

2016 Thought Leader Forum

Proceedings

May 19, 2016

Tarrytown, NY

Table of Contents

Michael Mauboussin, Credit Suisse
Introduction and Welcome 3

Bill Gurley, General Partner, Benchmark Capital
How to Miss by a Mile.....9

Pedro Domingos, Professor of Computer Science, University of Washington
The Master Algorithm 26

Cade Massey, Professor of Operations, Information and Decisions, University of Pennsylvania
Algorithm Aversion.....48

Michael Mauboussin

Introduction

Information and circumstances change constantly in the worlds of investing and business. As a consequence, we have to constantly think about what we believe, how well those beliefs reflect the world, and what tools we can use to sharpen our decisions. Because we operate in a world where we can succeed only with a certain probability, we have to learn from our mistakes. Hence, the theme for the Thought Leader Forum in 2016 was “What Being Wrong Can Teach You About Being Right.”

This year’s forum featured a venture capitalist, a computer scientist, an economist who focuses on decisions, and a leading sports executive. Each explored an area of how our thinking and decisions can come up short of the ideal. We heard about how assumptions deeply shape how you assess a company’s potential and how well-intentioned incentive systems can go awry. There was an exploration of how computers, through machine learning, can serve as a new source of knowledge, complementing evolution, experience, and culture. Notwithstanding the potential benefits of augmenting our intelligence through computers, we discussed why we humans have an aversion to algorithms and how to overcome it. And then there is the issue of the old and new guard: how can we convince some who have been successful in an old regime to accept new and better ways of doing things?

The theme of “what being wrong can teach you about being right” has lessons to teach us about naïve realism, man versus machine, and the role of change. Naïve realism is the sense that our view of the world is the correct one. But when confronted with reality, we need to revisit our beliefs.

For example, when we face someone who has beliefs different than ours, we tend to adopt one of three attitudes so that we can perpetuate our position. First, we might assume the other person is merely unequipped with the facts, so simple sharing will swing them to our side. Next, we believe that even with the facts, the other person lacks the mental capacity to see the consequences as we do. We can write off those people. Finally, there may be people who understand the facts as we do but turn their backs on what we perceive to be the truth. We categorize those people as evil.

Machine learning and artificial intelligence are again hot terms. Google DeepMind’s AlphaGo program, which beat a human champion in the board game of Go much sooner than most experts had predicted, is emblematic. The question is how we divide the cognitive work between machines and human judgment. If you are in the information business—and the chances are good this is true if you are reading this—then you must consider carefully how you might integrate computers and humans.

All of this implies change, something we are loathe to do. Changing your mind takes time, effort, and humility. This is especially pertinent when you have been successful in your domain. Strategy in sports is a good analogy. There are traditional ways to do things, and often those ways are effective. But more careful analysis has revealed strategies that fly in the face of conventional wisdom that are clearly better. Defensive shifts in baseball are but one example. Convincing the old guard to change—and eventually, we are all part of the old guard—is a difficult hurdle.

The following transcripts not only document the proceedings, they also provide insights into how you can improve your own ability to learn from mistakes and improve your odds of being right in the future. Bill Gurley suggested that the high valuations for some technology startups (so-called “unicorns”) and the low level of liquidity is a balance that is not tenable. Pedro Domingos explained how computers might be able to complete tasks that are out of the grasp of humans. Cade Massey showed that we don’t readily embrace algorithms but that there is a way to overcome this aversion and improve decisions. And Paul DePodesta suggested that the bias against change has less to do with the game you are playing and more to do with how we humans think.

WHAT BEING WRONG CAN TEACH US ABOUT BEING RIGHT



VISUALIZED by INKFACTORYSTUDIO.COM



Michael Mauboussin

Credit Suisse

Michael Mauboussin is a Managing Director of Credit Suisse in the Global Markets division, based in New York. He is the Head of Global Financial Strategies, providing thought leadership and strategy guidance to external clients and internally to Credit Suisse professionals based on his expertise, research, and writing in the areas of valuation and portfolio positioning, capital markets theory, competitive strategy analysis, and decision making.

Prior to rejoining Credit Suisse in 2013, he was Chief Investment Strategist at Legg Mason Capital Management. Michael originally joined Credit Suisse in 1992 as a packaged food industry analyst and was named Chief U.S. Investment Strategist in 1999. He is a former president of the Consumer Analyst Group of New York and was repeatedly named to *Institutional Investor's* All-America Research Team and *The Wall Street Journal* All-Star survey in the food industry group.

Michael is the author of *The Success Equation: Untangling Skill and Luck in Business, Sports, and Investing*, *Think Twice: Harnessing the Power of Counterintuition*, and *More Than You Know: Finding Financial Wisdom in Unconventional Places*. He is also co-author, with Alfred Rappaport, of *Expectations Investing: Reading Stock Prices for Better Returns*.

Michael has been an adjunct professor of finance at Columbia Business School since 1993 and is on the faculty of the Heilbrunn Center for Graham and Dodd Investing. He is also chairman of the board of trustees of the Santa Fe Institute, a leading center for multi-disciplinary research in complex systems theory. Michael earned an AB from Georgetown University.

Good morning. For those of you whom I haven't met, my name is Michael Mauboussin, and I am head of Global Financial Strategies at Credit Suisse. On behalf of all of my colleagues at Credit Suisse, I want to wish you a warm welcome to the 2016 Thought Leader Forum. For those who joined us last night, I hope you had a wonderful evening. We are very excited about our lineup for today.

I'd like to do a couple of things this morning before I hand it off to our speakers. First I want to highlight the levels at which you might consider today's discussion about the idea of how being wrong can inform you about being right. I then want to discuss the forum itself, including what you can do to contribute to its success.

You might listen to today's discussion at three different levels. Some of the points will span multiple levels, but these are some of the ideas that we'll hear about throughout the day.

The first relates to the ideas of naïve realism. In psychology, this is the human tendency to believe that we see the world around us objectively and that people who disagree with us must be uninformed, irrational, or biased.

The second is man versus machine. This is a theme that is popping up everywhere. What are algorithms good at and what are humans good at? How do we use algorithms to augment our performance? Why do we struggle to defer to algorithms in many settings?

The final is the issue of change. Organizational inertia is a huge issue in many firms. How can firms keep up? How do we integrate new information? What is the psychology of change?

Let's start with naïve realism. Here's a cartoon I love: as you can see, there are two armies preparing to square off, and the quote is: "There can be no peace until they renounce their Rabbit God and accept our Duck God." The picture shows that the flags of the competing armies are the exact same. This is based on the rabbit-duck illusion, an ambiguous picture that can be interpreted either as a rabbit or a duck.

The idea of naïve realism in psychology is that we all think that we have an objective reality of the world. As a consequence, we have a hard time accepting that others have different points of view. So we all walk around with beliefs that we think are true. Otherwise we wouldn't hold onto those beliefs. Things become interesting when those beliefs confront the world.

Here's a well-known experiment that demonstrates this point. A psychologist named Elizabeth Newton set up an experiment whereby there were "tappers" and "listeners." The tappers were given a list of 25 well-known songs, such as "Happy Birthday to You," and were asked to tap the rhythm of the song on the table. The task of the listener was to identify the song based on the taps.

She ran 125 trials of this. The listeners were able to identify only 3 of the songs, a success rate of about 2.5 percent. But when the researchers asked the tappers what percent they thought the listeners would be able to identify correctly, the answer was 50 percent! This is related to the curse of knowledge, which is also a huge impediment to communication. Again, we struggle to understand that others don't see the world as we do.

