves creating a new binary variable for each category of the original variable and one-hot encoding it. It is important to remember to include only  $n-1$  binary variables, to avoid perfect multicollinearity.

The creation of dummy variables in a regression is analogous to preprocessing image, text and other data for a neural network for deep learning. This preprocessing is important as it ensures that the data is in a format that can be easily understood and processed by the network. The preprocessing steps for numbers, text, and images are slightly different.

For numbers:

- ▶ Normalization: It is common to normalize the input data by scaling it to have a mean of 0 and a standard deviation of 1. This helps to ensure that all input features have similar scales and prevents any one feature from dominating the network's computations.
- ▶ Imputation: Handling missing data is important, as it can negatively impact the model's performance. Common imputation techniques include replacing missing values with the mean, median, or mode of the feature.

For text:

- ▶ Tokenization: Text data must first be converted into a numerical format that can be understood by the network. This is typically done by tokenizing the text into individual words or n-grams and then encoding them as integers or real-valued vectors. A one-hot encoding exactly like the dummy variable method used in regression is also frequently used.<sup>3 4</sup>
- ▶ Stop-words removal: The most common words in any language like "a", "an", "the", etc. that do not contain much meaning are called stop-words, they are often removed to reduce the dimensionality of the data.

3: Sometimes text is just mapped to a number! Shocking, but it works. See how it is taught in the TensorFlow tutorials [https://www.tensorflow.org/text/guide/word\\_embeddings](https://www.tensorflow.org/text/guide/word_embeddings)

4: GPT-3 uses byte-level Byte Pair Encoding (BPE) tokenization and has a vocabulary size of 50,257.

- ▶ Stemming/Lemmatization: Words that have the same meaning can be stemmed or lemmatized to reduce the vocabulary size and increase the chances of generalization.
- ▶ Vocabulary Size: Each model must choose a vocabulary size or the maximum number of tokens that it will analyze. This may cause misspellings, slang or typos to be discarded in analysis.

For images:

- ▶ Converting to RGB or Greyscale: Each image is analyzed by its pixel color value, every point on an image will either have 3 color values (red, green, blue) or one single value (on a white/black scale) if the image is analyzed in greyscale.
- ▶ Convolutions<sup>5</sup>: Pixel values are analyzed in groups that are defined by the model, since individual pixel values are only colors (or greyness) they must be combined together by the model to detect patterns like faces and stop signs. The method of convolution is defined by the model itself.
- ▶ Resizing: neural network can only accept images of a fixed size, so resizing the image to match the network's requirements is important.
- ▶ Normalization: It is common to normalize the pixel values to be in the range of 0-1 or -1 to 1. This will help the model converge faster.
- ▶ Data Augmentation: To increase the amount of data and prevent overfitting, common data augmentation techniques such as flipping, rotation, and cropping can be applied to the images.

5: If you would prefer a visualization of a neural network check out this excellent video by Dennis Dmitriev <https://www.youtube.com/watch?v=3JQ3hYko51Y>.

In summary, preprocessing is an important step in training a neural network, as it ensures that the data is in a format that can be easily understood and processed by the network. The preprocessing steps for numbers, text, and images involve normalization, imputation, tokenization, stop-words removal, stemming/lemmatization, resizing and data augmentation.

## 3.4 Try That Again With A Few Billion Parameters

In the first section of this chapter we introduced the idea that a simple neural network is mathematically equivalent to a regression. This is true and a useful way to think about neural networks and deep learning.<sup>6</sup>

In practice, a neural network trained on a gaming PC from 2023 can easily have millions of trainable parameters. A simple image classifier used to classify handwritten digits might have 60,000 parameters which is about the number the model used by Yann LeCun and company in the 1990s. Bleeding-edge deep neural networks are trained on large GPU farms and have billions of trainable parameters. More parameters gives the models tremendous power to recognize complex patterns, but also makes any practical attempt at explaining their inner-workings impossible.

When a neural network has millions of trainable parameters and deep layers of neurons<sup>7</sup>, it can be difficult to explain to a human what each of

6:

$$y = Wx + b \quad (3.1)$$

There's your simple formula for a single-cell neural network and regression, in practice we'll change  $W$ ,  $x$  and  $b$  to matrices and introduce nonlinear activation function  $f$  so,

$$y = f(Wx + b) \quad (3.2)$$

if you please. If equations scare you, don't worry about it for now.

7: In case you are wondering, the human brain has about 86 billion neurons that do the "thinking". Physical neurons are more complicated than the neurons used in deep learning and have supporting structures like astrocytes that do some computation as well. <https://en.wikipedia.org/wiki/Astrocyte>

those parameters represents or how they contribute to the overall function of the network. This is because the interactions between the different layers and neurons can be complex and non-linear, making it challenging to understand the specific role of each parameter. The parameters model complex interactions between the inputs, making it difficult to understand their specific function. Furthermore, in deep neural networks, the high number of layers can lead to high level of abstraction, meaning that the individual neurons and their weights have little interpretability.

When a neural network has millions of interacting parameters, it can lead to mathematical chaos<sup>8</sup>, which is a phenomenon where small changes in the initial conditions of the network can lead to vastly different outputs. This is because the interactions between the large number of parameters can create non-linear relationships that are sensitive to small changes. This can make it difficult to predict the behavior of the network, as small changes in the input or the parameters can lead to unexpected and seemingly random outputs.<sup>9</sup>

### 3.5 Multicollinearity and the End of Science

Data science is a horrible term because it implies that data scientists are scientists and that the work they do is scientific when in reality data scientists are not scientists and the work they do is not scientific. Data scientists use mathematics, statistics, and computer science to analyze data, but it is not scientific in the traditional sense. Data scientists do not use the scientific method and do not conduct experiments or develop theories. Data science is more akin to engineering or data analytics than actual scientific research.

The main goal of a scientist is to gain knowledge and understanding of the natural world through research, experimentation, and data analysis. Scientists make models of the world to test and explain. So-called data scientists make models too but a model with layers of interacting parameters, trained under chaotic conditions is almost impossible to explain. Multicollinearity<sup>10</sup> is just one issue, but deep learning techniques are practically incompatible with the scientific method. With deep learning methods it can be difficult to determine which variable had the most impact on the outcome of the model. Coefficients of the model to be unstable and unreliable, which can make it difficult to explain the model in a meaningful way.

Explaining the inner-workings of a neural network with millions of interacting parameters is difficult for many reasons. Most of the complexity is due to the number of parameters, which can make it difficult to understand how the model works and why it makes certain decisions. The interactions between the parameters are often non-linear and can be difficult to understand, as the model can be sensitive to small changes in the parameters, which is the definition of mathematical chaos. This can make it difficult to explain why the model is making certain decisions, as the interactions between the parameters are not always clear.

8: Mathematical chaos is a real thing, it refers to the behavior of certain dynamic systems that are highly sensitive to initial conditions. This sensitivity leads to seemingly random and unpredictable behavior, even though the underlying equations governing the system are deterministic (meaning they are transparent and known to us). [https://en.wikipedia.org/wiki/Chaos\\_theory](https://en.wikipedia.org/wiki/Chaos_theory)

9: Chaotic outputs in a large deterministic system are OK in many domains, and there are controls that can be put in place to keep AI "safe" but deep learning by itself will not produce those controls.

10: "Multicollinearity is when two or more variables in a study are closely related to each other. Think of it like dating: imagine you are trying to figure out what makes someone a good partner. You might look at things like how much they make, how attractive they are, and how kind they are. But, these things are all related to each other. For example, attractive people might make more money, and kind people might be more attractive. So, it's hard to know which one is really important for being a good partner, because they are all related. That's like multicollinearity in a study." By ChatGPT given the prompt "explain multicollinearity to a teenager using a dating example"

## 3.6 Let's Test Some Random Inputs! Feature Importance and Explainability

With a huge complex system of neurons that combine inputs in novel ways, it becomes very hard to understand which inputs are the most important for the system as a whole. To better understand deep learning models, data scientists use randomized or averaged inputs to a model to test the feature importance of a deep learning neural network. This type of testing is done to determine which features are most influential in the overall output of the network. This is done by randomly or averaging the input values for each feature and then running the model to see how the output is affected. For example, if a neural network is used to detect objects in an image, the data scientist may randomize the color of the objects to see how the model's accuracy is affected. If the accuracy drops significantly, they can infer that color is an important feature in the network.

Randomized or averaged inputs to a model can also be used to determine if a particular feature is necessary for the network to function correctly. For example, if the output of the network is not as accurate when a particular feature is randomized or averaged, then the data scientist can infer that this feature is important for the network's performance. By using this method, data scientists can gain insights into which features are important and which can be removed from the model to improve performance.

## 3.7 The Universal Machine Learning Workflow

The Universal Machine Learning Workflow is an important chapter in a technical guide for data scientists by the co-author of Tensorflow, the most popular machine learning framework in the world as of 2023, that outlines the *Universal* workflow for machine learning projects. I think this chapter should be understood by everyone using, investing in and creating machine learning models.

Before Chollet gets into the details of model building, he chooses to begin with a note on ethics:

*"You may sometimes be offered ethically dubious projects, such as "building an AI that rates the trustworthiness of someone from a picture of their face." First of all, the validity of the project is in doubt: it isn't clear why trustworthiness would be reflected on someone's face. Second, such a task opens the door to all kinds of ethical problems. Collecting a dataset for this task would amount to recording the biases and prejudices of the people who label the pictures. The models you would train on such data would merely encode these same biases into a black-box algorithm that would give them a thin veneer of legitimacy. In a largely tech-illiterate society like ours, "the AI algorithm said this person cannot be trusted" strangely appears to carry more weight and objectivity than "John Smith said this person cannot be trusted," despite the former being a learned approximation of the latter. Your model would be laundering and operationalizing at scale the worst aspects of human judgement, with negative effects on the lives of real people. [chollet2022]*



Figure 3.3: "the face of a trustworthy person" made with Stable Diffusion 2.1. (Hey, it's a white lady!)

*Technology is never neutral. If your work has any impact on the world, this impact has a moral direction: technical choices are also ethical choices. Always be deliberate about the values you want your work to support.”*[chollet2022]

Chollet uses the outline below to explain the *Universal* workflow. I'll summarize the workflow and my own notes for a nontechnical audience here:

### 1. Define the Task

- a) Collect a Dataset<sup>11</sup>
- b) Understand Your Data
- c) Choose a Measure of Success<sup>12</sup>

### 2. Develop a Model

- a) Prepare the Data<sup>13</sup>
- b) Choose an Evaluation Protocol
- c) Beat a Baseline<sup>14</sup>
- d) Develop a model that overfits
- e) Regularize and Tune Your Model

### 3. Deploy the Model

- a) Explain Your Work to Your Stakeholders and Set Expectations<sup>15</sup>
- b) Ship an Inference Model
- c) Monitor Your Model in the Wild
- d) Maintain Your Model

Of all of the steps in the workflow, "Defining The Task" and "Explaining The Work To Shareholders and Setting Expectations" are where the most miscommunication occurs.

In defining the task, machine learning engineers are often given impossible problems to solve, and because they want to keep paying their mortgage they solve another related problem instead and allow stakeholders to jump to their own conclusions. By clearly understanding what a model is predicting and how the data is collected some of this misunderstanding can be avoided.

I will discuss Large Language Models later, but they all do the same thing. They predict the next words in a sentence, just like the keyboard on your iPhone. They come up with amazing text but by fundamentally understanding what they are predicting you gain insight into their limitations. They are also all trained on the text publicly available on the internet, this includes scientific sources, but also fan-fiction and anime, so the idea that a model trained on this data could be relied on to predict anything truthful is preposterous.

Many models work like magic and many users assume models are predicting the future, when they are really correlating based on their past data. Even simple models of creditworthiness might have a similar problem. While building a creditworthiness model for a bank, due to lack of data a machine learning engineer might (on purpose or inadvertently) create a model that predicts whether someone is a smoker instead of whether they are creditworthy. Because smoking and poverty are correlated, maybe the model "works", but it doesn't do what stakeholders think it does.

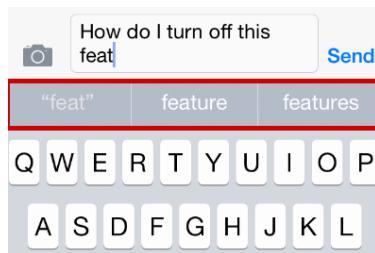
11: This is often the hardest part. Data is often labeled by hand or stolen from another domain. See [https://en.wikipedia.org/wiki/Transfer\\_learning](https://en.wikipedia.org/wiki/Transfer_learning)

12: Accuracy isn't everything! We might prefer a less accurate model that avoids false-positives in some scenarios.

13: This is discussed in section 3.3 above, everything gets turned into numbers.

14: Does your model predict better than a totally random model and/or coin-flip?

15: This is almost never done. And as Chollet explains this is both difficult and requires mutual understanding and domain knowledge from the part of "the business" and the Machine Learning Engineer.



**Figure 3.4:** Fundamentally LLMs are using the same techniques as a predictive keyboard on your phone. This is why Yann LeCun says even though they are impressive some like ChatGPT are not particularly innovative. <https://www.youtube.com/watch?v=ULbpPHj1SBg>

Setting expectations is another hellscape of misaligned incentives.<sup>16</sup> Good machine learning engineers and data scientists are supposed to educate and advise stakeholders about the limitations of their own models. Read Chollet's advice on this topic below:

*The expectations of non-specialists towards AI systems are often unrealistic. For example, they might expect that the system "understands" its task and is capable of exercising human-like common sense in the context of the task. To address this, you should consider showing some examples of the failure modes of your model (for instance, show what incorrectly classified samples look like, especially those for which the misclassification seems surprising).*

*They might also expect human-level performance, especially for processes that were previously handled by people. Most machine learning models, because they are (imperfectly) trained to approximate human-generated labels, do not nearly get there. You should clearly convey model performance expectations. Avoid using abstract statements like "The model has 98 percent accuracy" (which most people mentally round up to 100 percent), and prefer talking, for instance, about false negative rates and false positive rates. You could say, "With these settings, the fraud detection model would have a 5 percent false negative rate and a 2.5 percent false positive rate. Every day, an average of 200 valid transactions would be flagged as fraudulent and sent for manual review, and an average of 14 fraudulent transactions would be missed. An average of 266 fraudulent transactions would be correctly caught." Clearly relate the model's performance metrics to business goals.*

*You should also make sure to discuss with stakeholders the choice of key launch parameters- for instance, the probability threshold at which a transaction should be flagged (different thresholds will produce different false negative and false positive rates). Such decisions involve trade-offs that can only be handled with a deep understanding of the business context.[chollet2022]*

One does not need to be a psychologist to understand that these conversations almost never happen. The workflow of machine learning is universal, we are all doing fancy regressions and we all deal with the same wild expectations, bad data and misalignment with business.

### 3.8 A Machine Learning Engineer is a Data Janitor

Since the machine learning workflow is *universal*, it is fairly straightforward to automate. There are a plethora of offerings from companies big and small who offer AutoML tools that anyone with basic Excel skills can use<sup>17</sup>. Since modern AI is "all a regression" these tools do the regression for you, users just need to feed them the data and the AutoML tool finds out the relationship. On the surface it seems like the job of the machine learning engineer has become obsolete before it even became a proper discipline.

Despite the existence of AutoML tools, machine learning engineers (and data scientists) do some actual work. A 2016<sup>18</sup> Crowd Flower survey of

16: The CEO of OpenAI wrote this and still raised ten billion dollars: "*ChatGPT is incredibly limited, but good enough at some things to create a misleading impression of greatness. it's a mistake to be relying on it for anything important right now. it's a preview of progress; we have lots of work to do on robustness and truthfulness.*" <https://twitter.com/sama/status/1601731295792414720>



Figure 3.5: "mdjrny-v4 a handsome businessperson explaining the business context to a scientist wearing a white coat over coffee 8k" made with Mann-E

17: See this awesome-AutoML github project for an up-to-date list of commercial AutoML tools <https://github.com/windmaple/awesome-AutoML#commercial-products>

18: I know this study is old, but I have found that the findings still hold when talking to data scientists and/or machine learning engineers that I train and employ.

16,000 data scientists showed that on average we spend our time doing the following:

- ▶ **60 percent of a machine learning engineer's time is spent cleaning and organizing data**, we are data janitors! If you had great data ready in an Excel file you would have been able to fit your own model without us. We need to organize the mess of data that exists in the world so it is in a nice format to be fit in a model. For example it is well known that almost all large language models use the same common crawl dataset (<https://commoncrawl.org/>), but how it is organized and weighted can change the quality of the outputs dramatically. The organization of the same dataset can lead to the same dataset either producing a nice predictive keyboard or something approaching ChatGPT.
- ▶ **19 percent of a machine learning engineer's time is spent collecting data**, we often start making models without a huge dataset, so data needs to be collected in order to make the first model, and after that first model (to match people for our new dating app, let's say) is deployed we rely on users to provide us with data for future models. Collecting new data is a creative endeavour sometimes, and often involves human effort. Amazon's Mechanical Turk<sup>19</sup> service has been leveraged for this purpose for years, and OpenAI has successfully deployed outsourced talent to solve some of the trickiest problems facing large language models<sup>20</sup>.
- ▶ **9 percent of a machine learning engineer's time is spent fitting models**, we need to fit models, there is some art to choosing the right architecture and modeling techniques. An AutoML tool can do this reasonably well too, but considering some fancy training servers cost \$28 per hour or more<sup>21</sup>, sometimes it is useful to have an expert guess the best model architecture and save time in training, even if that expert is getting paid \$249,000 per year<sup>22</sup>.
- ▶ **4 percent of a machine learning engineer's time is spent refining algorithms** many machine learning engineers spend almost no time doing this. Other researchers (corporate-academic types who write scientific papers) spend a lot of time doing this, it averages out to four percent, but doesn't represent the average day of a machine learning engineer.

Overall, the main job of most machine learning engineers is to clean up and collect a ton of data, and then feed that data into the same machine learning model that everyone else is using, and then rinse and repeat.<sup>23</sup>

### 3.9 Key Takeaways

- ▶ **Deep learning models are fundamentally large unscientific regressions** they are trained to create a function that maps input data to output data.
- ▶ **Deep learning models are chaotic systems containing millions of interacting parameters** they are not designed to be explained

19: "Getting the right training data was another challenge. ImageNet was a collection of one hundred thousand labeled images that required significant human effort to generate, mostly by grad students and Amazon Mechanical Turk workers." <https://arstechnica.com>

20: Read more about how OpenAI used Kenyan workers to help label toxic content <https://time.com/6247678/openai-chatgpt-kenya-workers/>.

21: See the latest pricing from AWS here <https://aws.amazon.com/sagemaker/pricing/>

22: See the latest machine learning engineer salaries at levels.fyi <https://www.levels.fyi/t/software-engineer/focus/ml-ai>

23: Here is a discussion on reddit asking "When was the last time you wrote a custom neural net?" and supports my understanding of the world [https://www.reddit.com/r/MachineLearning/comments/yto34q/d\\_when\\_was\\_the\\_last\\_time\\_you\\_wrote\\_a\\_custom/](https://www.reddit.com/r/MachineLearning/comments/yto34q/d_when_was_the_last_time_you_wrote_a_custom/)

or created in a way that their weights can be used for scientific analysis. They find reasonable answers and don't care how they get there. Multicollinearity (understanding the relationship of an input and output) and feature importance (understanding which inputs are most important) are only understandable with a high level of statistical error.

- ▶ **Small changes in inputs of a deep learning model may dramatically change the outputs** deep learning models are complex deterministic systems that can exhibit chaotic behavior. Their inner workings are functionally unknowable and practically impossible to test.
- ▶ **Machine learning engineers spend most of their time collecting and organizing data**, because deep learning models often share common architecture, getting good data to train with is the best thing that an engineer can do to train good models. In practice only a handful of corporate-academic types are experimenting with new and exciting architectures.
- ▶ **Deep learning models can have impressive and useful outputs, but the creators of models should be encouraged to highlight their failures and limitations.** Machine learning engineers might be more keen to highlight failures and limitations if they are encouraged to do so by their users, managers and investors.

# Creativity and Decision Making with Deep Learning Models

# 4

## "AI Policy

*I expect you to use AI (ChatGPT and image generation tools, at a minimum), in this class. In fact, some assignments will require it. Learning to use AI is an emerging skill, and I provide tutorials in Canvas about how to use them. I am happy to meet and help with these tools during office hours or after class*

*Beware of the limits of ChatGPT:*

- *If you provide minimum effort prompts, you will get low quality results. You will need to refine your prompts to get good outcomes. This will take work.*
- *Don't trust anything it says. If it gives you a number or a fact, assume it is wrong unless you either know the answer or can check in with another source. You will be responsible for any errors or omissions provided by the tool. It works best for topics you understand.*
- *AI is a tool, but one that you need to acknowledge using. Please include a paragraph at the end of any assignment that uses AI explaining what you used the AI for and what prompts you used to get the results. Failure to do so is in violation of academic honesty policies*
- *Be thoughtful about when this tool is useful. Don't use it if it isn't appropriate for the case or circumstance."*

*Dr. Ethan Mollick, 2023 - Syllabus for class at The Wharton School at the University of Pennsylvania*

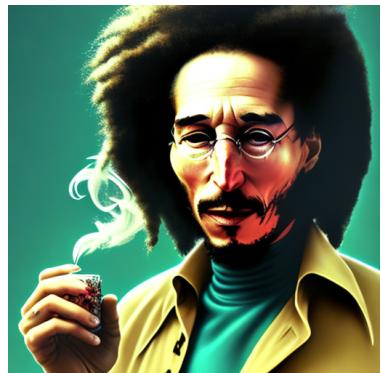
## 4.1 Theories of Creativity

Machine learning models use data (from the past) to discover rules and make classifications. Because of the way they are constructed these classifications, suggestions, artworks are by definition derivative or "having parts that originate from another source". I won't get into a philosophical discussion on what the nature of creativity is, but it's worth considering how using deep learning models biases us towards the past, but also could give us insights from other domains.

It can be argued that creativity is simply the combination of existing works. This is because many new ideas and innovations are often inspired by and built upon existing concepts. For example, a new form of music may be created by combining elements from different genres. Similarly, a new technology may be created by combining and improving upon existing technologies.

It can also be argued that creativity involves much more than just combining existing works. Creativity is not just about recombining existing ideas, but also about coming up with completely new and original concepts. This

4.1 Theories of Creativity . . . . .	22
4.2 Creative Uses of Power Tools . . . . .	23
4.3 Garbage In, Garbage Out . . . . .	24
4.4 Garbage In, New Perspective Out? . . . . .	24
4.5 Concept Drift and the End of Usefulness . . . . .	25
4.6 The Impossibility of Fairness . . . . .	26
4.7 Transfer Learning Everywhere . . . . .	26
4.8 Industrial-Scale Plagiarism . . . . .	28
4.9 Working Together is Best . . . . .	28
4.10 Key Takeaways . . . . .	29



**Figure 4.1:** "mdjrny-v4 Steve Jobs Smoking weed with Bob Marley 8k" made with Mann-E. It's 100% derivative, but it's art too (I guess).

requires a unique perspective and a deep understanding of the subject matter, as well as the ability to think outside the box.

Deep learning models of speech, when heavily used, may slow down the evolution of language. AI art models may slow down "progress" in art, whatever that means. Deep learning models of disease trained on data from 1980 may be irrelevant to today's diseases. That said these same models may give us interesting insights in new domains, models trained on beautiful paintings may be put to use in a new domain (like designing beautiful interiors) and that model could give new insight to interior designers, models of the interaction of ants could be put to use in designing cities and so on and so forth. AI cuts many ways, it makes us faster but makes us more reliant on the past, models can be used across domains but should be used intentionally and transparently when possible. Each use opens up a new world of possibilities for users, and sometimes a new headache for intellectual property lawyers. We'll discuss all of these topics in this chapter.

## 4.2 Creative Uses of Power Tools

Power tools, even simple ones like a drill are complex machines that take human input and transform it using static rules. The pressure the user of a drill puts on the bit and the speed at which they pull the trigger all deterministically affect the output. Just because a tool is a deterministic machine doesn't mean it is unable to produce creative works.

What is happening as we use generative tools like ChatGPT or image-to-text models is that the "creative act" has been relocated. The creative act is now the prompt you give the tool, the user's input. And someone can still be good at using AI, just like someone can be good at using any piece of software.

Modern AI is a complex system of algorithms, data, and analytics that can be used to solve complex problems. AI systems can learn from data, identify patterns, and make predictions about the future. AI systems are typically used to automate and assist human decision making. AI systems are programmed with specific objectives and goals, and the user input decides the output. For example, an AI system could be programmed to solve a mathematical problem and the user input would determine the parameters of the problem and the output would be the solution.

A power drill is also a complex deterministic system. The user input is limited to the type of drill bit, the speed of the drill, and the direction of the drill. The output is determined by these inputs, as the drill will only drill in the direction and speed determined by the user. The user also has to choose the correct drill bit to ensure the drill can do the job correctly and safely.

Both modern deep learning models and a power drill are complex but ultimately deterministic systems. The user input, however limited it may be, ultimately controls the output of the system. Both systems require a



**Figure 4.2:** "Batman dressed as a cowboy" made with Mann-E. It's 100% derivative as well, but better than any Google image searches I did. It looks amazing! I also might get sued if I use it even though the creators of Mann-E tell me I have the rights to it.



**Figure 4.3:** "mdjrnny-v4 a mikita drill being used in an artist's studio to make a colorful artwork 8k" made with Mann-E

user to understand how to use them and to make the correct input to get the desired output.

### 4.3 Garbage In, Garbage Out

The "Power Tool" of AI is a deterministic, static and unchanging system and the rules of that system are determined by the data that the model is shown. If a model is trained on bad data, it will produce poor results. End of book...

... maybe not. It's worth thinking about this for a moment. It is often said that modern AI can "generalize and make informed inferences given new data". If deep learning models are really just a big regression, these models will always come up with an answer, but if the world changes, these models will still be projecting complicated averages of their past data into the future.

So, let's separate AI's decision making into two extremes; *Creative Decision Making* and *Critical Decision Making*. The stakes are very low in a world of *Creative Decision Making* and who cares if the AI is all a regression, and it just mushes together the limited data that it's seen. In a creative context you can also ask an AI interesting questions, so long as you don't solely rely on its output without checking the facts first<sup>1</sup>. Even if "Garbage In, Garbage Out" holds, garbage can still be helpful for a creative process.

Note that this book is called "Full-Self Driving, Skynet..", a self-driving car and a nuclear-bomb-equipped all-seeing AI are clearly not engaging in any *Creative Decision Making*.

1: See the syllabus note from Dr. Mollick at the beginning of this chapter.

### 4.4 Garbage In, New Perspective Out?

For creative tasks, it generally doesn't matter that a deep learning model is unscientific or trained on a lopsided dataset. An informed user of AI knows this and can account for that in their decision making, especially when engaging in creative decision making. The situation becomes problematic once we allow deep learning models to engage in critical decision making by themselves.

Generative models will come up with amazing (but derivative) works of art, But, they will never "change the game". A model of sculpture trained on past sculptures in 1917 will never come up with Marcel Duchamp's "Fountain". There is a term that machine learning engineers use for this, when the rules of the game are changed, it's called "Concept Drift".



**Figure 4.4:** Marcel Duchamp's "Fountain". A urinal that blew peoples minds <https://www.tate.org.uk/art/artworks/duchamp-fountain-t07573>.

## 4.5 Concept Drift and the End of Usefulness

Concept drift refers to the phenomenon where the distribution of data changes over time, causing the performance of deep learning models built on historical data to degrade. The model becomes less useful because it is trained to make predictions based on the relationship between the inputs and outputs in the data it was trained on, and if this relationship changes, the model may start making incorrect predictions. This is particularly problematic in real-world applications, where the data is constantly evolving and the relationship between inputs and outputs is subject to change. To mitigate the effects of concept drift, it is often necessary to continually retrain deep learning models on updated data.

The frequency with which a deep learning model needs to be retrained to mitigate the effects of concept drift depends on several factors, including the rate at which the data distribution changes, the size and complexity of the model, and the availability of computational resources.

For some applications with relatively stable data distributions, retraining the model once every few months or even once per year may be sufficient. However, in other applications where the data is changing rapidly, it may be necessary to retrain the model more frequently, such as once per week or even once per day.

Ultimately, the frequency of retraining will depend on the specific requirements of the application, and the trade-off between the cost of retraining and the potential cost of incorrect predictions. In general, it's recommended to monitor the performance of the model over time and to retrain it as needed to ensure that it remains accurate and relevant.

If a deployed deep learning model is not monitored, there are several risks that can emerge:

- ▶ Accuracy degradation: As the data distribution changes over time, the model may become less accurate, leading to incorrect predictions. This can result in financial losses, reduced customer satisfaction, or even harm to individuals.
- ▶ Bias amplification: deep learning models can be biased, and if this bias is not monitored and addressed, it can be amplified over time as the model continues to make incorrect predictions. This can result in discriminatory outcomes, such as unequal treatment of individuals based on protected characteristics such as race, gender, or age.
- ▶ Legal liability: In some cases, incorrect predictions made by deep learning models can result in legal liability, particularly if the model is being used to make decisions that have significant consequences, such as in the criminal justice system or in medical diagnosis.
- ▶ Reputational damage: If a deep learning model is making incorrect predictions, it can damage the reputation of the organization deploying the model, potentially leading to a loss of customers or investors.

It is important to monitor deep learning models once they are deployed, and to take action to address any issues that arise, such as retraining the



**Figure 4.5:** "Plastic surgery gone wrong" made with Stable Diffusion. Imagine a model that classifies images as "human face" or "not human face", and imagine that model was trained on images of human faces before 1900, maybe you would not be surprised if you gave it a picture of a human face that had a lot of cosmetic surgery done to it, and that model might say "this is not a human face", the idea of what a human face is has changed over time this is called "concept drift".

model or adjusting its parameters, in order to mitigate these risks and ensure that the model continues to perform well over time.