So if you see the world one way and others see it a different way, you have to reconcile the views. And as we do so, we tend to assume one of three things. The first is that the other person simply doesn't know the facts that you do, and hence is ignorant. The answer is simply to inform them so that they will then see your point of view. The second is that the person has the facts, but they are just too stupid to understand them properly. The last assumption is that people know the facts and can comprehend them, but they just turn their backs on the truth. Unbelievers in religion are an example.

Now consider how you assess people who don't agree with you. Do you evoke one of these assumptions to reconcile their beliefs with yours?

We now turn to a theme that will spread through the day. I am calling it man versus machine but it may be just as accurate to say humans versus algorithms. The first point I want to make refers to what I call "the expert squeeze."

The way to think of it is as a continuum. On one side there are problems that are rules-based and consistent. Here, experts are often proficient but computers are quicker, cheaper, and more reliable. Today, of course, you have to point to the success of AlphaGo—Google DeepMind's program that beat a champion in Go.

At the other side of the continuum are problems that are probabilistic and in domains that change constantly. Here, the evidence shows that collectives do better than experts under certain conditions. Making sure those conditions are in place is crucial for a decision maker.

I'm now going to steal a bit of thunder from our second speaker and introduce various approaches to machine learning. But my point of emphasis is somewhat different. If your organization relies on fundamental research, do any of these approaches seem familiar?

For example, lots of investors like to appeal to analogies: this investment is like that investment from the past. The interesting question then becomes: what can we, as fundamental analysts, learn from what's going on in machine learning? The next step is considering how we can integrate machine learning techniques into a decision-making process. If you are relying on quantitative methods, how do you think about the biases built into the algorithms?

The final issue I'll mention for man versus machine is that we as humans tend to be uneasy letting our fate be decided by an algorithm, even if there's abundant evidence that the algorithm is better than a human.

This scene from *Moneyball* captures the tone: the old timers have a difficult time grasping the signal from the statistical analysis. This is true for a few reasons. They generalize from their own experience. They overemphasize recent performance. And they rely on what they see versus cause and effect. We'll talk today about how to overcome algorithm aversion, but it's a huge issue.

The final topic is that of change, which is hard. The first impediment is organizational inertia. Back in the day, I was a food industry analyst, and I recall a story that captured this well.

When David Johnson took over as CEO of Campbell Soup about 25 years ago, the performance of the company lagged its peers. So he did a full review to understand how to improve operations.

He noticed that the firm did a huge annual promotion of tomato soup in the fall every year. Tomato soup was one of their largest and most profitable products. When he asked the executive why they did it, the executive responded, "I don't know, we've always done it."

In World War I, Campbell's strategy was to grow its own tomatoes, harvest them, and then convert them to canned soup. With inventory up and the soup season still months ahead, Campbell used a promotion to clear its inventory. But of course the company long ago went to year-round suppliers, eliminating the post-harvest spike in supply. This evokes a quote from Peter Drucker: "If we did not do this already, would we go into it" now, knowing what we now know?

Perhaps the most challenging thing to do is to update your beliefs when you receive new information.

Here's a famous example from *Thinking, Fast and Slow* by Daniel Kahneman [page 166].

"A cab was involved in a hit-and-run accident at night. Two cab companies, the Green and the Blue, operate in the city. You are given the following data:

- 85% of the cabs in the City are Green and 15% are Blue.
- A witness identified the cab as Blue. The court tested the reliability of the witness under the circumstances that existed on the night of the accident and concluded that the witness correctly identified each one of the two colors 80% of the time and failed 20% of the time.

What is the probability that the cab involved in the accident was Blue rather than Green?"

The most common response is 80 percent, based on the reliability of the witness. But the correct answer is just a little over 41 percent. In Phil Tetlock's terrific book, *Superforecasting*, he has a great line: "Beliefs are hypotheses to be tested, not treasures to be guarded." This is really easy to say and very difficult to do in practice. Changing our minds takes time, effort, in some cases technical skills, and can be embarrassing. Most of us would prefer to keep believing what we believe.

My final thought is on loss aversion. Everyone in this room deals with decisions that work only with some probability. We suffer losses more than we enjoy comparable gains. So we tend to stick to conventional ways of doing things because if we fail, we have lots of company.

There are lots of instances of this in sports. One example is the decision to go for it on fourth down in football. Most coaches prefer the more conservative route even if it gives them a lower probability of winning, because the potential pain of getting stopped on fourth down is a lot worse than the upside of a fresh set of downs.

Before I speak about the goal of the forum, I want to mention the talented folks from Ink Factory.

Dusty and Ryan will be graphically recording all of our presenters today. This means they will be synthesizing the words of our speakers into images and text to capture the key concepts. Their slogan is "*you talk. we draw. it's awesome.*" And we think you will agree. Please feel free to take pictures of the artwork and to tweet the images. And we encourage you to ask them questions—after they are done drawing of course!

Let me end by highlighting what our goals are for the day. First, we want to provide you access to speakers whom you may not encounter in your day-to-day interactions but who are nonetheless capable of provoking thought and dialogue. Second, we want to encourage a free exchange of ideas. Note that our speaking slots are longer than normal. This is in large part because we want to leave time for back-and-forth.

Second, we purposefully call this a "forum" instead of a "conference" precisely for this reason. We want to encourage an environment of inquiry, challenge, and exchange.

Finally, we want this to be a wonderful experience for you, so please don't hesitate to ask anyone on the Credit Suisse team for anything. We will do our best to accommodate you.



Bill Gurley
Benchmark Capital

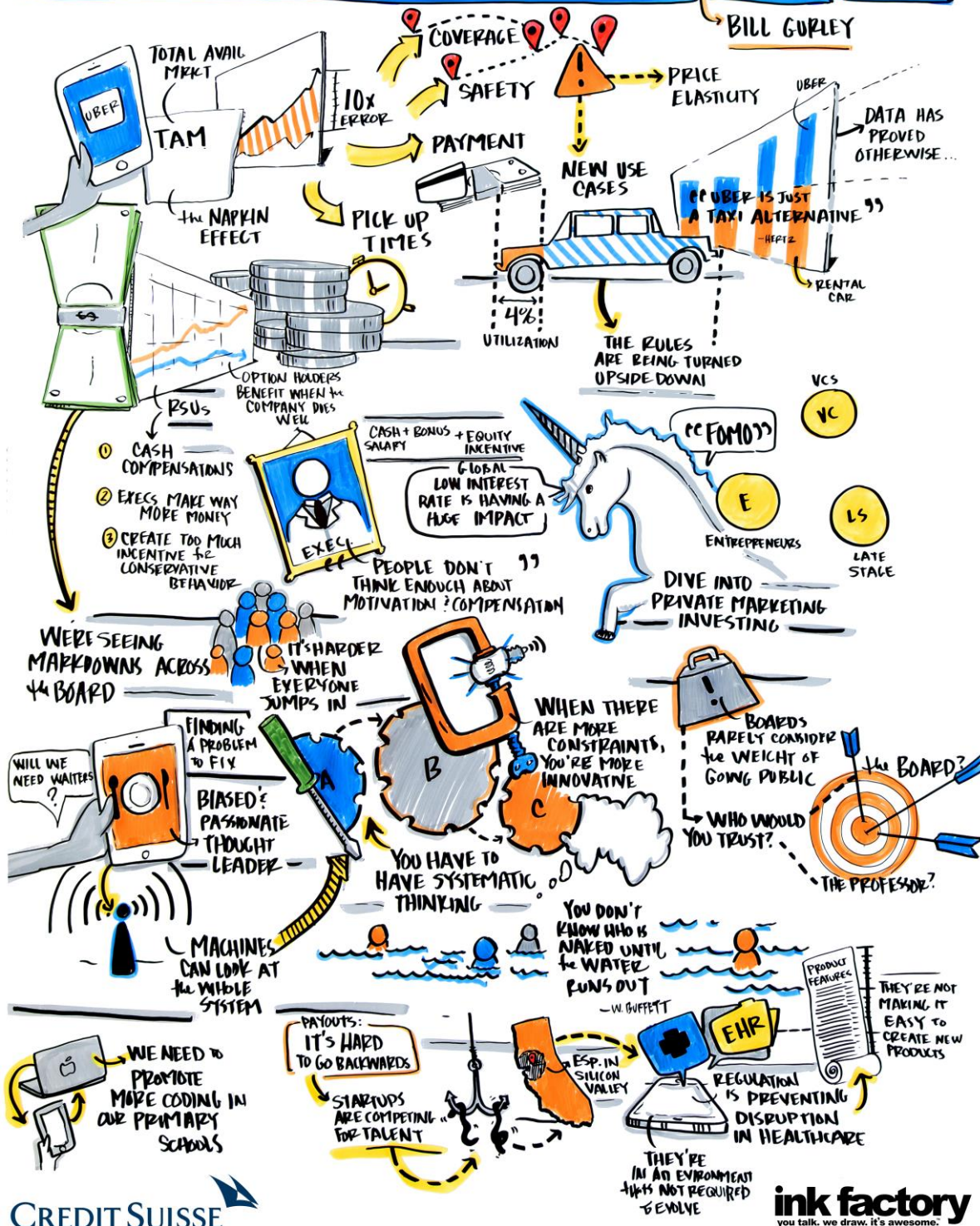
Bill Gurley has spent over 10 years as a General Partner at Benchmark Capital. Prior to Benchmark, Bill was a partner with Hummer Winblad Venture Partners.

Before entering the venture capital business, Bill spent four years on Wall Street as a top-ranked research analyst, including three years at CS First Boston focusing on personal computer hardware and software. His research coverage included such companies as Dell, Compaq, and Microsoft, and he was the lead analyst on the Amazon.com IPO. In both 1995 and 1996, Bill was a member of *Institutional Investor's* All-America Research Team.

Prior to his investment career, Bill was a design engineer at Compaq Computer, where he worked on products such as the 486/50 and Compaq's first multi-processor server. For the past fifteen years, Bill has authored the "Above the Crowd" blog, which focuses on the evolution and economics of high technology businesses.

Bill is on the advisory board of the McCombs School of Business at the University of Texas and a board member at KIPP Bay Area Schools. He received his MBA from the University of Texas in 1993 and a BS in computer science from the University of Florida in 1989.

HOW TO MISS BY A MILE



CREDIT SUISSE

ink factory
you talk. we draw. it's awesome.

Michael Mauboussin: I'm very pleased to introduce our first speaker this morning, my good friend and former colleague, Bill Gurley. Bill is a general partner at Benchmark Capital, one of the world's leading venture capital firms. Bill combines a lot of attributes that I admire: he's wicked smart; intellectually curious; a terrific strategic thinker; financially savvy; and he doesn't take himself too seriously. You can find his writings at abovethecrowd.com, and I recommend reading everything he writes.

In fact, one of Bill's essays deeply inspired the theme for today's forum. Here's the story: one of Bill's portfolio companies is Uber. Aswath Damodaran, a professor at NYU (New York University) and valuation expert, suggested that the company's value should not exceed a fairly modest amount because it was a substitute for cabs and limos. Bill's response—and this was all very civil—was called "How to Miss by a Mile." Bill reframed the TAM (Total Addressable Market) for Uber and suggested that Damodaran could be off by a factor of 25 times.

Prior to joining the venture capital world, Bill was a Wall Street analyst. I had the fortune of working with him in his early days out of business school, where he established himself, in very short order, as a go-to analyst.

Please join me in welcoming Bill Gurley.

Bill Gurley: Before I get started, I want to thank Michael not just for having me here but for being there for me 23 years ago when I started my investment career, and for being a thought leader for the past 23 years. I would not be where I am today probably if I hadn't met Michael back then. He was the food analyst and I was the PC (personal computer) analyst, and somehow he found a way to shape everything I did and have a huge impact on my career. So, thank you, Michael.

When I first met Michael, we both fell in love with this book, *Complexity*, by Mitchell Waldrop, about the rise of the Santa Fe Institute, and I know some of the people in the room have spent a lot of time out there. I tell people this book has affected how I think more than any other book that I've ever read because most of the things we deal with as investors are complex systems, and I'm going to come back to this at the end, but that's why I wanted to point it out here.

Now here's the professor Michael mentioned. I didn't know who he was two and a half years ago. It turns out that he's a rather thoughtful valuation professor, and here's the blog post that he published on Uber where I sit on the board. Not only did he publish this long post, but he did a summary for Nate Silver's FiveThirtyEight.

The title of this one was "Uber Isn't Worth \$17 Billion," and he did a lot of work to calculate that he thought the value should be around \$5 billion. He made a number of critical errors in his thinking from my point of view, and Michael asked me to talk about them today. He said, "Why don't you talk about 'How to Miss by a Mile'?" And he wanted me to incorporate three or four different things that I've seen and that we've talked about over the years.

So this is just the first one, but he made a number of errors that I think are interesting and I want to highlight. The first one was he didn't theorize that there might be a network effect. This napkin drawing was actually done by David Sacks, one of the angel investors in Uber. It basically shows that more demand drives more drivers. More drivers creates more geographic coverage. That leads to less downtime which can lead to lower prices because you in effect have more utilization, like an airplane. It also leads to faster pickups. Lower prices and faster pickups drives more demand, and so you have a circle here. He didn't even consider this, but I don't even think that was his biggest mistake.

His biggest mistake related to TAM, total available market. He assumed the market that Uber was attacking was taxis and black cars. Basically he looked at the revenue for taxis and black cars at the time, and he assumed that Uber could get some fraction of that. He wrote this two and a half years ago, and just to show you how far off

he was on that assumption already, the market for taxis and black cars in San Francisco in 2011 was \$120 million, and the market for ridesharing in San Francisco right now is \$1.2 billion, a 10 times differential and totally outside of his scope of analysis. It's still growing, and we've only touched 13 percent of the population, so it's likely going to be off by more than 10 times, just his TAM assumption.

So how does this happen? What did he miss? He didn't think about any of these things. He didn't think about the fact that the pickup times were much quicker than a taxi. If you've ever ordered a taxi in a city like Houston, it takes 30 minutes for it to show up. Greater coverage density. Uber already has more density in areas where taxis have never historically served.

Easier payment, higher civility. It's actually a better experience because of the dual rating and higher trust and safety. People routinely now rate this experience as way better than taxis from a trust standpoint and, in fact, there's numerous blog posts that you can find on the web of people getting their phones back, their keys back, their money back all through the system.

He made an even bigger mistake for a financial professor which is he didn't think about price elasticity. Now this is taught I think in early microeconomics courses. At the time he wrote it, Uber was already about half of taxi prices in many cities, and these prices were published. So he could have found them, but he didn't. He didn't look at it.

He also didn't consider new use cases, expanded geographic coverage, a rental car alternative, couples night out. Uber peaks on Friday and Saturday night and has become a huge DUI deterrent. Where I live near Palo Alto, a couple in Menlo Park will go out to eat in Palo Alto. They used to drive, but they don't anymore. They just take Uber. They don't have to park. They don't have to worry about drinking.

Transporting kids, seniors, supplement of mass transit are all other things that he didn't consider. UberPOOL, where you share rides in a car, now comprises about 20, 25 percent of rides and is competing with bus and subway.