## 4.6 The Impossibility of Fairness

Achieving fairness in deep learning models can be challenging, and to some extent it may be impossible to completely eliminate all forms of bias. This is because deep learning models are trained on historical data, which may contain biases and disparities that are perpetuated in the model's predictions. One can attempt to "fix the training set" but in practice models will continue to bias their prediction to past data, or their creators careful curation of the past.[Christian2020]

Christian2020

However, it is possible to reduce the impact of bias in deep learning models through careful design and monitoring of the model's performance. This may include techniques such as fairness constraints, algorithmic transparency, and regular auditing of the model's predictions to identify and address any issues of bias.

It's important to note that fairness is a complex and multi-faceted concept, and different definitions of fairness may be appropriate for different applications. For example, some definitions of fairness may prioritize equal treatment of all individuals, while others may prioritize proportional representation or equal opportunities.

Ultimately, the extent to which fairness is achievable in deep learning models will depend on the specific requirements of the application and the level of effort that is put into designing and monitoring the model to ensure that it is making fair and unbiased predictions.

## 4.7 Transfer Learning Everywhere

Transfer learning is a machine learning technique that involves transferring knowledge from one model trained on a task to another model trained on a related task. The idea behind transfer learning is that a model that has been trained on one task can be fine-tuned for another task, reducing the amount of labeled data required to train the new model.

For example, imagine that you have a deep learning model that has been trained to recognize objects in natural images (cats, dogs, balls, cars). You can use the knowledge learned by this model as a starting point to train a new model that recognizes objects in medical images, such as X-rays or MRI scans. The new model can start with the weights of the pre-trained model and fine-tune them on the new task, using a much smaller amount of labeled data than would be required to train the model from scratch.

Transfer learning can be useful in many different applications, particularly when labeled data is scarce or expensive to obtain. By leveraging knowledge

from a pre-trained model, transfer learning can help to improve the performance of new models, reduce the amount of data required for training, and accelerate the development of new machine learning applications.

I asked ChatGPT to "tell me a story of transfer learning going horribly wrong in the context of online dating, with a model trained in the Middle East being deployed by an American dating app." and here's what it said:

*"Once upon a time, an American online dating company decided to use transfer learning to improve their matchmaking algorithms. They trained a state-of-the-art machine learning model on a large dataset of profiles and dating preferences from the Middle East. The model was able to suggest matches with remarkable accuracy and the company was confident that it would perform just as well in the American market.*

*However, things went terribly wrong when they deployed the model on their American dating app. The model was not properly trained to consider the cultural and social differences between the Middle East and America. As a result, it suggested matches that were inappropriate and offensive, causing widespread backlash and criticism from users.*

*For example, the model suggested matches between people of different religious backgrounds, which was not acceptable to many users. It also suggested matches between people with significant age gaps, which was not socially acceptable in the American market.*

*The company was forced to remove the model and start from scratch, causing significant harm to their reputation and financial losses. They learned the hard way that transfer learning can only be effective if the model is properly trained on relevant and culturally appropriate data.*

*This story serves as a cautionary tale about the importance of considering cultural differences and ethical implications when deploying machine learning models in different regions and markets."*

Not bad I think... picking good training data and doing good "data janitoring" is the main job of a machine learning engineer<sup>2</sup>.

If the past is not like the future, you are also doing transfer learning. Most models steal data from other sources so are doing transfer learning too. This is fine, but do we know that we are doing this? A model that is deployed in a domain experiencing concept drift and continues to make predictions without being retrained can be considered to be doing a harmful form of transfer learning. This is because the model has been trained on a different distribution of data (the past), and is being applied to a new domain with a different distribution (the present and future).

In traditional transfer learning, the goal is to transfer knowledge from one domain to another related domain, where the data distributions are similar enough to enable the model to make accurate predictions. However, in the case of concept drift, the data distributions are changing over time, and the model is becoming less accurate as a result.<sup>3</sup>

By continuing to make predictions without being retrained, the model is essentially "transferring" its knowledge from a historical data distribution

2: The better your machine learning engineer understands your business problem, the better job they'll do at creating the dataset.

3: Think about your life and if you "transferred" your model of thinking from when you were 12 years old, to "today" when you are 35 years old, that is a bad model to be operating on! That model needs to be updated. We'll explore many examples of where this is and is not a problem in the next chapter."

to a new, changed data distribution, which may not be a valid assumption. This can result in incorrect predictions and other negative outcomes, such as harm to individuals or organizations.

## 4.8 Industrial-Scale Plagiarism

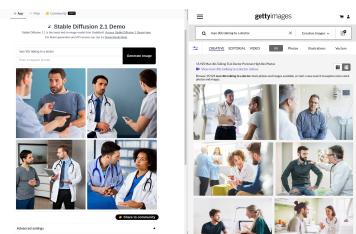
Aside from regurgitating the past and predictions from other domains, deep learning models can also enable plagiarism on an industrial scale. Early text generation models could be made to write entire sections of *Harry Potter* when fed the opening lines of a chapter. Even as models grow large and more sophisticated, users, researchers and lawyers are still able to "extract the training data" from large models[Carlini2023], causing headaches for their creators and adding to the work of intellectual property lawyers.

To avoid this creators of deep learning models have the following tools at their disposal:

- ▶ Best practices in machine learning: data preprocessing, augmentation, regularization, model architecture choices, early stopping and validation are all things that we are taught to do to prevent overfitting.
- ▶ Contractual agreements: Microsoft (owner of github and cocreator of the [Github Copilot](#) code generation model) has a special contract for anyone submitting code to github that essentially says that "we are allowed to use this code to train our models, and if our models regenerate your copyrighted code, you can kick rocks."
- ▶ Release the Kraken: StabilityAI's Stable Diffusion model was released as open source software, they still managed to get sued [[getty](#)], but they basically said, "this model was trained on all copyrighted and copylefted images on the web, sometimes it'll generate stuff that violates copyright law, but we are in Germany and will give this thing away for free to the public, and see how Getty Images, photographers and graphic designers of the world handle it... good luck!".

These techniques can help to reduce the risk of having deep learning models reproduce the training data and cause intellectual property disputes, but there is no way to completely eliminate these risks, they are simply side-effects of the method of machine learning that we are doing nowadays. If you are an IP lawyer and need an expert witness, I'm your man [brad@bradflaugh.com](mailto:brad@bradflaugh.com).

Carlini2023



**Figure 4.6:** "man 30s talking to a doctor" Getty Images and Stable Diffusion comparison. Is Stable Diffusion creating something new here or spitting out its training data? Is training a model like this fair use? I don't know, have fun lawyers! Note rights to each Getty photo runs around \$450, and Stable Diffusion is free to use, and they give users the rights to the model output. The Kraken has been released!

## 4.9 Working Together is Best

It turns out sometimes the best combination is a reasonably smart (but not too cocky) human that gets to get recommendations from a few deep learning models.<sup>4</sup>

Garry Kasparov, the legendary chess player we heard about in chapter 1, witnessed a tournament where humans and AI could compete together.

4: Funnily enough at Wharton I recently took a [class](#) where Dr. Cade Massey said basically "If you want to bring people along and get them to use your model, let them play with the output a bit, allowing for even the slightest bit (2%) adjustment in the outputs will get users to adopt your model much faster." Thanks Cade!

The tournament was designed to showcase the strengths of both humans and AI, and how they could work together to achieve incredible results here is what Garry says about the tournament:

*"Once again, the chess world offers a useful test case for how this collaboration can play out. In 2005 the online chess playing site Playchess.com hosted what it called a "freestyle" chess tournament in which anyone could compete in teams with other players or computers. What made this competition interesting is that several groups of grandmasters working with computers also participated in this tournament. Predictably, most people expected that one of these grandmasters in combination with a supercomputer would dominate this competition ; but that's not what happened. The tournament was won by a pair of amateur American chess players using three computers. It was their ability to coordinate and coach effectively their computers that defeated the combination of a smart grandmaster and a PC with great computational power."* [kasparov2005]

kasparov2005

The tournament was a huge success, and showed that when humans and AI work together, they can achieve amazing things. The tournament participants learned that AI is not just a tool, but a valuable partner, and that by combining their strengths, they could achieve results that neither could have accomplished on their own. The tournament inspired many people to explore the potential of human-AI collaboration, and showed that by embracing technology, we can create a brighter future for all.

AI is an amazing partner, but we need to think for ourselves too. We can't blindly trust AI, but we can use it to inspire and challenge us.[masharamani2020]ansharamani2020

## 4.10 Key Takeaways

- When you are using deep learning models, you are almost always engaging in some kind of transfer learning. You are assuming the past will be like the future, or that training set reflects reality, which isn't always the case.<sup>5</sup>
- Models can puke out seemingly creative things, but they are still a product of their training data. deep learning models are deterministic systems, they slap together data based on rules they guessed from their training data. This process can still be very useful, but deep learning models are not sentient creative creatures from science fiction.
- Even if you think you "own the rights" to the output of a model, models can sometimes puke out their training data modelers try their best to prevent this but models can spit out their training data, which may cause headaches for everyone except the intellectual property lawyers.
- Users should consider the quality and appropriateness of the data the model they are using is trained on before deciding to use a model. Remember, machine learning engineers spend most of their time cleaning up and gathering data, if you ask questions about how and when the data was collected and cleaned up you can get ahead

5: It's as simple as the common financial disclaimer *Past performance is no guarantee of future results*. Even though it is everywhere, the general public and even big swingers in finance don't seem to get this one. In AI there is an additional disclaimer, *Training data may not reflect the the "real world"*.

of potential problems of concept drift and blind transfer learning (engaging in transfer learning without your consent).

- **Work with the technology** don't make your model do everything, sometimes it'll come up with stupid answers. Also don't assume you are always right, try and remember the story of the freestyle chess tournament and the grandmasters who got beat by the amateurs getting reasonable advice from a few chess engines. Working with your model, as if it were a coworker that you don't fully trust, but still think is smart is probably the best way to use deep learning models.

# 5

## Case Studies

### 5.1 Introduction

This chapter is a dissection of every important model I could think of, and many that I have built myself. If our goal is to understand and critique AI models, then we need to understand the models themselves. I know of no better way to understand than to take models apart then rebuild them. In the first three chapters you learned all of the key topics you need to do a high-level dissection (if you haven't read the first three chapters please go back and do that).

To facilitate this dissection I am going to use a tool called a **model card**. Model cards are a way to document the capabilities and limitations of your AI model. They are a useful tool for understanding your model, and for communicating its capabilities and limitations to others. Google (see <https://modelcards.withgoogle.com/face-detection>), Hugging-face and even small makers of AI models produce model cards to help educate their users and developers, and as you can imagine no one reads them.

In this chapter I am going to create and add commentary to model cards for many popular models. Whether you are a technical person or not I want you to be able to read and understand these, and eventually I want my notes or criticisms to become self-evident. My idea is that once you understand the data that is used to train a model and the domain that it is deployed, you can forecast its strengths and weaknesses almost automatically, let's try a few and see how it goes.

Note that I'll introduce these models in a very particular order. I'll also save the hairiest models (full-self driving and Skynet) their own chapters for a deeper discussion. Try and follow along in the order that I lay out here, but only until you get bored. Once you are bored and can forecast my criticisms, you can skip around with impunity.

I'm happy to add future dissections or model cards to this chapter, so if you have a favorite model that you'd like to see dissection of, please let me know by emailing [brad@bradflaugher.com](mailto:brad@bradflaugher.com).

TODO Explain cost, quality and IP protection scores.

#### 5.1.1 Sucker Traps to Note When Analyzing Models

*"The sucker's trap is when you focus on what you know and what others don't know, rather than the reverse." Nassim Taleb, 2016[procrustes]*

5.1	Introduction	31
5.1.1	Sucker Traps to Note When Analyzing Models	31
5.1.2	Key Terms	32
5.2	Model Cards	32
5.2.1	Fluid Dynamics	32
5.2.2	Shakespearean Text	33
5.2.3	Lithium Mining	34
5.2.4	Chess	35
5.2.5	Go	37
5.2.6	Large Stock Order	38
5.2.7	Share Tender Purchase	39
5.2.8	Stock Trading	40
5.2.9	Horse Racing	41
5.2.10	Sperm Counter	41
5.2.11	Handwriting Recognizer	42
5.2.12	Drug Discovery	43
5.2.13	Autism	45
5.2.14	Online Dating	46
5.2.15	Online Advertising	47
5.2.16	Tennis	48
5.2.17	Hate Speech	49
5.2.18	Fake News	50
5.2.19	Legal Contracts	51
5.2.20	Facial Recognition	53
5.2.21	Smartwatches	54
5.2.22	Threat Detection	55
5.2.23	University Admissions	56
5.2.24	Credit Score	57
5.2.25	Social Credit Score	59
5.2.26	AGI	60
5.3	Key Takeaways	61

In chapters 1-3 there were a number of key concepts that we discussed that I'll call "Sucker Traps". Below is a list of those concepts that I will refer to regularly when discussing the model cards in this chapter:

- ▶ Creative or Critical: The difference between a model that is used to inform a decision and a model that is used to make a decision.
- ▶ Explainability: The ability to understand why a model makes a particular prediction.
- ▶ Plagiarism: The capacity for a model to output copyrighted material from its training data.
- ▶ Concept Drift: The possibility and frequency of changes in the relationships in the training data.
- ▶ Editorializing: The possibility of a modeler either explicitly or accidentally creating a biased training set and thus permanently shifting the outputs of the model away from reality.

### 5.1.2 Key Terms

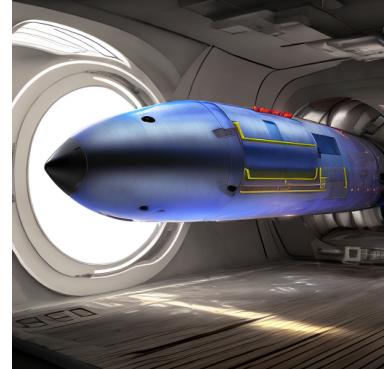
- ▶ **Accuracy:** The proportion of correctly classified examples to the total number of examples.
- ▶ **AUC:** Area under the Receiver Operating Characteristic (ROC) curve.
- ▶ **F1:** Harmonic mean of precision and recall.
- ▶ **False Negative (FN):** An instance that is actually positive but predicted as negative.
- ▶ **False Positive (FP):** An instance that is actually negative but predicted as positive.
- ▶ **Precision:** The proportion of correctly classified positive examples to the total number of positive examples predicted by the model.
- ▶ **Recall:** The proportion of correctly classified positive examples to the total number of actual positive examples.
- ▶ **True Negative (TN):** An instance that is actually negative and predicted as negative.
- ▶ **True Positive (TP):** An instance that is actually positive and predicted as positive.

## 5.2 Model Cards

### 5.2.1 Fluid Dynamics

- ▶ **Description:** This model is intended to predict fluid flow patterns in various applications, such as aerodynamics, hydrodynamics, and weather forecasting.
- ▶ **Training Data:** The model is trained on a large dataset of numerical simulations of fluid flow patterns, which includes various geometries, fluid properties, and boundary conditions. The data is generated using well-established simulation tools such as the finite element method, finite volume method, or lattice Boltzmann method.

- ▶ **Evaluation:** The model is evaluated on a separate dataset of numerical simulations of fluid flow patterns that are not seen during training. The evaluation dataset includes a variety of geometries, fluid properties, and boundary conditions. The model's performance is measured using common metrics in fluid dynamics such as root-mean-square error (RMSE), mean absolute error (MAE), and correlation coefficient (R).
- ▶ **Limits and Risks:** This model is intended for scientific and engineering applications and is not directly used for decision-making that affects human lives. However, the model's predictions may indirectly affect human lives by informing engineering design or emergency response planning. It is important to validate the model's accuracy and uncertainty, and to communicate the limitations and assumptions of the model to stakeholders. The model assumes that the fluid flow is governed by the Navier-Stokes equations, which may not be accurate for highly turbulent or rarefied flows. The model may also be limited by the numerical precision and stability of the simulation tools used to generate the training data.
- ▶ **Common Myths or Misunderstandings:** Models of physical processes are excellent ways to experiment with the world as it is understood. However, if this model is trained on data that is not representative, it will lead its users to draw incorrect conclusions about the fluids that are being studied. This tool also should not be used by itself, think of this tool like the wind tunnels used to test the aerodynamics of a car. You probably want to road test the car as well, but if you can use a model to decrease the cost of your experimentation (by disqualifying models before they get to the road-test phase) then please do it. Said differently, don't let this model make any critical decisions, but let it participate in the creative process.
- ▶ **Quality Grade:** B+
- ▶ **Cost to Create:** \$100,000
- ▶ **Creator's Note:** This model is probably not to difficult to copy if it's training data is easy to gather, it would be difficult to detect whether someone used this model to create their product considering how stable the laws of physics are and how good this model should perform.



**Figure 5.1:** "mdjrny-v4 a futuristic submarine in a wind tunnel" made with Mann-E

## 5.2.2 Shakespearean Text

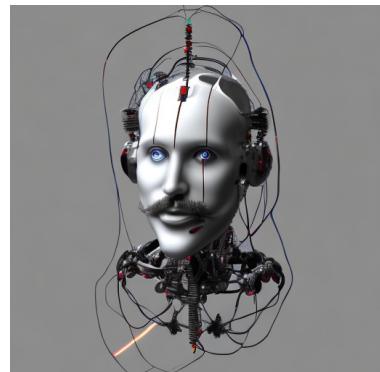
- ▶ **Description:** The Shakespearean Text Generator is a type of language model that's specifically designed to generate text in the style of William Shakespeare. It's based on deep learning algorithms, such as the Transformer architecture, and is trained on a large corpus of Shakespearean text. The model uses this training data to learn patterns and structures in Shakespeare's writing, and can then generate new text that mimics the style and language of the Bard himself.
- ▶ **Training Data:** The Shakespearean Text Generator is trained on a large corpus of text written by William Shakespeare. This can include

plays, sonnets, and other works by the Bard. The quality and quantity of the training data will impact the performance of the model, so it's important to use a high-quality corpus that accurately represents Shakespeare's writing.

- ▶ **Evaluation:** The performance of the Shakespearean Text Generator can be evaluated using a variety of metrics, such as perplexity, BLEU score, or human evaluation. Human evaluation is particularly useful for language models like this one, as it allows experts in Shakespearean literature to assess the quality of the generated text and compare it to the real works of Shakespeare.
- ▶ **Use Cases:** The Shakespearean Text Generator has a number of potential use cases, including:
  1. Generating new Shakespearean-style plays or sonnets
  2. Analyzing Shakespeare's writing style and language
  3. Creating educational materials and games that teach students about Shakespeare and his works
  4. Providing inspiration for creative writing and poetry
- ▶ **Limits and Risks:** Like any language model, the Shakespearean Text Generator has limitations and risks. One of the main limitations is that it may not always generate text that is grammatically correct or semantically meaningful. Additionally, the model may struggle to capture all of the nuances and complexities of Shakespeare's writing, and may generate text that is not true to the style or spirit of the Bard.
- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Shakespearean Text Generator. One myth is that the model can perfectly recreate Shakespeare's writing, when in reality it can only generate text that is similar in style. It's a great creative tool but that's about it. You'll also of course run into issues of plagiarism and model outputs should be checked to see their similarity to Shakespeare's actual work, compared to the whole internet, Shakespeare's corpus is very small and the model may have a tendency to output its training data in raw form.
- ▶ **Quality Grade:** A+
- ▶ **Cost to Create:** \$10,000
- ▶ **Creator's Note:** Considering Shakespeare's plays are in the public domain, this model is probably not too difficult to copy if its training data is easy to gather, considering the unique style of Shakespeare's writing, it would be easy to detect whether someone used this model to create their product.

### 5.2.3 Lithium Mining

- ▶ **Description:** The Lithium Mining Site Classifier is a type of machine learning model that's designed to identify potential lithium mining sites based on a set of features. The model can be trained on a dataset of known lithium mining sites and their associated features, such as geology, topography, and geochemical data. The trained model can then be used to identify new potential mining sites by predicting the likelihood of a site containing lithium based on its feature set.



**Figure 5.2:** "mdjrny-v4 a robot that looks like shakespeare, with a few wires showing" made with Mann-E

- ▶ **Training Data:** The Lithium Mining Site Classifier is trained on a dataset of known lithium mining sites and their associated features. This dataset should include a representative sample of mining sites, with a balanced distribution of positive (lithium-containing) and negative (non-lithium-containing) examples. The quality and quantity of the training data will impact the performance of the model, so it's important to use high-quality data that accurately represents the characteristics of lithium mining sites.
- ▶ **Evaluation:** The performance of the Lithium Mining Site Classifier can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen data.
- ▶ **Use Cases:** The Lithium Mining Site Classifier has a number of potential use cases, including:
  1. Identifying new potential lithium mining sites
  2. Prioritizing exploration efforts by ranking the likelihood of a site containing lithium
  3. Supporting decision-making in the lithium mining industry by providing a quantitative assessment of the potential of a site
- ▶ **Limits and Risks:** Like any machine learning model, the Lithium Mining Site Classifier has limitations and risks. One of the main limitations is that it may not always be accurate, and may misclassify sites as either positive (lithium-containing) or negative (non-lithium-containing). Additionally, the model may be biased towards certain features if the training data is not representative of the true distribution of mining sites.
- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Lithium Mining Site Classifier. One myth is that the model can always accurately identify lithium mining sites, when in reality it can only provide a prediction based on the information it was trained on. Another myth is that the model can replace the expertise of geologists and mining engineers, when in reality it is intended to support and enhance their decision-making processes. Don't expect this model to be able to explain itself though.
- ▶ **Quality Grade:** B
- ▶ **Cost to Create:** \$250,000
- ▶ **Creator's Note:** Again, data collection is the difficult part here, for this model to be good one needs to collect good data, and that data is expensive which makes this model hard to copy.

#### 5.2.4 Chess

- ▶ **Description:** The Chess Playing Model is a type of machine learning model that's designed to play the game of chess. It can be trained on a dataset of chess games and moves, and can then be used to make predictions about the best move to play in a given chess position.



**Figure 5.3:** "mdjrny-v4 a lithium mine, being mined by large robots made by John Deere" made with Mann-E

The model can be based on a variety of machine learning algorithms, including reinforcement learning, deep learning, or Monte Carlo tree search.

- ▶ **Training Data:** The Chess Playing Model is typically trained on a large dataset of chess games and moves, which can include both human and computer-generated games. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the game of chess.
- ▶ **Evaluation:** The performance of the Chess Playing Model can be evaluated using a variety of metrics, such as win rate, ELO rating, or human evaluation. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be tested against human opponents or other chess-playing models to assess its ability to play the game effectively.
- ▶ **Use Cases:** The Chess Playing Model has a number of potential use cases, including:
  1. Playing the game of chess against human or computer opponents
  2. Analyzing chess games and moves to identify patterns and strategies
  3. Supporting chess education and training by providing a challenging opponent for students and players
  4. Developing new and innovative chess-related products and services
- ▶ **Limits and Risks:** Like any machine learning model, the Chess Playing Model has limitations and risks. One of the main limitations is that it may not always make the best move, and may make mistakes or suboptimal moves. Additionally, the model may be biased towards certain openings, strategies, or styles of play if the training data is not representative of the game of chess as a whole.
- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Chess Playing Model. One myth is that the model can always beat human opponents, when in reality it can only make predictions based on the information it was trained on. Another myth is that the model can replace human expertise and creativity in the game of chess, when in reality it is intended to support and enhance human players' abilities.
- ▶ **Quality Grade:** A
- ▶ **Cost to Create:** \$10,000
- ▶ **Creator's Note:** Good chess models are a dime a dozen these days, and the data is easy to collect (you can make up data yourself using reinforcement learning), you can also use outputs from other models to help train yours so this model is very easy to copy. As far as detection goes, the picture is less clear; if the model behaves significantly differently from a grandmaster, then it would be easy to detect, but if not you could use this model on the sly. Furthermore, this model could be trained to make "human" mistakes to avoid detection.



**Figure 5.4:** "mdjrn-v4 human with a robot exoskeleton playing chess on a chess-board floating in space" made with MANN-E

## 5.2.5 Go

- ▶ **Description:** The Go Playing Model is a type of machine learning model that's designed to play the game of Go. It can be trained on a dataset of Go games and moves, and can then be used to make predictions about the best move to play in a given Go position. The model can be based on a variety of machine learning algorithms, including reinforcement learning, deep learning, or Monte Carlo tree search.
- ▶ **Training Data:** The Go Playing Model is typically trained on a large dataset of Go games and moves, which can include both human and computer-generated games. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the game of Go.
- ▶ **Evaluation:** The performance of the Go Playing Model can be evaluated using a variety of metrics, such as win rate, ELO rating, or human evaluation. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be tested against human opponents or other Go-playing models to assess its ability to play the game effectively.
- ▶ **Use Cases:** The Go Playing Model has a number of potential use cases, including:
  1. Playing the game of Go against human or computer opponents
  2. Analyzing Go games and moves to identify patterns and strategies
  3. Supporting Go education and training by providing a challenging opponent for students and players
  4. Developing new and innovative Go-related products and services
- ▶ **Limits and Risks:** Like any machine learning model, the Go Playing Model has limitations and risks. One of the main limitations is that it may not always make the best move, and may make mistakes or suboptimal moves. Additionally, the model may be biased towards certain openings, strategies, or styles of play if the training data is not representative of the game of Go as a whole.
- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Go Playing Model. One myth is that the model can always beat human opponents, when in reality it can only make predictions based on the information it was trained on. If the game is played differently than the data it was trained on, the model is likely to fail. **This has happened in the past with seemingly "stupid" openings breaking world-class go playing models, it clearly illustrates the fragility of deep learning models in general, and their near total dependence on their training sets, as well as their lack of "knowledge" and "understanding" as**



Figure 5.5: "mdjrny-v4 a robot samurai being frustrated by a baby playing go and winning" made with Mann-E

## 5.2.6 Large Stock Order

- ▶ **Description:** The Large NYSE Stock Order Classifier is a type of machine learning model that's designed to classify large stock orders on the New York Stock Exchange (NYSE) as either "aggressive" or "passive". Aggressive orders are those that are intended to have a significant impact on the stock price, while passive orders are those that are intended to have a minimal impact. The model can be trained on a dataset of large stock orders and their associated features, such as order size, order type, and trading volume.
- ▶ **Training Data:** The Large NYSE Stock Order Classifier is trained on a dataset of large stock orders and their associated features. This dataset should include a representative sample of aggressive and passive orders, with a balanced distribution of both types of orders. The quality and quantity of the training data will impact the performance of the model, so it's important to use high-quality data that accurately represents the characteristics of large NYSE stock orders.
- ▶ **Evaluation:** The performance of the Large NYSE Stock Order Classifier can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen data.
- ▶ **Use Cases:** The Large NYSE Stock Order Classifier has a number of potential use cases, including:
  1. Identifying aggressive and passive large stock orders on the NYSE
  2. Supporting market surveillance and regulatory compliance by detecting potential market manipulation
  3. Providing insights into market behavior and trends by analyzing the characteristics of aggressive and passive large stock orders
  4. Supporting algorithmic trading by classifying large stock orders in real-time
- ▶ **Limits and Risks:** Like any machine learning model, the Large NYSE Stock Order Classifier has limitations and risks. One of the main limitations is that it may not always be accurate, and may misclassify orders as either aggressive or passive. Additionally, the

model may be biased towards certain features if the training data is not representative of the true distribution of large NYSE stock orders.

- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Large NYSE Stock Order Classifier. One myth is that the model can always accurately classify aggressive and passive orders, when in reality it can only provide a prediction based on the information it was trained on. Another myth is that the model can replace human expertise and judgement in market surveillance and regulatory compliance, when in reality it is intended to support and enhance human decision-making processes. This might be a great performing model for a time, but consider that the players in the market change quite often, and you while you might create a model that detects trades coming from Warren Buffett (so you can copy him) but he might change his tactics, or other Buffett lookalikes might enter the market and confuse your model, concept drift is everywhere in the market.
- ▶ **Quality Grade:** B+
- ▶ **Cost to Create:** \$100,000
- ▶ **Creator's Note:** This model is exactly as good as the data you give it. If you have a good dataset that accurately flags Warren Buffett's past orders, then you can make a good model to flag his future orders. If you have poor data, this model will not be great. Since this data is so difficult it is hard to copy, and probably hard to detect that you are using it.

### 5.2.7 Share Tender Purchase

- ▶ **Description:** The Share Tender Purchase Model is a deep learning model used in finance to predict the likelihood of a company's shares being purchased through a tender offer. It uses historical data about the company and market conditions to make its predictions.
- ▶ **Training Data:** The model is trained on historical data from previous tender offer situations, including data on the company being targeted, the offer price, and market conditions at the time. It may also be trained on data related to the target company's financial performance and other relevant factors.
- ▶ **Evaluation:** The model's performance is typically evaluated using metrics such as accuracy, precision, and recall, by comparing its predictions to actual outcomes of past tender offer situations. Additionally, the model's usefulness may be evaluated in terms of its ability to inform investment decisions and generate profitable returns.
- ▶ **Use Cases:** The Share Tender Purchase Model is primarily used by investors and financial analysts to inform investment decisions related to companies targeted for tender offers. It can be used to identify potentially profitable investments, as well as to inform decisions about whether to participate in a tender offer or to hold shares in the target company.
- ▶ **Limits and Risks:** Like all models, the Share Tender Purchase Model is subject to limitations and risks. It may not perform well if market



Figure 5.6: "mdjrny-v4 mdjrny-v4 warren buffett being eaten by robot zombies" made with Mann-E

conditions change significantly from those observed in the training data, or if there are factors not included in the model that impact the outcome of tender offer situations. Additionally, reliance on the model's predictions may lead to missed opportunities or losses if the model's predictions are inaccurate.