Then, a really big one is as an alternative for car ownership, which changes the perspective entirely. The professor didn't think about car rental alternatives. The CEO of Hertz didn't think about it either because he constantly says that Uber's a "taxi alternative," a phrase he uses. This is Hertz's 12-month stock price.

[Shows video clip of Jim Cramer on CNBC]: Why doesn't Hertz just own up to Uber? I mean again they disappointed. Again they said that things had gotten worse this quarter than last quarter. Again they denied without really saying it's denial that Uber's the problem, but I think we all have Uber in our cell phone.

Uber is one of the prestige brands that I talk about. I talk about cosmetics because when you walk outside you've got to be dressed up these days. Put your stuff on. I talk about the iPhone as being a prestige brand and Uber being a prestige brand.

I think that is directly impacting Hertz. You would think that after all the consolidation in the rental business that rates would go up, but you know what? They can't because they're really competing against Uber.

Bill: I don't know if any of you have recently used rental cars. I used to go to Seattle and L.A. frequently for business, and the difference is astounding because with a rental car you've got to arrive 30 to 45 minutes earlier.

You've got to get on the shuttle. Ride the shuttle out there. You have to have maps for everywhere you're going. You have to know where you're going to park. You have to get there 15 minutes earlier to get to the parking structure to get to your car. And by the way, when you get to the hotel you get charged \$40 to put it there.

Now Uber's just completely better, and I would pay three times as much for the experience I get on Uber as I would for renting a car. If you think about a car ownership alternative . . . oh, wait.

This is expense data that's now coming out from companies that track corporate expenses. So not only can you presume that it might affect rental cars, you can actually see it. That top section is Uber, and this blue section which was at 50 percent but is now at 30 percent is rental cars.

So now there are hard data to support the theory if you will. Now, Michael said it's okay if I show something produced by a different investment bank. I want you to contrast someone who thought the TAM was just taxi and black car with the perspective in this video. This is my last video and then we'll move on.

[Video Clip]

Bill: I thought that was very clever, but just highlighting a difference. Coming at an approach from the top down of being a car alternative gets you a vastly different result than if you think about the TAM the way that the professor did.

Now I think he made a number of errors, and what's interesting is in his piece he spent a lot of time talking about judgment errors that investors make. [Laughs]

These quotes which I won't read to you were all his reasoning as to why the investor group that had valued Uber had made mistakes. So he thinks about errors in investment judgment. He just didn't think about them in himself.

In addition to many biases he may have had, the blog post actually had this disclosure which I think is quite hilarious. He admitted that he had never used the product. [Laughs] In fact, he lives in New York and only takes subways and doesn't own a car. So he didn't have a great position to make a judgment from.

Let me move on to another example that's a little more esoteric but relates to investing and compensation. I wrote the quote from the cartoon here so you can see it larger, and it's talking about the re-pricing of stock options, which was a common thing that happened in 2000, 2001. And stock options became vilified for a number of reasons. Enron and WorldCom were a big part of it. This always pissed me off because this wasn't Silicon Valley.

Now admittedly Silicon Valley did some really silly things in '99, but they didn't do this. Enron and WorldCom were outside of Silicon Valley but, as we move through the re-pricing and these two examples, options became vilified. Here's *The Wall Street Journal*. Here's *Harvard Business Review*. Even ISS (Institutional Shareholders Services), who claims to be the champion of the shareholder, came out and said options are bad.

Here are the reasons. The first one was they encourage too much risk-seeking. And so it was felt that it was just way too potent of a compensation scheme and that it actually causes you to do illegal things. That was one of them.

They were considered highly dilutive. People were upset about three to four percent a year dilution. They were routinely re-priced, which was what the cartoon talked about, and then they weren't properly accounted for.

So these were the four reasons that we decided to get rid of them. In their place we've used a new thing called an RSU, a restricted stock unit, and at least in Silicon Valley the vast majority of these are zero basis stock units.

Typically you look at what you would have given someone in options, run a Black-Scholes comparison, and then give out an RSU. Starting in 2001 this became common even at the big Microsoft, Intel, Cisco, those types of companies.

What I've done here is I've made some assumptions for the Black-Scholes model, but I'm showing you the comparative payoff for an option holder, or an equivalent value amount in the way they actually move to RSUs through this math of an RSU.

So the option holder, if the company doesn't perform, makes no money, it's worth zero obviously. And this is what it's worth if it goes up. The RSU holder has this payout scheme relative to the stock performance.

Now here I've said let's look at what happens when the company just stays even, when the stock price doesn't move. The RSU executive makes the same compensation when the stock price doesn't change as the option executive makes if the stock went up 50 percent. In all these cases where the stock underperforms, they're still getting paid.

So based on this, do you think executives prefer RSUs over options? Yes? No? Absolutely. They love RSUs. They eat them up. So what has happened? Well, I think this was an error in judgment because I don't think everyone thought through the systematic things that would happen afterwards.

First of all, employees don't hold RSUs, and if you talk to any compensation executive, board member, or CFO about the companies you're investing in that use RSUs, they will tell you some number close to 97 percent of RSUs are sold on the vest date. So they're not a form of stock ownership. They're a form of cash compensation. No one's holding them, and that's huge because options were a form of stock ownership and aligned incentives.

Second, they routinely pay out when shareholders do not see a return, and this is something that I don't know if ISS thought about, but I really wish they had. Executives are making way more money in companies that don't perform than they did when they were holding options. Way more money.

They dilute all the time. Options were dilutive, but they were only dilutive in upside scenarios. I think we were more aligned with options than we are with RSUs.

Thirdly, we talked about options maybe creating incentive for too much risk-seeking. I think RSUs create too much incentive for super-conservative executive behavior, and my mind really understood this when I was recruiting an executive into one of our startups, and I was competing with a large public company.

I sat down with the executive and said, "Well, show me what package they're offering you at the other company," and he had built a spreadsheet, and he had put in all these different things and the RSU package they had given him. I said, "Well, what's this?" And he goes, "Oh, I'm just assuming the stock's flat for the next four years."

So he was accepting a job, thinking about the compensation. He had already decided he didn't care if the stock moved or not. For me, that's just not what you want. I mean you can hold debt if you want that kind of return.

Then the accounting became even worse, which is typical I think in these situations. An in-the-money option prior to 1999 would have unquestionably been expensed. You just wouldn't think about it.

Now the majority of stock compensation is through a zero basis, totally in-the-money option, and yet Wall Street and others still want you to look at non-GAAP (Generally Accepted Accounting Principles) operating income, which excludes SBC (stock-based compensation) and is even more like cash now than it ever was with options.

Just to highlight a few examples — this isn't scientifically statistically significant I understand, but I'm going to do it anyway.

This is Cisco since they went to RSUs in 2000, 15 years ago. If someone were to go and calculate the amount

of stock-based earnings that Cisco management has gotten via RSUs over this timeframe, it's going to be, I don't know, 100 times what they would have gotten from options over this time frame. Some really large number.

Here's Intel. Same kind of thing. I've been at public investor forums before, and I've said to people, "I don't think you understand what this compensation scheme looks like. It's not anything like options were."

It got even worse, so I'm going to walk you through how a lot of compensation committees make decisions these days.

First they take an executive. They go run a compensation survey and they say the total target compensation for this executive should be X, and I use the example here of a CFO. They might say this public CFO should expect to earn \$2 million a year. And so they take the salary, maybe \$500,000, and the bonus, \$250,000, and they say the rest should be stock comp. And then they assign that person \$1.25 million worth of RSUs, and they do this every year, and maybe they put it out in front of them.