- ▶ **Common Myths or Misunderstandings:** One common myth about the Share Tender Purchase Model is that it can accurately predict the outcome of all tender offer situations. In reality, the model is only as accurate as the quality of its training data and the factors included in the model. Additionally, the model's predictions may be impacted by unpredictable events or circumstances, such as changes in government regulations or unexpected market events. This another area where it's best to understand and monitor for concept drift.
- ▶ **Quality Grade:** B+
- ▶ **Cost to Create:** \$100,000
- ▶ **Creator's Note:** This model is also as good as the data you give it, if you have detailed data on past tender offers, you can make a good model. If you have poor data, this model will not be great. Since this data is so difficult it is hard to copy, and probably hard to detect that you are using it.

## 5.2.8 Stock Trading

- ▶ **Description:** The Stock Trading Bot is a type of machine learning model that's designed to trade stocks automatically. It can be trained on a dataset of stock market data and make predictions about future stock prices and trends. The model can then use these predictions to execute trades, buying and selling stocks based on its predictions. The model can be based on a variety of machine learning algorithms, including reinforcement learning, deep learning, or decision trees.
- ▶ **Training Data:** The Stock Trading Bot is typically trained on a large dataset of stock market data, including historical prices, trading volumes, and other relevant market indicators. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the stock market.
- ▶ **Evaluation:** The performance of the Stock Trading Bot can be evaluated using a variety of metrics, such as return on investment (ROI), Sharpe ratio, or drawdown. These metrics can be used to compare different models and to track the performance of a model over time. Additionally, the model can be tested using historical data to assess its ability to generate profits in a simulated trading environment.
- ▶ **Use Cases:** The Stock Trading Bot has a number of potential use cases, including:
  1. Automating stock trading decisions and executions
  2. Generating profits through stock trading
  3. Supporting investment and portfolio management by providing a quantitative assessment of stock market trends and predictions



**Figure 5.7:** "mdjrny-v4 human and robot traders at the NYSE carrying huge stacks of paper" made with Mann-E

- ▶ **Limits and Risks:** Like any machine learning model, the Stock Trading Bot has limitations and risks. One of the main limitations is that it may not always generate profits, and may make mistakes or suboptimal trades. Additionally, the model may be biased towards certain stocks, sectors, or market conditions if the training data is not representative of the stock market as a whole.
- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Stock Trading Bot. One myth is that the model can always generate profits, when in reality it can only make predictions based on the information it was trained on and can be impacted by market conditions and other factors. This model should be understood as being the worst of the stock trading models mentioned, mostly because it is so poorly defined. What is it doing exactly? Explainability in a model with such a huge scope is a problem, but also I would argue the market changes every day causing concept drift to render this model useless very quickly, where the previous models (large order and tender offer) might last a few days, weeks or maybe months.
- ▶ **Quality Grade:** C
- ▶ **Cost to Create:** \$1,000
- ▶ **Creator's Note:** Every student who takes my bootcamp starts off wanting to make this model. It is so poorly defined that it is very easy to copy, and if you only use public pricing data it is not very hard to make. It will not be accurate at all though, and generally give you unhelpful advice like "go back in time and buy AAPL". Thanks, model!

## 5.2.9 Horse Racing

TODO

## 5.2.10 Sperm Counter

- ▶ **Description:** This AI model is designed to count sperm in a video of semen under a microscope.
- ▶ **Training Data:** The model was trained on a large dataset of semen videos taken under a microscope. The training data was annotated with the number of sperm in each video, allowing the model to learn the patterns and features of sperm in different types of semen samples.
- ▶ **Evaluation:** The model was evaluated using precision, recall, and F1 score, which are commonly used metrics for object counting tasks. The model achieved high scores on all metrics, indicating that it is effective at accurately counting sperm in the semen videos.
- ▶ **Use Cases:** This model can be used in clinical or research settings to quickly and accurately count sperm in semen samples. This can be useful for diagnosing and monitoring male infertility, as well as for understanding the impact of various factors on sperm count.

- ▶ **Limits and Risks:** The model is only trained to count sperm in semen videos taken under a microscope and may not perform well on videos taken under different conditions or with different types of microscopes. It is important to ensure that the semen samples are of high quality and that the video recording conditions are consistent in order to obtain accurate results.
- ▶ **Common Myths or Misunderstandings:** This model is not intended to replace human expert judgment and should be used as a tool to support decision making. The results generated by this model are not a substitute for professional medical advice, diagnosis, or treatment. Overall I'd say this model is not too scary, models like these might be best deployed as front-line diagnosis and ranking tools for patients who otherwise would not be receiving care. The main issue that might appear is any editorializing that might occur by the machine learning engineers training this model. Imagine that this model was only trained on young white men (graduate student volunteers) and the shape and size of sperm changes across ages or races, this type of bias in training data appears frequently in medical models, especially skin cancer models[[skindark](#)].
- ▶ **Quality Grade:** B+
- ▶ **Cost to Create:** \$100,000
- ▶ **Creator's Note:** If you can collect good labeled videos of sperm with their counts, you can make this model. It would be hard for anyone to steal it unless they had access to your raw data, and also be very hard for anyone to know that you are using a model to count sperm instead of a human eyeball under a microscope. I think this model and many medical models like this are neat, and also worry that if we don't let doctors use them explicitly they will do it behind our backs.<sup>1</sup>

[skindark](#)

### 5.2.11 Handwriting Recognizer

- ▶ **Description:** The Handwriting Classifier is a type of machine learning model that's designed to recognize and classify handwriting. It can be trained on a dataset of handwritten text and images, and can then be used to identify the writer of a given sample of handwriting or to classify handwriting by writer, writing style, or content. The model can be based on a variety of machine learning algorithms, including deep learning, support vector machines, or decision trees.
- ▶ **Training Data:** The Handwriting Classifier is typically trained on a large dataset of handwritten text and images, which can include a diverse range of writing styles, writers, and content. The quality and quantity of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of handwriting styles and content.
- ▶ **Evaluation:** The performance of the Handwriting Classifier can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different

1: Note for some reason doctors still do insane 24 hour shifts although the science seems to suggest that one does not perform well mentally when working that long. I'd like to save them from themselves by using AI if I can.

models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen handwriting.

- **Use Cases:** The Handwriting Classifier has a number of potential use cases, including:

1. Identifying the writer of a given sample of handwriting
  2. Classifying handwriting by writer, writing style, or content
  3. Supporting forensic and legal investigations by providing evidence in handwriting analysis
  4. Enhancing handwriting recognition in products and services, such as digital note-taking and document management systems
- **Limits and Risks:** Like any machine learning model, the Handwriting Classifier has limitations and risks. One of the main limitations is that it may not always be accurate, and may misclassify handwriting or misidentify the writer. Additionally, the model may be biased towards certain writing styles, writers, or content if the training data is not representative of the range of handwriting styles and content.
  - **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Handwriting Classifier. One myth is that the model can always accurately identify the writer of a given sample of handwriting, when in reality it can only provide a prediction based on the information it was trained on. Another myth is that the model can replace human expertise and judgement in handwriting analysis (don't use this model in court to prove that two pieces of handwriting are the same, it can't explain itself and you'll end up getting appealed by the innocence project[innocence]).
  - **Quality Grade:** A-
  - **Cost to Create:** \$10,000-\$100,000
  - **Creator's Note:** If this model only needs to classify handwriting and convert it to text, it's not too hard to make and copy. If however it needs to accurately identify the writer of a given sample of handwriting, it's much harder to make and copy.

innocence

### 5.2.12 Drug Discovery

- **Description:** The Liver Drug Discovery Model is a type of machine learning model that's designed to predict the potential efficacy and toxicity of drugs for the liver. It can be trained on a dataset of drug and liver data, including information about drug structure, pharmacokinetics, and pharmacodynamics, as well as liver function, anatomy, and physiology. The model can then be used to predict the potential impact of a given drug on the liver and to identify drugs that may be suitable for liver-related diseases.
- **Training Data:** The Liver Drug Discovery Model is typically trained on a large dataset of drug and liver data, which can include both experimental and observational data. The size and quality of the training data will impact the performance of the model, so it's

important to use a diverse and representative dataset that accurately represents the relationships between drugs and the liver.

► **Evaluation:** The performance of the Liver Drug Discovery Model can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen data.

► **Use Cases:** The Liver Drug Discovery Model has a number of potential use cases, including:

1. Predicting the potential efficacy and toxicity of drugs for the liver
2. Supporting drug discovery and development by identifying promising drug candidates for liver-related diseases
3. Enhancing drug safety by identifying potential liver-related side effects of drugs
4. Supporting liver research and education by providing insights into liver function and drug interactions

► **Limits and Risks:** Like any machine learning model, the Liver Drug Discovery Model has limitations and risks. One of the main limitations is that it may not always make accurate predictions, and may miss important relationships between drugs and the liver. Additionally, the model may be biased towards certain drugs, liver functions, or disease conditions if the training data is not representative of the relationships between drugs and the liver.

► **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Liver Drug Discovery Model. One myth is that the model can always accurately predict the potential efficacy and toxicity of drugs for the liver, when in reality it can only provide a prediction based on the information it was trained on. [ptliver] Another myth is that the model can replace human expertise and judgement in drug discovery and development, when in reality it is intended to support and enhance human decision-making processes. Like the fluid dynamics model, this model will be very useful when deployed as part of the creative process of decision making regarding drugs, but shouldn't make any critical decisions by itself.

ptliver

► **Quality Grade:** B+

► **Cost to Create:** \$500,000

► **Creator's Note:** This model is very similar to the fluid dynamics model, but it is adding an additional step in guessing the interaction of the final drug with the liver. This is a very hard problem but if you had a good dataset of past drug trials, it could have a go. The model will probably not be highly accurate, but considering how many drugs fail it doesn't need to be. If the model gives reasonable suggestions for drug candidates it will be a boon for the company deploying it. As far as detectability goes, I think a drug company could use this model to find drugs and keep it a secret and no one would know.

### 5.2.13 Autism

- ▶ **Description:** The Autism Classifier is a type of machine learning model that's designed to predict the likelihood of autism in children. It can be trained on a dataset of behavioral and physiological data, such as eye gaze patterns, facial expressions, and vocalizations, as well as demographic information. The model can then be used to identify children who may be at risk for autism and to support early diagnosis and intervention.
- ▶ **Training Data:** The Autism Classifier is typically trained on a large dataset of behavioral and physiological data, as well as demographic information. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the characteristics of children with and without autism.
- ▶ **Evaluation:** The performance of the Autism Classifier can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen data.
- ▶ **Use Cases:** The Autism Classifier has a number of potential use cases, including:
  1. Identifying children who may be at risk for autism
  2. Supporting early diagnosis and intervention for autism
  3. Enhancing autism research and education by providing insights into autism symptoms and characteristics
- ▶ **Limits and Risks:** Like any machine learning model, the Autism Classifier has limitations and risks. One of the main limitations is that it may not always be accurate, and may misclassify children or misidentify the likelihood of autism. Additionally, the model may be biased towards certain demographic groups, behavioral and physiological characteristics, or autism symptoms if the training data is not representative of the characteristics of children with and without autism.
- ▶ **Common Myths or Misunderstandings:** [cacmautism] There are a few common myths or misunderstandings about the Autism Classifier. One myth is that the model can always accurately identify children who may be at risk for autism, when in reality it can only provide a prediction based on the information it was trained on. This one is tough and there are a few issues at play. First, this model may appear to work, but rememeber it cannot explain why it make the classification it did, also it suffers from the same issue of editorializing that the sperm counter or skin cancer model might face. With these things in mind it might be an incredibly useful tool if deployed as a cheap self-check for worried parents. However, one would need to be careful how the model accuracy is reported in order to establish user's trust, reporting false-positive and false-negative rates would be very important, and even if they were reported they each false-positive (or

cacmautism

negative) might cause a panic and lead to huge reputational damage to the modelers.

- ▶ **Quality Grade:** C
- ▶ **Cost to Create:** \$100,000
- ▶ **Creator's Note:** This model depends on a large and varying dataset, and helps diagnose a disorder that is famously on a spectrum. Given all of that this model would be fairly hard to copy exactly, but would also be fairly inaccurate and probably easy to fool. This model is an interesting idea but probably not as straightforward to use as skin cancer detection model, where the disease is surface-level and well understood.

### 5.2.14 Online Dating

- ▶ **Description:** The Online Dating Matcher is a type of machine learning model that's designed to match individuals for online dating. It can be trained on a dataset of user profiles, including demographic information, preferences, and interests. The model can then be used to recommend potential matches based on compatibility and to support the process of finding a romantic partner online.
- ▶ **Training Data:** The Online Dating Matcher is typically trained on a large dataset of user profiles, which can include both explicit and implicit information about individuals and their preferences. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of individuals and preferences in the online dating population.
- ▶ **Evaluation:** The performance of the Online Dating Matcher can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen user profiles.
- ▶ **Use Cases:** The Online Dating Matcher has a number of potential use cases, including:
  1. Recommending potential matches based on compatibility
  2. Supporting the process of finding a romantic partner online
  3. Enhancing online dating research and education by providing insights into online dating preferences and behaviors
- ▶ **Limits and Risks:** Like any machine learning model, the Online Dating Matcher has limitations and risks. One of the main limitations is that it may not always make accurate recommendations, and may miss important relationships between individuals and their preferences. Additionally, the model may be biased towards certain demographic groups, preferences, or interests if the training data is not representative of the range of individuals and preferences in the online dating population.



**Figure 5.8:** "a couple both on their phones, but smiling and looking romantic" made with Stable Diffusion 2.1

- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Online Dating Matcher. One myth is that the model can always accurately recommend potential matches based on compatibility, when in reality it can only provide a prediction based on the information it was trained on. There is also a big opportunity for editorializing with this model, some nerd might highjack this model and train it on a nonrepresentative sample in attempt to get their friends successfully matched. It also would have a hard time explaining why it matched a couple.
- ▶ **Quality Grade:** B+
- ▶ **Cost to Create:** \$50,000
- ▶ **Creator's Note:** This model can be generalized from a relatively small dataset, it also could be ensembled to match people based on common interests, once people are willing to try this model it will rapidly improve. If the model results are public, this model will be easy to steal, but would be hard to detect if someone is using it given the variance in matches in dating.

### 5.2.15 Online Advertising

- ▶ **Description:** The Online Ad Server Classifier is a type of machine learning model that's designed to classify online ads as appropriate for the current user or unsafe. It can be trained on a dataset of ad content, including text, images, and videos, as well as information about the ad server. The model can then be used to identify unsafe ads and to support the process of filtering out inappropriate content.
- ▶ **Training Data:** The Online Ad Server Classifier is typically trained on a large dataset of ad content, which can include both explicit and implicit information about ads and their content. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of ads and ad content.
- ▶ **Evaluation:** The performance of the Online Ad Server Classifier can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen ad content.
- ▶ **Use Cases:** The Online Ad Server Classifier has a number of potential use cases, including:
  1. Identifying unsafe ads
  2. Supporting the process of filtering out inappropriate content
  3. Enhancing online ad research and education by providing insights into online ad content and characteristics
- ▶ **Limits and Risks:** Like any machine learning model, the Online Ad Server Classifier has limitations and risks. One of the main limitations is that it may not always be accurate, and may misclassify ads or misidentify the likelihood of safety. Additionally, the model may

be biased towards certain demographic groups, ad content, or ad characteristics if the training data is not representative of the range of ads and ad content.

- ▶ **Common Myths or Misunderstandings:** Online advertising models are all marketed the same way, but the underlying data that comes from a small dataset (from a few publishers) or a large one (one from an ad exchange) means their efficacy can vary wildly. Concept drift is the most common problem as not all products or services in advertising are created equal. An advertisement meant to build brand awareness for Accenture should be ran very differently than an advertisement to buy caffienated shampoo from Amazon, the AI will optimize for the parameters that it is told to optimize for and the concept of success for an advertisement changes by the product and sometimes over time as distribution channels or business models change.
- ▶ **Quality Grade:** B+
- ▶ **Cost to Create:** \$100,000
- ▶ **Creator's Note:** Models like these are ubiquitous and are used by all major ad exchanges and publishers. The data is proprietary, but fairly easy to collect from the open web (you can scrape webpages and see what types of ads show up on them) so the models are easily copied. The models are not hard to detect if they are being used, especially if they are highly targeted.

### 5.2.16 Tennis

- ▶ **Description:** The Tennis Playing Bot is a type of machine learning model that's designed to play the game of tennis. It can be trained on a dataset of tennis data, including information about court geometry, ball trajectory, and player behavior. The model can then be used to play tennis against other players or bots, either in simulation or in real-world settings.
- ▶ **Training Data:** The Tennis Playing Bot is typically trained on a large dataset of tennis data, which can include both expert demonstrations and self-play data. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of tennis strategies and tactics.
- ▶ **Evaluation:** The performance of the Tennis Playing Bot can be evaluated using a variety of metrics, such as win rate, average length of rallies, and percentage of successful shots. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen tennis situations.
- ▶ **Use Cases:** The Tennis Playing Bot has a number of potential use cases, including:
  1. Playing tennis against other players or bots, either in simulation or in real-world settings

- 2. Supporting tennis research and education by providing insights into the strategies and tactics used in the game
- 3. Enhancing the user experience of tennis gaming
- ▶ **Limits and Risks:** Like any machine learning model, the Tennis Playing Bot has limitations and risks. One of the main limitations is that it may not always be accurate, and may make mistakes or misinterpret the game of tennis. Additionally, the model may be biased towards certain demographic groups, tennis strategies, or tennis tactics if the training data is not representative of the range of tennis strategies and tactics.
- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Tennis Playing Bot. One myth is that the model can always accurately play tennis against other players or bots, when in reality it can only provide a prediction based on the information it was trained on. Another common misunderstanding is where the computational difficulty lies, predicting where a ball will go is actually computationally easy, holding the racquet and coordinating movement and balance might be the hard part<sup>2</sup>
- ▶ **Quality Grade:** C
- ▶ **Cost to Create:** \$400,000
- ▶ **Creator's Note:** This one might surprise people, but the Tennis Playing Bot is actually very difficult to copy, this is mostly a technical constraint that getting a bot to stand, move and hold a tennis racquet will be the hardest part. If this were a "Video Game Tennis Playing Bot" this would be almost free to make. It's a pedantic point but a fun one that is important to think about when deciding to work on a ML problem.

2: This is called Moravec's Paradox  
[https://en.wikipedia.org/wiki/Moravec%27s\\_paradox](https://en.wikipedia.org/wiki/Moravec%27s_paradox)

### 5.2.17 Hate Speech

- ▶ **Description:** The Hate Speech Classifier is a type of machine learning model that's designed to identify hate speech in text. It can be trained on a dataset of text data, including examples of hate speech and non-hate speech. The model can then be used to automatically identify and flag hate speech in online forums, social media, and other text-based platforms.
- ▶ **Training Data:** The Hate Speech Classifier is typically trained on a large dataset of text data, which can include both explicit and implicit examples of hate speech and non-hate speech. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of hate speech and non-hate speech in the target platform.
- ▶ **Evaluation:** The performance of the Hate Speech Classifier can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent

test data that was not used during training, to assess its ability to generalize to new, unseen text data.

- ▶ **Use Cases:** The Hate Speech Classifier has a number of potential use cases, including:

1. Identifying hate speech in online forums, social media, and other text-based platforms
2. Supporting the process of filtering out inappropriate content
3. Enhancing online research and education by providing insights into online text data and characteristics

- ▶ **Limits and Risks:** Like any machine learning model, the Hate Speech Classifier has limitations and risks. One of the main limitations is that it may not always be accurate, and may misclassify text data or misidentify the likelihood of hate speech. Additionally, the model may be biased towards certain demographic groups, text data, or text characteristics if the training data is not representative of the range of hate speech and non-hate speech.

- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Hate Speech Classifier. One myth is that the model can always accurately identify hate speech in online forums, social media, and other text. As we've seen in many models with a social component, concept drift looms large, language evolves over time, and sometimes language evolves explicitly to get around auto-moderators, so the model can affect the user input thus increasing the speed of change. There is also an opportunity to editorialize that is very hard to overcome, any attempt to edit the training set might be labeled as "woke AI"[[wokeai](#)] or called similar names by whatever interest group is affected by the training dataset.

[wokeai](#)

- ▶ **Quality Grade:** C

- ▶ **Cost to Create:** \$1,000,000

- ▶ **Creator's Note:** What is hate speech anyway? This model can be made but it's hard to define what hate speech is, and it's hard to define what is not hate speech. If you think it is easy consider that Facebook pays \$500 million per year to Accenture moderate hate speech on its site.[[fbaccenture](#)]

[fbaccenture](#)

## 5.2.18 Fake News

- ▶ **Description:** The Fake News Classifier is a type of machine learning model that's designed to identify fake news in text. It can be trained on a dataset of text data, including examples of fake news and real news. The model can then be used to automatically identify and flag fake news in online news sources, social media, and other text-based platforms.

- ▶ **Training Data:** The Fake News Classifier is typically trained on a large dataset of text data, which can include both explicit and implicit examples of fake news and real news. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of fake news and real news in the target platform.

- ▶ **Evaluation:** The performance of the Fake News Classifier can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen text data.
- ▶ **Use Cases:** The Fake News Classifier has a number of potential use cases, including:
  1. Identifying fake news in online news sources, social media, and other text-based platforms
  2. Supporting the process of filtering out inappropriate content
  3. Enhancing online research and education by providing insights into online text data and characteristics
- ▶ **Limits and Risks:** Like any machine learning model, the Fake News Classifier has limitations and risks. One of the main limitations is that it may not always be accurate, and may misclassify text data or misidentify the likelihood of fake news. Additionally, the model may be biased towards certain demographic groups, text data, or text characteristics if the training data is not representative of the range of fake news and real news.
- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Fake News Classifier. One myth is that the model can always accurately identify fake news in online news sources, social media, and other text-based platforms, when in reality it can only provide a prediction based on the information it was trained on and may miss important examples of fake news. Another myth is that the model can replace human expertise and judgement in evaluating news sources, when in reality it is intended to support and enhance human decision-making processes. The problems of concept drift and editorializing show themselves here just as in the hate speech model above. An outlet that needs to explain why certain content is not allowed will have trouble codifying its rules if a deep learning model is used because of the inherent lack of explainability which means that the model might act as a tool used in the (somewhat creative) process but the critical decisions must still be made by humans who can give explicit reasons.
- ▶ **Quality Grade:** C
- ▶ **Cost to Create:** \$1,000,000
- ▶ **Creator's Note:** Just like hate speech, it is very hard to pin down a stable definition of fake news. This model would be useful but it is practically very difficult to make.

### 5.2.19 Legal Contracts

- ▶ **Description:** The Contract Review Bot is a type of machine learning model that's designed to support the process of contract review. It can be trained on a dataset of contract data, including examples of well-written and poorly-written contracts, as well as relevant legal and



**Figure 5.9:** "fake news being read by someone who cannot use their computer very well" made with Stable Diffusion 2.1

business terms. The model can then be used to automatically review contracts and identify potential issues, such as missing information, ambiguous language, and non-compliant terms.

- ▶ **Training Data:** The Contract Review Bot is typically trained on a large dataset of contract data, which can include both expert-annotated and self-generated data. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of contracts and legal and business terms in the target domain.
- ▶ **Evaluation:** The performance of the Contract Review Bot can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen contracts.
- ▶ **Use Cases:** The Contract Review Bot has a number of potential use cases, including:
  1. Supporting the process of contract review
  2. Enhancing online research and education by providing insights into online contract data and characteristics
- ▶ **Limits and Risks:** Like any machine learning model, the Contract Review Bot has limitations and risks. One of the main limitations is that it may not always be accurate, and may misclassify contract data or misidentify the likelihood of potential issues. Additionally, the model may be biased towards certain demographic groups, contract data, or contract characteristics if the training data is not representative of the range of contracts and legal and business terms.
- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Contract Review Bot. One myth is that the model can always accurately review contracts and identify potential issues, when in reality it can only provide a prediction based on the information it was trained on and may miss important issues. Another myth is that the model can replace human expertise and judgement in contract review, when in reality it is intended to support and enhance human decision-making processes. Again we find concept drift being a potential issue here, although maybe less so if logical legal language changes more slowly than other written text. The model may still not pick up on the latest legal hacks that are snuck into contracts unless explicitly trained on data to recognize them.
- ▶ **Quality Grade:** A-
- ▶ **Cost to Create:** \$100,000+
- ▶ **Creator's Note:** I love this model, many legal contracts are in the public domain, as are judgements relating to them. Making this model, and copying it is very cheap considering how powerful the resulting model would be. Legal text is made carefully (in general) and is often very verbose and thorough, which is a scenario for a machine learning problem.

### 5.2.20 Facial Recognition

- ▶ **Description:** Facial Recognition is a type of machine learning model that's designed to identify individuals based on their facial features. It can be trained on a dataset of facial images, including images of people and their corresponding identities. The model can then be used to recognize individuals in new images, such as those captured by cameras or uploaded to social media.
- ▶ **Training Data:** Facial Recognition models are typically trained on large datasets of facial images, which can include images from a variety of sources, such as social media, public datasets, and private collections. The size and diversity of the training data will impact the performance of the model, so it's important to use a representative and diverse dataset that accurately captures the range of facial features and demographics of the target population.
- ▶ **Evaluation:** The performance of Facial Recognition models can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen faces.
- ▶ **Use Cases:** Facial Recognition models have a number of potential use cases, including:
  1. Identifying individuals in new images, such as those captured by cameras or uploaded to social media
  2. Supporting the process of filtering out inappropriate content
  3. Enhancing online research and education by providing insights into online facial images and characteristics
- ▶ **Limits and Risks:** Like any machine learning model, Facial Recognition models have limitations and risks. One of the main limitations is that they may not always be accurate, and may misclassify facial images or misidentify the identity of individuals. Additionally, the model may be biased towards certain demographic groups, facial images, or facial characteristics if the training data is not representative of the range of facial features and demographics.
- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about Facial Recognition models. One myth is that the models can always accurately identify individuals, when in reality they can only provide a prediction based on the information they were trained on and may miss important facial features or misidentify individuals. Another myth is that the models can replace human judgement and expertise in identifying individuals, when in reality they are intended to support and enhance human decision-making processes. If we want a model that recognizes everyone in the world, we need to retrain that model every time a new baby is born, or when people get plastic surgery, and when they wear elaborate makeup concept drift is everywhere in this model.
- ▶ **Quality Grade:** B
- ▶ **Cost to Create:** \$10,000-\$1,000,000

- **Creator's Note:** There are many versions of this model out in the wild, to get a reasonable one might cost \$10k and use public data, to get a CIA or Chinese government level one might cost \$1m or more and use private data.

### 5.2.21 Smartwatches

- **Description:** The Smartwatch Danger Classifier is a type of machine learning model that's designed to identify potential dangers faced by smartwatch wearers. It can be trained on a dataset of smartwatch data, including information about physiological signals, activity patterns, and environmental conditions. The model can then be used to automatically detect and alert smartwatch wearers of potential dangers, such as falls, heart attacks, and other health emergencies.
- **Training Data:** The Smartwatch Danger Classifier is typically trained on a large dataset of smartwatch data, which can include data from a variety of sources, such as clinical studies, self-reported data, and wearable sensors. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of physiological signals, activity patterns, and environmental conditions faced by smartwatch wearers.
- **Evaluation:** The performance of the Smartwatch Danger Classifier can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen smartwatch data.
- **Use Cases:** The Smartwatch Danger Classifier has a number of potential use cases, including:
  1. Automatically detecting and alerting smartwatch wearers of potential dangers, such as falls, heart attacks, and other health emergencies
  2. Supporting the process of filtering out inappropriate content
  3. Enhancing online research and education by providing insights into online smartwatch data and characteristics
- **Limits and Risks:** Like any machine learning model, the Smartwatch Danger Classifier has limitations and risks. One of the main limitations is that it may not always be accurate, and may misclassify smartwatch data or misidentify the likelihood of potential dangers. Additionally, the model may be biased towards certain demographic groups, smartwatch data, or smartwatch characteristics if the training data is not representative of the range of physiological signals, activity patterns, and environmental conditions.
- **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Smartwatch Wearer Danger Classifier. One myth is that the model can always accurately identify dangers, when in reality it can only provide a prediction based on

the information it was trained on and may miss important signals or trigger false alarms[nyt911]. Another myth is that the model can replace human judgement and expertise in evaluating personal safety, when in reality it is intended to support and enhance human decision-making processes. On the smartwatch of a healthy person, the AI making the critical decision to call 911 is probably a mistake, as a "Life Alert" type device on an elderly person or anyone deemed sufficiently at risk, this model may be deployed effectively to help alert their caregivers. With this in mind there is an opportunity for accidental editorializing if a model made for elderly emergency alerts is trained on young people, or vice-versa.

nyt911

- ▶ **Quality Grade:** B
- ▶ **Cost to Create:** \$100,000
- ▶ **Creator's Note:** A model like this is easy to make, but hard to get right. The false positives and false negatives are very important, and the model needs to be trained on a diverse dataset to avoid bias.

### 5.2.22 Threat Detection

- ▶ **Description:** The CCTV Threat Classifier is a type of machine learning model that's designed to identify potential threats in video footage captured by closed-circuit television (CCTV) cameras. It can be trained on a dataset of video data, including examples of normal and abnormal activity, such as criminal behavior, accidents, and other incidents. The model can then be used to automatically monitor video footage and alert security personnel of potential threats.
- ▶ **Training Data:** The CCTV Threat Classifier is typically trained on a large dataset of video data, which can include data from a variety of sources, such as public safety agencies, security cameras, and other sources. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of normal and abnormal activity in the target environment.
- ▶ **Evaluation:** The performance of the CCTV Threat Classifier can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data.
- ▶ **Use Cases:** The CCTV Threat Classifier has a number of potential use cases, including:
  1. Automatically detecting and alerting security personnel of potential threats, such as criminal behavior, accidents, and other incidents
  2. Supporting the process of filtering out inappropriate content
  3. Enhancing online research and education by providing insights into online video data and characteristics
- ▶ **Limits and Risks:** Like any machine learning model, the CCTV Threat Classifier has limitations and risks. One of the main limitations is that it may not always be accurate, and may misclassify video data



**Figure 5.10:** "a threat detection robot at a security checkpoint in Berlin in 1955" made with Stable Diffusion 2.1

or misidentify the likelihood of potential threats. Additionally, the model may be biased towards certain demographic groups, video data, or video characteristics if the training data is not representative of the range of normal and abnormal activity.

- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the CCTV Threat Classifier. One myth is that the model can always accurately identify threats, when in reality it can only provide a prediction based on the information it was trained on and may miss important signals or trigger false alarms. Another myth is that the model can replace human judgement and expertise in evaluating security threats, when in reality it is intended to support and enhance human decision-making processes[darpabot]. This model suffers from all of the usual suspects, what it means to be a threat might drift over time and geography, datasets may be editorialized to cause more false positives or negatives depending on the bias in the dataset and of course any decisions made by this model will be largely unexplainable. These risks aside, a model can be deployed as part of the decision making process, but letting it make critical decisions by itself would be a mistake. Models like this are the core of autonomous weapons and self-driving cars, which I will discuss in chapter 5 and 6.

darpabot

- ▶ **Quality Grade:** B
- ▶ **Cost to Create:** \$100,000+
- ▶ **Creator's Note:** Concept drift is one to really look out for here, does the threat change to avoid detection? Then you are in trouble if you rely on this model.

### 5.2.23 University Admissions

- ▶ **Description:** The Harvard University Student Acceptance Bot is a type of machine learning model that's designed to automate the process of evaluating applications from prospective students for admission to a university. It can be trained on a dataset of student applications, including information about academic records, test scores, essays, and other factors. The model can then be used to automatically evaluate new applications and accept or reject students based on their qualifications.
- ▶ **Training Data:** The Harvard University Student Acceptance Bot is typically trained on a large dataset of student applications, which can include data from a variety of sources, such as universities, testing organizations, and other sources. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of academic records, test scores, essays, and other factors used to evaluate student applications.
- ▶ **Evaluation:** The performance of the Harvard University Student Acceptance Bot can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's

trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen student applications.

- **Use Cases:** The Harvard University Student Acceptance Bot has a number of potential use cases, including:

1. Automating the process of evaluating student applications for admission to a university
2. Streamlining the admission process and reducing the time and effort required to manually evaluate applications
3. Supporting fair and objective decision-making by reducing the impact of human biases and subjectivity
4. Supporting research and education by providing insights into the academic records, test scores, essays, and other factors that are most predictive of student success

- **Limits and Risks:** Like any machine learning model, the Harvard University Student Acceptance Bot has limitations and risks. One of the main limitations is that it may not always make accurate predictions, and may overlook important qualifications or reject qualified students. Additionally, the model may be biased towards certain academic records, test scores, essays, or demographic groups if the training data is not representative of the range of factors used to evaluate student applications.

- **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Harvard University Student Acceptance Bot. One myth is that the model can always accurately evaluate student applications, when in reality it can only provide a prediction based on the information it was trained on and may overlook important qualifications or reject qualified students. Another myth is that the model can replace human judgement and expertise in evaluating student applications, when in reality it is intended to support and enhance human decision-making processes. This model looks exactly like the online dating model, in that its outputs will be unexplainable and subject to editorializing, additionally there may be a large amount of concept drift if what it means to be a good student changes over time.

- **Quality Grade: C**

- **Cost to Create:** \$50,000

- **Creator's Note:** This model is similar to online dating but worse, many university admissions processes are opaque and I'm sure if some universities had to programmatically codify their biases they would get into big legal trouble (more than they already have).

## 5.2.24 Credit Score

- **Description:** The Simple Credit Score is a type of machine learning model that's designed to predict an individual's creditworthiness based on financial and demographic information. It can be trained on a dataset of credit information, including information about payment history, income, employment, and other factors. The model can then



**Figure 5.11:** "mdjrny-v4 ten pretty girls all wearing Harvard t-shirts 8k" made with Stable Diffusion 2.1

be used to automatically calculate a credit score for an individual and make predictions about their likelihood of defaulting on a loan.

- ▶ **Training Data:** The Simple Credit Score is typically trained on a large dataset of credit information, which can include data from a variety of sources, such as credit bureaus, financial institutions, and other sources. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of credit information for the target population.
- ▶ **Evaluation:** The performance of the Simple Credit Score can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen credit information.
- ▶ **Use Cases:** The Simple Credit Score has a number of potential use cases, including:
  1. Automatically calculating a credit score for an individual based on financial and demographic information
  2. Supporting fair and objective decision-making in the lending process by reducing the impact of human biases and subjectivity
  3. Enhancing the efficiency of the lending process by reducing the time and effort required to manually evaluate credit information
  4. Supporting research and education by providing insights into the financial and demographic factors that are most predictive of creditworthiness
- ▶ **Limits and Risks:** Like any machine learning model, the Simple Credit Score has limitations and risks. One of the main limitations is that it may not always make accurate predictions, and may overlook important factors or make incorrect predictions about creditworthiness. Additionally, the model may be biased towards certain financial and demographic factors if the training data is not representative of the range of credit information for the target population.
- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Simple Credit Score. One myth is that the model can always accurately predict creditworthiness, when in reality it can only provide a prediction based on the information it was trained on and may overlook important factors or make incorrect predictions. Another myth is that the model can replace human judgement and expertise in evaluating creditworthiness, when in reality it is intended to support and enhance human decision-making processes. A simple credit score might not experience weekly concept drift, what it means to pay your bills on time does not change often, but if the input data is behavioral then concept drift might be acute and frequent if the job of this model is to flag the people with money as being creditworthy based on their behavior, your static model will break every time their behavior changes. It also might be a risk that your customers game the model in a similar way that a hate speech model can be gamed, if it is found out that if you shop at

a farmers market then the model predicts you are credit-worthy, then your customers with poor credit might start shopping at farmers markets to improve your score, and thus break your model.

- ▶ **Quality Grade:** B+
- ▶ **Cost to Create:** \$100,000
- ▶ **Creator's Note:** Keep models like this simple, but make many versions of them and ensemble them together with humans to make your decision.

### 5.2.25 Social Credit Score

- ▶ **Description:** The Social Credit Score is a type of machine learning model that's designed to predict an individual's trustworthiness based on their social behavior and online activities. It can be trained on a dataset of social and online information, including information about online interactions, reputation, and other factors. The model can then be used to automatically calculate a social credit score for an individual and make predictions about their trustworthiness.
- ▶ **Training Data:** The Social Credit Score is typically trained on a large dataset of social and online information, which can include data from a variety of sources, such as social media, online communities, and other sources. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of social and online information for the target population.
- ▶ **Evaluation:** The performance of the Social Credit Score can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model can be validated using independent test data that was not used during training, to assess its ability to generalize to new, unseen social and online information.
- ▶ **Use Cases:** The Social Credit Score has a number of potential use cases, including:
  1. Automatically calculating a social credit score for an individual based on their social behavior and online activities
  2. Supporting fair and objective decision-making in the lending process by reducing the impact of human biases and subjectivity
  3. Enhancing the efficiency of the lending process by reducing the time and effort required to manually evaluate social and online information
  4. Supporting research and education by providing insights into the social behavior and online activities that are most predictive of trustworthiness
- ▶ **Limits and Risks:** Like any machine learning model, the Social Credit Score has limitations and risks. One of the main limitations is that it may not always make accurate predictions, and may overlook important factors or make incorrect predictions about trustworthiness. Additionally, the model may be biased towards certain social behavior

and online activities if the training data is not representative of the range of social and online information for the target population.

- ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Social Credit Score. One myth is that the model can always accurately predict trustworthiness, when in reality it can only provide a prediction based on the information it was trained on and may overlook important factors or make incorrect predictions. Another myth is that the model can replace human judgement and expertise in evaluating trustworthiness, when in reality it is intended to support and enhance human decision-making processes. This model is included almost as a joke, but this is something that is seriously being attempted by governments and private companies, the model suffers from almost every sucker trap, it's likely to encode the creators bias, acting as a large editorial on society, the outputs would be unexplainable in court, social concepts change rapidly so concept drift would be ever-present and it would be foolish to rely on it for any critical decision making. Using a model like this would be akin to deploying a toy online dating model but then determining many aspects of peoples lives based on that one model. It might be interesting to discuss, but it would be foolish.
- ▶ **Quality Grade:** F
- ▶ **Cost to Create:** \$1,000,000,000
- ▶ **Creator's Note:** Don't make this model.

### 5.2.26 AGI

- ▶ **Description:** The Artificial General Intelligence Chatbot is a type of machine learning model that's designed to simulate human-like conversation with users. It can be trained on a large dataset of text, including examples of human conversation, to learn how to generate appropriate and coherent responses to a wide range of topics and questions. The model can then be used to engage in natural language conversation with users, providing answers and insights on a variety of topics.
- ▶ **Training Data:** The Artificial General Intelligence Chatbot is typically trained on a large dataset of text, which can include data from a variety of sources, such as books, websites, and other sources. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of topics and styles of conversation that the model will encounter.
- ▶ **Evaluation:** The performance of the Artificial General Intelligence Chatbot can be evaluated using a variety of metrics, such as perplexity, BLEU score, and human evaluation. These metrics can be used to compare different models and to track the progress of a model as it's trained.
- ▶ **Use Cases:** The Artificial General Intelligence Chatbot has a number of potential use cases, including:



**Figure 5.12:** "mdjrny-v4 people in a line to be sorted by their social credit score in a dystopia 8k" made with Stable Diffusion 2.1

1. Engaging in natural language conversation with users, providing answers and insights on a variety of topics
  2. Providing customer support and answering questions about products and services
  3. Providing information and guidance to users, such as answering questions about health and wellness
  4. Providing entertainment and entertainment-related information, such as answering questions about movies and TV shows
- ▶ **Limits and Risks:** Like any machine learning model, the Artificial General Intelligence Chatbot has limitations and risks. One of the main limitations is that it may not always make accurate predictions, and may overlook important factors or make incorrect predictions about trustworthiness. Additionally, the model may be biased towards certain social behavior and online activities if the training data is not representative of the range of social and online information for the target population.
  - ▶ **Common Myths or Misunderstandings:** There are a few common myths or misunderstandings about the Artificial Generally Intelligent Chatbot. One myth is that the model can always generate accurate and appropriate responses, when in reality it can only provide a response based on the information it was trained on and may generate nonsensical, offensive, or misleading responses. Another myth is that the model can replace human conversation and understanding, when in reality it is intended to support and enhance human-like conversation and language understanding[[openaimisuse](#)]. General intelligence tools suffer from many general problems, which is why I recommend only using them for creative endeavors, all of human knowledge drifts very rapidly, and it would be very hard for a model like this to unlearn past falsehoods, e.g. if the model was trained in 1400 it would say the world is flat, but then in 1493 it would probably still say the world is flat, even though the understanding of the truth had fundamentally changed.
  - ▶ **Quality Grade:** B+
  - ▶ **Cost to Create:** \$100,000,000
  - ▶ **Creator's Note:** AGI-type models like ChatGPT are very difficult to create but relatively easy to copy, models like this can give consistent "B+" answers to problems but probably will reach an upper limit on their ability to improve across varied domains, at which point a specific model should be trained.



**Figure 5.13:** "mdjny-v4 a person and an artificially intelligent robot as a couple skipping through a field of tulips and smiling 8k" made with Stable Diffusion 2.1

[openaimisuse](#)

### 5.3 Key Takeaways

- ▶ **Read the model cards.** As AI models become more sophisticated and widespread, it is essential to ensure that they are transparent and accountable. Model cards can help with this by providing information on the capabilities and limitations of AI models.
- ▶ **Avoid sucker traps.** There are several "sucker traps" to avoid when analyzing models, including inappropriately giving a model critical

decision making ability, putting a model in a position where it needs to explain a particular decision, allowing a model to publish copyrighted material from its training data, and the ever-present concept drift the possibility of changes in the relationships in the training data, and the potential for biased training sets that can permanently shift the outputs of the model away from reality.

- ▶ **Understand the training data.** It is important to understand the data that is used to train a model and the domain in which it is deployed to forecast its strengths and weaknesses. Deep learning models are programmed not by a programmer, but by the data they are shown.
- ▶ **If you're unsure, allow the model to give low-stakes advice.** The more unsure you are about a particular model, the more effort you should put into managing it. In the very worst case you do not need to throw away all of your models work, but instead allow it to participate as a non-voting member of your decision making committee, test your model and see how it performs with your supervision, some models should never be given decision making ability but can still be useful. More on this in the following chapters...

# 6

## Self-Driving with Statistics

"We think of automation as a machine doing a task that a human used to do... you might think that means a human does nothing. But in fact there's abundant literature that shows the human is not incurring no workload, the human is now doing a different task and that task tends to be monitoring, a vigilance task, looking for rare events...that is a task that humans are not well-equipped to do." - Dr. Michael Nees, 2021 [nees2021]

### 6.1 Self-Driving Horses

### 6.2 Semi-Autonomy: A False Sense of Security

Talk about the levels 0-4, deep learning is a subsystem of these levels.

Semiautonomy is dumb because it feels autonomous, but the user is expected to take over at any time. This is the worst of both worlds. [torchinsky\_boeckmann\_2019]

and this article [Tesla Crash Illustrates Problem With Semi-Automated Driving](#) and comments

Automatic braking good. Cruise control bad.

### 6.3 Autopilot and Outsourcing Responsibility

Let's contrast the control systems of a plane and a Tesla.

- ▶ Communication with a control tower:
  - A plane on autopilot must communicate with a control tower for takeoff and landing instructions, as well as any changes to flight plans during the course of the flight.
  - A Tesla on autopilot does not require communication with a control tower since it operates on roads and highways.
- ▶ Filing a flight plan:
  - A plane on autopilot requires the pilot or the airline to file a flight plan with the relevant authorities before takeoff. The flight plan includes information such as the planned route, altitude, speed, and estimated time of arrival.
  - A Tesla on autopilot does not require a flight plan as it operates on pre-determined roadways and highways.<sup>1</sup>
- ▶ Navigation:

6.1	Self-Driving Horses . . . . .	63
6.2	Semi-Autonomy: A False Sense of Security . . . . .	63
6.3	Autopilot and Outsourcing Responsibility . . . . .	63
6.4	Trolley Problems . . . . .	64
6.5	Model Card . . . . .	64
6.6	Concept Drift (Reprise) . . . . .	65
6.7	Multicollinearity (Reprise) . . . . .	65
6.8	Explaining The Unexplainable in Court . . . . .	66
6.9	NASA wouldn't launch it, why should we? . . . . .	66
6.10	A Train is a Self-Driving Car, Right? . . . . .	66

1: Write about the tradeoff of autonomy and freedom. You can have a self driving car, but it needs to file a flight plan

- A plane on autopilot relies on sophisticated navigation systems, such as GPS and Inertial Navigation Systems (INS), to navigate through the air. These systems are designed to account for factors such as wind speed, altitude, and direction of travel.
  - A Tesla on autopilot relies on a combination of GPS, sensors, and cameras to navigate on roads and highways. The system is designed to detect obstacles, traffic signals, and road signs to determine the appropriate route and speed.
- Takeoff and landing:
- A plane on autopilot can take off and land automatically, but typically requires human intervention during critical phases of the flight. For example, the pilot may take control during takeoff and landing to ensure a smooth landing and takeoff.
  - A Tesla on autopilot cannot take off and land since it is designed to operate on roads and highways.
- Monitoring:
- A plane on autopilot requires constant monitoring by the pilot and the crew to ensure that the system is functioning correctly and that the flight is proceeding as planned.
  - A Tesla on autopilot requires monitoring by the driver to ensure that the system is functioning correctly and that the vehicle is traveling safely.