This has basically led us to where we're building compensation programs dollar-based not percentage-based. So what would happen if share prices fell dramatically?

In 2015 many of the mid-cap Internet stocks' market caps fell dramatically—LinkedIn, Twitter, Yelp—and stock-based compensation as a percentage of market cap shot up through the roof for some of these companies six to nine percent a year. So we were worried about options being diluted at three percent a year.

Now we're at six to nine percent a year, and because it's a zero-based thing, the option had the cash part that was coming back. So it's even worse than this. We wanted to get to a place where there was less dilution, but we've gotten to a place where there's dramatically more dilution and a misaligned incentive.

I'm a venture capitalist by day, but someone asked me to look at a couple stocks they were thinking about buying. I literally went and built a few models, and I figured this out. Then I went and downloaded the models from Wall Street research analysts, and they hadn't figured it out.

They hadn't figured out that their share counts didn't have the incremental dilution in them. I was like, "oh my god." Literally two months later reports start flying out where people have figured this out and are starting to look at how excessive it is.

This is one of my favorite quotes, this [Charlie] Munger quote. He had a list of [25] biases that people bring to the table around investing, and this is my favorite one. He says his number one is that people don't think enough about motivation and compensation. I think the RSU is just a horrible, horrible way to compensate somebody if you want to align interest with shareholders.

Now I'm going to move on to the birthing of unicorns. This won't be quite as dramatic as *Game of Thrones*, *Mother of Dragons*, but it'll be the financial equivalent.

This is how IPOs (Initial Public Offerings) used to work a long time ago in Silicon Valley. Apple went public in 1980 with a \$1.8 billion market cap. Microsoft in 1986 at \$780 million. Cisco went at \$224 million.

More recently something different has happened. Google went public at \$27 billion and then Facebook waited until it was worth \$100 billion to go public. This is causing anxiety.

These are two people you may or may not know. On this side is Yuri Milner of DST (Digital Sky Technologies). Yuri invested almost a billion dollars in Facebook at about a \$10 billion valuation as a private company. I don't know exactly when he got out, but it's worth \$300 billion today, so you run the math.

This is Scott Shleifer of Tiger [Global Management]. He invested \$130 million in JD.com as a private company and returned about 6 or 7 billion dollars to shareholders.

Now if you're working for one of these firms and you see companies going public later at \$100 billion and you see people like Yuri and Scott making tons of money by investing in private companies, you have FOMO, fear of missing out.

It's a well-studied behavioral science problem that causes misjudgment error, and these firms and many others and probably many of your firms decided they were going to do what Scott and Yuri did and dive into private market investing, which created the unicorn. That's exactly what created the unicorn.

Today there are over 200 private companies that are valued at over a billion dollars, and most of them have raised over \$100 million each.

I call this the great experiment that's never been done before because we've never crammed this much private capital into immature private companies ever. We've put way more money into these companies than was ever considered in 1999, and there will be consequences.

Let me tell you a little bit about what's reinforcing this and making it worse than it could possibly be. Venture Capital has put money in entrepreneurs. They then raise money from late-stage investors and mark up what the company is worth. That price is then shown to their limited partners (LP) who then give more money to these guys because they're doing such a great job.

They even get paid. They get paid on the fact that this guy got this guy to write a check at a higher price, and it's measured that way. Now let me show you the most important slide in my entire deck.

My industry has record high IRRs (internal rate of return) right now and almost zero liquidity, no M&A (mergers and acquisitions), no IPOs. Is anything wrong with this? This is a classic mark-to-market accounting problem, and it's going to come unwound. It has already started to come unwound.

You've probably read about this. I noticed it's on the cover of the *Wall Street Journal* again today. Raise money at \$9 billion. LPs marked it at \$9 billion. This is a company called Zenefits. They were doing about \$25 million in revenue and raised money at \$4.5 billion.

That \$4.5 billion was then marked to the LPs. Not only do they get the bonus. They make distributions based on these returns. They told people they'd do \$100 million in sales in 2015. They did about \$60 [million]. They've since had to fire their sales force.

A board member told this guy, Parker Conrad, he wasn't being ambitious enough. So he wrote a program to automate filling out compliance exams for their sales force, and there are probably going to be criminal investigations. It's not worth \$4.5 billion.

Palantir's financials leaked a week and a half ago. This company's raised money at \$20 billion, and it's been marked that way to the LPs and distributed.

It says they did \$400 million in sales in 2015 with a 50 percent growth rate and were unprofitable. Most of their work is really consultant-like work. So what is a consultancy that's unprofitable at \$400 million growing 50 percent worth?

I asked a group of investors. No one kept their hand up after \$1.5 billion. So \$1.5 billion marked at \$20 billion is a huge differential. By the way, within the past three months there were secondary trades around \$17 billion.

This is what's happening. This is how it's coming undone. As a result you're seeing markdowns across the board

and across many firms, and then unicorns aren't so beloved anymore.

What's interesting once again is that in an attempt to find returns, you get the exact opposite. You get losses. How does this happen? What did they miss?

First of all, this strategy works great if DST and Tiger are doing it by themselves. But when everybody jumps in the game, it gets a lot harder, and this is just classic consensus versus non-consensus thinking, contrarian investing.

Here's the bigger factor: the money affects the ecosystem. They assume that if you do this, there won't be any ramifications, but there are many, many problems. These companies have raised more money than ever.

These companies don't have capex. They don't build stores. They don't build factories. If you give them more money, they hire more people and they create bigger burn rates, and they get further away from their core unit economics. It's hard to know if they'll ever bring it back together.

One thing I didn't put on here: a lot of people think that Steve Jobs's greatest contribution to design was that he created constraints. He wouldn't tell the team what to do. He would tell them it has to be this thin or this tall or weigh this amount, and you go figure it out.

I think the same thing is true with startups. If you have a constraint, you make better decisions. If you have unlimited constraints, you do everything. You make worse decisions, and I think the quality of execution in these companies is way worse than it would have been had you not handed out all the money.

Then it creates excess competition, and I think this is now impacting the mid-market public Internet stocks because there's just such an excessive amount of money.

So if any of you track the new lending companies, OnDeck's stock is at \$5 and Lending Club's is at \$3. Well, SoFi and Avalon have raised a billion dollars of private equity, and the SoFi CEO said the reason Lending Club's not working is because they're not ambitious enough.

Now I personally don't think lending and ambition are two things you want to combine, But the problem is excessive competition. You can go through almost every vertical and there's someone out there who's willing to lose \$200 million, \$400 million, in this case \$600 million in a year, and it's very hard to be profitable if your competitor's able to do that. So this is just overfunding the category.

Then lastly the IPO process provided immense value that people didn't think about. Companies prepare for this event. They take it remarkably serious. They get GAAP compliant. The auditors take it more seriously.

This is probably just an amazingly hard thing to fathom. The auditors try harder when you're about to go public than they do when you're private. I guarantee you. And that's why you get statements right before you go out and this kind of thing because they send it to national.

Most of these companies are raising money without audited financials. Some of them, they get the audit. In a private company you might get audited financials for 2015 done in November of 2016, so no one knows if the math that's put on the PowerPoint is even going to be accurate, if it were properly accounted.

The bankers, everyone makes sure you're ready, and companies and management and boards consider the weight of being public, and they don't think about this relative to being private.

Here's a quote from Mike Moritz who's one of my peers at Sequoia. He wrote a fabulous article about "subprime unicorns." It's not very long if you want to read it, it was in the *Financial Times*. And he says, "It is easier to conceal weaknesses, present an aura of invincibility and confound investors as a private company that can

escape by making fewer disclosures than as a publicly traded one.”

I would say that we created a playing field ripe for over-promotion because everyone was being encouraged to be a unicorn, and people were putting together PowerPoints that were fantastical and didn't adhere to any of the same principles that you normally would see in an IPO process. I think you're going to see a lot more companies like the three that I highlighted over the next 18 months.

This one's really short, and then I'll turn it over to Q&A. This is actually someone I know well, a guy named Nick Hanauer. He was an early investor in Amazon.com and then invested in this company, Avenue A/aQuantive that Microsoft bought for \$6 billion in ad space.

So he's done really well for himself. But he's become a very outspoken person on minimum wage, and he pushed to have Seattle go to \$15. And now he wants Seattle to go to \$28. Others have followed suit. You probably read about New York and California now going to go to \$15.

When Nick talks about this, I'll just highlight one of the things he says is most of the job growth has been in service jobs, and he says 39 percent is in food. He talks about waiters and waitresses.

I live in Silicon Valley, and I was an investor in OpenTable, and there are some new companies that look a little bit like OpenTable that often call on me. One of them is called E la Carte, and this is their product called Presto. and the company's just been kind of middling along. And this one is in Dallas called Ziosk. These things sit on tables and replace waitresses and waiters.

I went and visited him two weeks ago, and his inbound leads have never been higher there in Seattle, New York, and California. I don't think Nick intended to accelerate the demise of waiters and waitresses, but he may have in fact done so with his efforts. And these are articles about this technology now taking off. I am highly confident in low- and mid-service restaurants that these will become pervasive in the next five or ten years.

So here's my summary on this. In each of these cases, even in the Uber case, I think you had a biased and passionate advocate who had a problem they wanted to solve that they just felt extremely passionate about.

In the professor's case, I think he just thought that the investors were being silly, and he wanted to prove that they were being silly. In the other three cases, I think people felt like either something wasn't fair or it was a problem they needed to fix, and so they were looking for a very blunt instrument to fix it.

I think all of these things exist in very complex systems, and that's why I reference the book *Complexity*, because you have interconnected state machines, multiple variables. You have human behavior. And I think you need systematic thinking to look at these approaches. You have to think through all the inputs. How is A going to cause B going to cause C? And what are all the ramifications that are going to play out when you move to a new solution?

I don't think they did that in any of those last three examples. Then the ultimate irony is in three of the cases you end up in a worse situation than the problem you intended to fix, which I think happens frequently. Those are my prepared remarks at 42 minutes, so I'm three minutes under where I was supposed to be.

Question: Thanks very much. You mentioned you'd only heard of Professor Damodaran two and a half years ago, but what you may not know is among quants he has a tremendous reputation, not because he's as exciting a stock picker as Jim Cramer, but because he's put his valuations online, in spreadsheets, and has demonstrated that you don't have to understand the companies a lot. Some fairly simple metrics can give you good answers.

So I have a hypothetical for you. Let's say you wanted to know the total value of all private technology

companies. Would you trust the professor's calculation or the sum of a survey of the board members of these companies?

Bill: The professor. Yeah, I would trust the professor. The reason I say that is there's an amazing amount of optimism in my industry, and maybe it's a requirement for the job.

I don't think if you were just inherently a skeptic you'd do very well in venture capital, but there's just an inherent amount of optimism. And this goes in cycles, but in my industry in the past five or ten years, a lot of money's been given to people who have never been investors.

They celebrate their operating history, but they don't reflect whether they have any investing credentials or credibility, and I don't think they understand even how capital markets work. That's part of what's led to the problem.

Question: This is really interesting. Thanks. You gave examples where humans got it wrong and missed by a mile. One of your points in your conclusion was these instances require systematic thinking.

Do you believe that machines would have been better? And in particular in some of the examples you gave, I find it hard to construct a narrative where we could build a machine that would have figured out what the people didn't.

Bill: This is a subject that I think other speakers are going to have a lot more knowledge on than myself. I'm exposed to it a little bit because my industry tends to get hyper-excited about certain themes and go a little nuts, and one of them right now is machine learning and AI (Artificial Intelligence), and we've studied internally the types of problems that work well and those that don't and ones that might be investable or not.

I think that the types of problems that I highlighted would probably be some of the last things that would be possible just because of all the complexity that I mentioned. Some of these decisions are based on how other humans react, and that's constantly changing. It's just a very complex surface area but, once again, I'd maybe re-pose the question to someone who knows more.

Michael: Pedro, do you have a thought on that?

Pedro: Boy, do I have thoughts on that. [Laughter] I mean my thought is that, yes, I agree with you. These are probably some of the harder things for machines to do today.

I think there are a lot of things that a machine learning system might notice that people wouldn't just because it's looking at a lot of signals that people aren't, that people have forgotten about. Suddenly one of those signals starts to be really clear, and I think there are cases where that's happened, and that might be possible with some of these things.

One advantage that the machines have is that they can look at more of the picture than the humans can. If that means they get more confused, then it's bad. It also means that they actually understand what's going on better.

Your reference to Mitch Waldrop's book I think is a very interesting one because this is really what it's all about. The behavior of the complex system is different from the behavior of the individual parts, and people just look to model the behavior of the individual parts.

Today most machine learning algorithms are also just modeling the behavior of the individual parts. But the better algorithms actually start to model systems as a whole. And I think once you do that you can do a lot better.

Bill: Of all of them, I think that compensation would be the one that it could potentially figure out and nail if I'm right about my thesis.

Question: How do you see the imbalance on your scale slide playing out, and how does that impact what you're doing?

Bill: I think about that every single day. The part I left out in the unicorn story that I think is inherently true is that the low interest rate environment on a global scale is having a remarkable impact on the amount of money that's willing to come into my field and others.

Even today after those stories have blown up, I could probably take 20 meetings with people who want to put more money in my industry in the next week, and these are inbounders that we're deferring. A lot of it's coming globally, out of Russia, the Middle East, China, looking for global diversification. There's just vast amounts of money, and so I don't know what it will take.

A lot of these companies also have raised so much money that they don't run out right away. And so this shake-up started October/November 2015. It'll probably take a full 18 months after that when people literally run out of money before you start to see catastrophic effects or behavior change.

Now some of the unicorns I've said, though, become zombies. There's a lot of different metaphors. I think companies like Zenefits have done massive layoffs of their sales force. What I mean by zombies is they've given up on the growth trajectory that was used to raise the money at the super high valuation, but they have the cash, and so they downshift, and they're no longer chasing the gold they once were. That company might stay around for a very long time.

Another factor that played a role, especially because some of you might be considering joining this wonderful dance, is a term in private investing called "liquidation preference."

It basically says that if you want, you can get your money back and not convert to common. And so if a company sold for less than you paid, you can still get your money back. And that's how the term works.

Most of the people that came into the market, like many of the names on that chart, thought that it would act like debt, and so they thought their money was safe. Price doesn't really matter because I just want an option on this being the next Google or Facebook.

What they don't understand is that returns paid out due to liquidation preference are a vast minority. What happens when these companies stumble is they get recapitalized, and a couple of them like Foursquare and Jawbone have already been recapped, so your liquidation preference goes away. The board just votes and elects or the shareholders wipe you out, and most of the terms in these unicorn deals don't have protection against that. So it'll take a while.

I had hoped that it would happen like in 1999 or 2001 because I feel a duty and obligation to perform on those IRRs that our LPs have already been bonused on. [Laughs] It's very hard to do payouts in this environment. It's very hard to get to liquidity with the weird expectation mismatch, with the ability to raise money at private prices that are vastly different than public prices. The M&A's not going to happen until the expectations shift.