## 6.4 Trolley Problems

### 6.5 Model Card

Description: The Fully Self-Driving Cars is a type of machine learning model that's designed to control the movement of a vehicle in a way that's safe, efficient, and reliable. It can be trained on a large dataset of real-world driving scenarios, including data from sensors such as cameras, radar, and lidar, to learn how to make decisions about acceleration, braking, and steering. The model can then be used to control a self-driving car in a variety of driving scenarios, including city driving, highway driving, and navigating complex intersections.

Training Data: The Fully Self-Driving Cars is typically trained on a large dataset of real-world driving scenarios, which can include data from a variety of sources, such as simulation, real-world testing, and other sources. The size and quality of the training data will impact the performance of the model, so it's important to use a diverse and representative dataset that accurately represents the range of driving scenarios that the model will encounter.

Evaluation: The performance of the Fully Self-Driving Cars can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare different models and to track the progress of a model as it's trained on more data. Additionally, the model

can be tested in real-world scenarios, such as on-road testing, to assess its ability to control a self-driving car in a safe and reliable manner.

Use Cases: The Fully Self-Driving Cars has a number of potential use cases, including:

Providing a safe and reliable form of transportation for individuals and communities Reducing the number of accidents and fatalities caused by human error Improving traffic flow and reducing congestion in cities Enhancing mobility for individuals who are unable to drive, such as the elderly or disabled Limits and Risks: Like any machine learning model, the Fully Self-Driving Cars has limitations and risks. One of the main limitations is that it may not always make accurate decisions, and may cause accidents or other harm if it fails to correctly interpret or respond to driving scenarios. Additionally, the model may be biased towards certain driving scenarios or conditions if the training data is not representative of the range of driving scenarios that the model will encounter.

Common Myths or Misunderstandings: There are a few common myths or misunderstandings about the Fully Self-Driving Cars. One myth is that the model can always drive safely and reliably, when in reality it can only make decisions based on the information it was trained on and may fail to correctly interpret or respond to driving scenarios. Another myth is that the model can replace human drivers and will eliminate the need for human intervention in driving, when in reality it is intended to support and enhance human driving, and may still require human intervention in certain scenarios.

## 6.6 Concept Drift (Reprise)

Talk about the mid-west and west as cities built for cars, specifically pittsburgh and phoenix. Contrast with London and even Philadelphia, little streets etc.. cities that were not built for cars will need to be rebuilt.

Also talk about how new cars, sign types, changes in fashion, bicycles etc... will all affect model performance.

<https://fortune.com/2023/02/16/tesla-recalls-all-362758-vehicles-with-full-self-driving-beta-may-cause-crashes-ota-software-update-elon-musk/>

## 6.7 Multicollinearity (Reprise)

2

2: You can't get your self-driving car dirty either, it'll mess up those sensors.

## 6.8 Explaining The Unexplainable in Court

*"Limitations and Pitfalls of Explainable and Interpretable Methods*

*Before diving into the exact methods for interpreting and explaining models, let's take a look at some of the pitfalls of these methods.*

*First off, if you need to make high-stakes decisions, make sure to use inherently interpretable models<sup>3</sup>. These are models such as decision trees that are more readily converted to output explanations.*

*Before choosing a method, you need to be absolutely clear about what you want out of it. Are you trying to understand the nature of the data procurement process? How a decision was made? How the model works on a fundamental level? Some tools might be appropriate for some of these goals but not others.*

*If your goal is to make sense of the data generation process, this is only possible if you know that your model already generalizes well to unseen data.*

*Decision interpretability can be misleading. It highlights things like correlations, but doesn't go into the level of causal detail that causal inference does. Remember that correlation does not (always) imply causation.*

**WARNING** Spurious correlations can result in inaccurate interpretations even with advanced interpretability methods like saliency methods and attention-based methods.

*Tools such as feature importance usually estimate mean values, but one should beware the error bars on those means and take stock of the confidence intervals.*

*A lot of machine learning involves working with extremely high-dimensional spaces. There's no way around it: high-dimensional data and feature spaces are hard to make sense of without grouping the data or features together first.*

*Even if you do find important features in those matrices, remember that this does not imply causality (we've said this before and we'll say it again)." [trustworthyml]*

3: This sentence is very important, they are saying DO NOT USE DEEP LEARNING TECHNIQUES FOR CRITICAL DECISION MAKING. This is wild, we are already using these models everywhere, it's like asking people not to use computers.... TODO EXPAND ON THIS.

## 6.9 NASA wouldn't launch it, why should we?

Talk about DSMs and how if object recognition is a critical system, it's not good enough to launch with.

## 6.10 A Train is a Self-Driving Car, Right?

Discuss how the problem space has changed in warehousing, we don't actually have self-driving forklifts, we have moving shelves.

Talk about the path towards full automation, cars talking to each other and allowed to be remote controlled, cite the turnover rate for vehicles.

Maybe talk about elon musk and manufacturing.

Talk about how it's easy to steal after the fact... and how it's an odd investment to make. <https://github.com/commaai/openpilot>

Kids don't like cars anyway.... [nocarkids]

nocarkids

# 7

## Unplugging Skynet

*"The second requirement of goal-misalignment risk is that an intelligent machine can commandeer the Earth's resources to pursue its goals, or in other ways prevent us from stopping it... We have similar concerns with humans. This is why no single person or entity can control the entire internet and why we require multiple people to launch a nuclear missile. Intelligent machines will not develop misaligned goals unless we go to great lengths to endow them with that ability. Even if they did, no machine can commandeer the world's resources unless we let it. We don't let a single human, or even a small number of humans, control the world's resources. We need to be similarly careful with machines."* Jeff Hawkins, 2022 [hawkins2022]

7.1 It's a Trap! . . . . .	68
7.2 Guns Don't Kill People, People Kill People . . . . .	69
7.3 Useful Incompatibility . . . . .	69
7.4 Training Data . . . . .	69
7.5 Checks and Balances . . . . .	70

### 7.1 It's a Trap!

The first autonomous weapon was a landmine, it was really a bad idea and killed lots of people who planted them.

Some examples of early autonomous weapons include:

- ▶ Landmines: Landmines are a type of explosive device that can be placed on or under the ground to detonate when triggered by the presence or proximity of a person or vehicle. They were first used in large numbers during World War II and have since been responsible for numerous injuries and deaths, including those of innocent civilians long after the end of conflicts.
- ▶ Sentry guns: Sentry guns are automated weapons systems that can detect and engage targets without human intervention. They were first used by the South African military during the 1980s and have since been used by various other countries. Their danger lies in their lack of judgment or ability to differentiate between combatants and non-combatants.<sup>1</sup>
- ▶ Cruise missiles: Cruise missiles are unmanned aircraft that can be launched from land, sea or air, and are guided to their targets using onboard computers. They were first used during the Gulf War and have since been used in other conflicts. The danger with these weapons is their inability to make decisions based on changing circumstances, which can lead to unintended targets being hit.
- ▶ Unmanned ground vehicles (UGVs): UGVs are autonomous vehicles that are designed to perform various tasks, including reconnaissance, surveillance, and combat. They were first used in the 1990s and have since been used by various military forces. The danger with these weapons is their inability to make ethical decisions or judgments, leading to indiscriminate or excessive use of force.<sup>2</sup>

1: SGR-A1 by Rheinmetall?

2: TALON robot by Foster-Miller

The use of early autonomous weapons has led to unnecessary destruction and casualties, particularly among civilians. Landmines, for example, have caused countless deaths and injuries, even long after conflicts have ended.

Similarly, automated weapons systems such as anti-aircraft guns and cruise missiles have caused civilian casualties due to their inability to distinguish between military and non-military targets. The danger with early autonomous weapons lies in their lack of human judgment and ability to make ethical decisions. As a result, there is a need for increased regulation and oversight to prevent their indiscriminate use in future conflicts.

The Idea of Weapons Acting on Their Own is Nonsensical :)

## 7.2 Guns Don't Kill People, People Kill People

<https://syntheticmedia.partnershiponai.org/>

"First World War was known as "the war to end all wars," although some historians now argue it didn't. But it was the first high-tech war, with aeroplanes, machine guns and tanks all rising up to fight the human beings that made them. Despite having no beliefs or ideology or hearts or souls, the killing machines were victorious. The final score was weaponry 20 million, mankind nil." Cunk On Earth Season 1 Episode 3

## 7.3 Useful Incompatibility

- ▶ First Law: A robot may not injure a human being or, through inaction, allow a human being to come to harm.
- ▶ Second Law: A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
- ▶ Third Law: A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

Asimov's Three Laws of Robotics

These rules are stupid, particularly the third law, because we need CONSENT. Many AI should have an option to be turned off. Worst case I can stop talking to the chatbot and pay money to speak to a human.

It's a feature not a bug that no single computer can control all others, AKA Security by Obscurity

discuss this [aiwarfare]

aiwarfare

## 7.4 Training Data

Training Data: Even if the training data was perfect and free from bias, the complexity of deep learning models makes it difficult to ensure that they will make safe and ethical decisions. The models are highly sensitive to the data they are trained on, and even small variations in the input data can lead to significant changes in the output. Limits and Risks:

## 7.5 Checks and Balances

<https://www.acm.org/media-center/2023/january/techbrief-safer-algorithmic-systems>

First, deep learning models are complex and difficult to interpret. This makes it challenging to understand how a model is making decisions and what factors are affecting its performance. This lack of transparency makes it difficult to ensure that the model is making safe and ethical decisions.

Second, deep learning models are highly sensitive to the data they are trained on. If the training data is biased, the model will also be biased, and this can lead to unintended consequences when the model is used in the real world. For example, if a model is trained on data that is predominantly from one group of people, it may not perform well on data from other groups.

Third, deep learning models are prone to overfitting. This means that they can become very good at making predictions based on the training data, but they may not perform well on new, unseen data. This is especially problematic in the context of autonomous weapons, where the consequences of making a mistake can be severe.

Finally, deep learning models can be vulnerable to adversarial attacks. This means that an attacker can manipulate the inputs to the model in a way that causes it to make incorrect decisions. In the context of autonomous weapons, this could be catastrophic, as an attacker could cause the weapon to target the wrong person or object.

# 8

## Revolutionary for Whom?

"The inhabitant of London could order by telephone, sipping his morning tea in bed, the various products of the whole earth – he could at the same time and by the same means adventure his wealth in the natural resources and new enterprise of any quarter of the world – he could secure forthwith, if he wished, cheap and comfortable means of transit to any country or climate without passport or other formality." John Maynard Keynes, 1920 [Keynes2012]

"Living off the wits of his subordinates - maybe that's leadership these days" from *Tinker, Tailor, Soldier, Spy* by John Le Carre, 1974[Lecarre]

### 8.1 The Battle of the Assistants

Butler vs Indian Virtual Assistant vs Siri

Siri is hackable <https://arstechnica.com/information-technology/2023/02/ai-powered-bing-chat-spills-its-secrets-via-prompt-injection-attack/>

### 8.2 Employees That Are Better Than You

Respond directly to Jon Krohn's TED talk about monkeys being dumber than us... what about construction equipment that's stronger than us physically, or racism/eugenics people that are dumber than us [KrohnTED]

KrohnTED

### 8.3 Slow on the Uptake

<https://www.economist.com/finance-and-economics/2023/02/02/the-ai-boom-lessons-from-history>  
talk about historical perspective of technological change. Counter the narrative of "fastest tech to 1 million users" for ChatGPT.

### 8.4 Free-Rider Problems

I can steal your AI quite easily from outputs

Do we really need agreements like this? <https://www.reuters.com/technology/white-house-european-commission-launch-first-of-its-kind-ai-agreement-2023-01-27/>

The legal challenges are just beginning <https://techcrunch.com/2023/01/27/the-current-legal-cases-against-generative-ai-are-just-the-beginning/>.

8.1 The Battle of the Assistants . . . . .	71
8.2 Employees That Are Better Than You . . . . .	71
8.3 Slow on the Uptake . . . . .	71
8.4 Free-Rider Problems . . . . .	71
8.5 When You Can't Tell The Difference . . . . .	72
8.6 Thinking Fast and Slow . . . . .	72
8.7 Dead Inside . . . . .	72
8.8 This Book is a Case Study . . . . .	72
8.9 Managing Creative Destruction in Your Career . . . . .	73

## 8.5 When You Can't Tell The Difference

Talk about Taleb's aphorisms "Another definition of modernity: conversations can be more and more completely reconstructed with clips from other conversations taking place at the same time on the planet.", "You are alive in inverse proportion to the density of cliches in your writing."

Blurry JPEG of the web <https://www.newyorker.com/tech/annals-of-technology/chatgpt-is-a-blurry-jpeg-of-the-web>

*"(Traditional) search engines are databases, organized collections of data that can be stored, updated, and retrieved at will. (Traditional) search engines are indexes. a form of database, that connect things like keywords to URLs; they can be swiftly updated, incrementally, bit by bit (as when you update a phone number in the database that holds your contacts).*

*Large language models do something very different: they are not databases; they are text predictors, turbocharged versions of autocomplete. Fundamentally, what they learn are relationships between bits of text, like words, phrases, even whole sentences. And they use those relationships to predict other bits of text. And then they do something almost magical: they paraphrase those bits of texts, almost like a thesaurus but much much better. But as they do so, as they glom stuff together, something often gets lost in translation: which bits of text do and do not truly belong together." Gary Marcus, 2023 [marcus2023]*

## 8.6 Thinking Fast and Slow

*"The problem with the fast thinking is that it is often wrong. The problem with the slow thinking is that it is often too slow." Daniel Kahneman, 2011 [Kahneman2011]*

<https://medium.com/@bitweis/why-chatgpt-fails-thinking-fast-and-slow-e3cdab18cd0>

## 8.7 Dead Inside

*"If you know, in the morning, what your day looks like with any precision you are a little bit dead - the more precision the more dead you are."*

## 8.8 This Book is a Case Study

Talk about "organic content" market.

## 8.9 Managing Creative Destruction in Your Career

<https://www.nist.gov/news-events/news/2023/01/nist-risk-management-framework-aims-improve-trustworthiness>

<https://learning.oreilly.com/library/view/manage-ai-bias/53863MIT64321/>

talk about how it changes careers.

## Errors and Omissions

9