Question: Are there things that are underappreciated publicly now about the Lyft/Didi/Uber phenomenon where you feel highly confident that Uber succeeds? One, and then if could sneak in two, the phenomenon of WeWork.

Bill: [Laughs] Is that a question?

Question: Yeah.

Bill: [Laughs] All right, we'll pause for WeWork. The thing that's happening with Uber's competition—you

mentioned Didi in China and Lyft here in North America—is tied to the capital environment that I'm talking about, and so Lyft was burning.

They made a decision in 2015 to aggressively push the gas, and I don't know what their burn rate was but they went up to \$25 million a month. Then when they got some money from General Motors, they went to \$50 million a month.

This is published. It's out in the press. So there's a \$600 million run rate, and they started what I call renting market share, buying market share. They might call it earning. I don't know.

What's going on is if you had a systematic or scale advantage, in a normal company you might have one company with no profits and one with 30 percent profits, and you say they have a scale advantage or a network effect.

In this case there's so much capital available that those things are happening with losses. One company loses way more money per ride than another company loses per ride. None of this information's public, so everybody's guessing what these numbers are, and you just won't know.

There's a famous Buffett quote I think where he says something like, "You don't know who's naked until the water goes out." And I think that's the situation we're in, in this case.

One of the things that's happened that I think was unpredictable was how many people have come around to the view in the video as opposed to the professor's view, and there are lots of interested parties as a result.

On the second one, WeWork. [Laughs] Most of you probably don't even know who WeWork is. We're an investor in a company called WeWork that's very atypical for us as venture capitalists.

This entrepreneur, who's really an amazing entrepreneur, named Adam Neumann, decided that we were undergoing a cultural revolution in how people like to work and live, and that you could restructure office space to meet that need. He basically rented whole floors of tall buildings, and he retrofitted them with much smaller individual work units but a lot of glass and common areas and common meeting rooms.

If you ever go into one of these places, and you really have to understand what WeWork is, there's a totally different vibe than in a historic shared office space environment. And if you work for a two-person PR firm, you feel like you're going to work with a lot of people as opposed to just going in and closing the door and not knowing anyone else.

There's a big urban shift with millennials and a lot more people doing independent work And he believes this ties in to all that. And he's been quite successful at filling these and extracting a rent per square foot that's dramatically above what he paid.

The company gets a lot of questions—about how it should be valued, is it just a leased tenant kind of thing, should it be valued like that, or should it be valued as a tech company which I think Adam would try and argue.

He recently launched a new product called WeLive that is basically a dorm for post-college people. And if you look at rents in San Francisco, I think most starting rents are three grand a month. And so he'll be providing a product that'll compete at like \$1,100 a month and also will be more social than the others. So I don't know exactly how to value it. Different people in this room I think even have valued it fairly highly.

Question: A question on this IPO situation because from where I sit it's actually easier to IPO a company than it's ever been. This dichotomy of these companies staying private for as long as possible just doesn't make sense. On one hand you have an enormous amount of companies that are staying private, and on the other

hand, it's never been easier to go public.

Bill: So look. The change to the Jobs Act, especially the one that allowed the first filing to be private, has been awesome. And you get two or three iterations with the SEC (Securities Exchange Commission) for those that aren't public, like Groupon had and some of these other high-profile ones, and then your window from flip to price is a lot shorter, which is way positive, all good because there's less risk in that period.

There are people that started a rhetoric four or five years ago in Silicon Valley that staying private was better. I think there are certain entrepreneurs who were open to that message. Entrepreneurs who don't like scrutiny, , and think that anything that's rule-oriented is bureaucratic.

They put a whole narrative around it like it's harder . . . they blame it on the stock market's short-term thinking, saying they want to do long-term things and no one's letting them. I'd like to highlight that it never stopped Jeff Bezos or Marc Benioff or Reed Hastings from doing anything.

But I think they're basically afraid, and they don't have to be. And some of these Silicon Valley boards, even in the unicorns that you guys have helped fund, are just printing secondary.

Palantir did a secondary way above \$10 billion, and so the entrepreneur's getting paid, and they don't have to go through the scrutiny. I've told them, "Look, the minute you took on shareholders and the minute you granted options to your employees, you were on a course. You have an obligation, and if you don't feel that way, you should get out of the chair." That's how I feel about it.

Imagine a quarterback who had a great college career saying, just a week before the draft, "You know I've decided I'm not going to play." And they go, "Why?" And he goes, "Well, the scrutiny on Sunday is going to be horrible. They're going to track every play, every pass. They're going to record every metric. It's going to be horrible. I can't operate. I can't plan my long-term career with that kind of . . ."

If a quarterback did that, he'd be laughed out of the game. But there are people that are acting that way. And so it's been a playground for the past three or four years. And by the way you can go read these articles.

A new CEO went into Evernote and found out they were giving away two days of house cleaning services to every employee. The perks are off the hook. No one's looking at profits. No one cares.

Question: To return to the compensation discussion, Charlie Munger said something to the effect of, "Never before have so many that earn so much earned so little." And I think it's one of the things that has changed in my lifetime in finance. When I first came in, the people who made a lot of money were the partners at the end of their career who had shown integrity and hard work and developed things. It certainly has benefited my bank account today where pay for performance in the short-term has become a lot more, and I think through your discussion it's really in the tech sector too.

A lot of people got really rich at the end of the century. I think one of the things in our space that we need to come back to is I think society really respects people who create fantastic organizations: Jeff Bezos, Warren Buffett, Steve Jobs. But I think there's this resentment of people who came in at the right place, right time, or heads I win, tails you lose.

Don't we need to move the compensation system to ten-year payouts? Turn it so that you're partners again in the business, and that you live and die with this business. To that end, you create benchmarks, and people should be benchmarked to that industry space.

In our space, the benchmarks I think have become much more prevalent. So I was just curious about the question around compensation.

Bill: A couple different things I'd say to that. One, in non-tech sectors I've noticed from looking at other proxy statements, which I do occasionally, is that they've moved to performance-based RSUs.

The thing about that is there are two problems with performance-based RSUs, although I think they're better than just simple RSUs. They requires the board to know what creates stock value, and Michael could lead a discussion on whether that's possible or not. Then second, often when they miss the metrics, you'll see the board decide to pay it anyway. Those are two problems with that approach.

The problem in Silicon Valley that you have, and maybe it's a problem with compensation in general, is it's hard to go backwards. It kind of only goes forwards. And that's a competitive thing driven mostly by Google, where Google is now paying their top executives what I like to call Alex Rodriguez money. And it's all what they call GSUs (Google Stock Units), but it's the same thing.

It's essentially a cash paycheck of 15 to 20 million dollars a year, and our startups compete for the same talent with companies that are paying that out. So I don't know how you correct it. It's part of another issue in Silicon Valley with all these competitors having so much money.

We're in a company called Hortonworks, which is public. Their private competitor, Cloudera, one day raised \$900 million. You say, well, what do you do? The playing field gets messy for everybody.

You can't choose not to play. If you do, you yield the field and you lose all the customers. The same thing is happening with this compensation issue. I could be idealistic and try and have a ten-year option period, and I might not hire anybody in Silicon Valley.

Question: Thank you, Bill. You recently wrote "On the Road to Recap", where you listed a bunch of challenges. Two related questions.

One, what pushback or criticism have you received on that from VCs or otherwise?

Two, do you accept the thrust of the premise in some of the stuff you said today in saying you want to hedge your unicorn exposure or get the other side, other than public market stuff or people who say lease space? Have you guys thought of . . .

Bill: Thought of hedging?

Question: . . . Hedging, or how do you take the other side?

Bill: I've thought about it. [Laughs] I've gotten mostly positive feedback on that piece that I wrote. I've gotten some negative feedback from VCs that were out trying to accelerate their fundraising on their paper marks which is one of the points that I made, and it's definitely been going on.

Q1 of 2016 had record high new LP money into new venture firms. I've even heard some of them are done with their commitments for the year because they have so many allocations they can do. So there's a rush to raise which is just a signal that they know what's going on. Hedging. Yeah, I've thought about it a lot. There are certain stocks that you could look at that have exposure.

I look at something like Rocket Internet. There's very little chance that that stock will do well if our companies don't do well. But I've never actually put anything on, and we've never hedged internally.

Some firms have started trying to sell. Founders Fund has sold a few positions. They sold a Lyft position and Andreessen sold a Lyft position right when GM was buying. It's not something that's typically done in our industry by people that sit on boards, but there are people starting to do it now. The minute everyone starts to do it, it'll accelerate the fall. Anyone else?

Michael: I do. What's exciting? I actually saw an interview you did and you talked a little bit about some work you did looking at health care. To be more positive, what kinds of things are exciting to you in the next few years?

Bill: [Laughs] To be more positive. I'm an optimist. Let me just mention the health care sector. So there are other investors who have spent way more time in it than me, and I probably spent two and a half years looking at it. I just now feel comfortable making bets, and I've actually made a few very early-stage ones.

It's a segment that's ripe for technology to add value to. I've been involved with other vertical players like Zillow or GrubHub or OpenTable or Uber, and you say, "Boy, you should be able to use this smart phone and all this technology and fix this problem." It's a ridiculously messed up industry.

I have this theory that democracy and capitalism destroy one another if you give them time, and that decay is happening most in telecom, health care, and finance where you have the most regulation.

The incumbents are very adept at using the regulation to prevent disruption—things like HIPAA (Health Insurance Portability and Accountability Act). People think HIPAA is looking after their best interests. HIPAA's looking after the incumbent's best interests. It makes it almost impossible to share data, and all the solutions you would build to fix the problem require sharing of data, and there's just all this kind of morass that happens.

I think every entrepreneurial endeavor someone takes on assumes a market-based approach, and our health care system is not a market-based approach. The people paying for it aren't the buyers. I think most people are ignorant of the Recovery Act. We paid doctors to implement electronic health record (EHR) systems, and we paid them 44 grand each.

It's like an ERP (Enterprise Resource Planning) for doctors. To think about the ignorance of this decision is just mind-numbing to me. Forty-four thousand dollars to not the top one percent but the top thousandth of a percent to put in software that they didn't want to put in anyway. And the reason they don't want to put it in is because they're not in a competitive environment that requires them to evolve.

Then it gets stupider if that's a word. The thing you'd worry about is if a doctor was paid to put in something is that he wouldn't use it. Right? So two years later they give them \$17 thousand if they can prove they're using the software you paid them \$44K to use. And this came from our federal government. They wrote billions and billions of dollars of checks to doctors, the downtrodden doctor, to put in software.

By the way, this is how hard it is for a startup to do well. In order for your software to qualify for the EHR payments, it had to have a certain set of features, and there are Excel spreadsheets you can find on the Internet where the government lists the product features required for a software solution. Now is there any chance a disruptive software solution would be built that way? Zero. And that's how messed up it is.

So it's just hard to build systems that change when they're structured that way. I'm enamored with the—talk about incentives—the Singapore health care system. We're at say 17, 18 percent of GDP (Gross Domestic Product). Singapore's at four. And on any broad-based health metric you can't find a difference. And what they do is everyone's a payer.

The rich pay 80 or 90 percent of their bill. The poor pay 10 to 20 percent of their bill. But no one's taking on work that they're not shopping for. I quite frankly think we need to get the employer out of the game. There's no reason the employer's in the business. There's all kinds of stuff that needs to happen. It's difficult.

Michael: Any other areas that are more positive but not health care?

Bill: There's a company that we're working with called Stitch Fix that I mentioned last night that is taking a

Moneyball approach to women's fashion, which is something you wouldn't think would be possible. Each customer that comes in fills in a 15-page profile where they talk about their size, their style, their geographic area, whether they buy clothes more for work or for going out and that kind of thing.

Every item that comes into their inventory we collect 67 metrics on. How wide is the shoulder? How pliable is it? The colors.

Then we start sending products to people and looking at what they keep and don't keep. And there are over 60 data scientists in the company. They study the patterns, not only of an individual but also of groups and how they apply to different things. It's at the point now where the merchandisers that are creating new products will test against our algorithms before they even build it.

So that's an exciting use to me of machine learning in a way that's applicable. Everybody else is saying that it will give you an AI bot that you can chat with. It'll be a lot of fun.

There was this great meme. If you're into AI and chatbots, there's this great meme going around Twitter of two circles that were completely separate, and one of them said, "Things bots do," and the other one said, "Things humans need." [Laughter] There's no overlap.

Question: At the end you mentioned a little bit about paying attention to labor share and GDP and technologists, and we're kind of seeing some folks vote, and obviously that dictates a little bit of policy. How do you feel that transition evolves over the next five to ten years?

Bill: It's a question way above my pay grade because of all the different things it involves. I go back to the fact that a few short couple hundred years ago like 98 percent of our population were farmers, and today it's less than one. We made that transition, and I don't think anyone regrets that we made the transition.

That's going to happen in a lot of new fields because of automation. I don't know the solution. I don't think you can stop it, and I think if you tried to stop it, you would actually slow progress on a global basis—not for the individual that's impacted but for everyone else.

There's a book that I'm a huge fan of called *The Rational Optimist* where Matt Ridley talks about the biggest increases in the standard of living. They're typically around innovation and the sharing of ideas and open capitalism.

China's unlocking for the past 20 years has probably been the biggest impact to standard of living. So slowing technological change I don't think helps the global standard of living, but it might help an individual's situation. But again that might slow it for everybody else. So it's a difficult problem.

One thing that we could all do—and there's actually a group of entrepreneurs out of Seattle that have created a nonprofit, I think it's called Code.org—is just promote programming everywhere. [Laughs]

There are engineers being hired into Silicon Valley companies out of university at \$175K right now, and there is a complete undersupply of jobs. China puts 35 percent I think of students into engineering, and we put five or something.

We've created a social and cultural bias, the whole nerd thing. That's a real problem for our country and for this issue, and so do everything you can to get coding pushed into middle schools, high schools, that kind of thing.

Michael: I think we'll call it there. Thank you very much, Bill. Great.



Pedro Domingos
University of Washington

Pedro Domingos is one of the world's leading experts in machine learning, artificial intelligence, and big data. He is a professor of computer science at the University of Washington in Seattle and the author of *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. He is a winner of the SIGKDD Innovation Award—the highest honor in data science—and a Fellow of the Association for the Advancement of Artificial Intelligence. He has received a Fulbright Scholarship, a Sloan Fellowship, the National Science Foundation's CAREER Award, and numerous best paper awards.

Pedro is the author or co-author of over 200 research publications, and has given over 150 invited talks at conferences, universities, and research labs. He received his PhD from the University of California at Irvine in 1997 and co-founded the International Machine Learning Society in 2001. He has held visiting positions at Stanford, Carnegie Mellon, and MIT. His research spans a wide variety of topics, including scaling learning algorithms to big data, maximizing word of mouth in social networks, unifying logic and probability, and deep learning.

THE MASTER ALGORITHM

HOW the QUEST FOR the ULTIMATE LEARNING MACHINE WILL REMAKE OUR WORLD

PEDRO DOMINGOS



CREDIT SUISSE

ink factory
you talk. we draw. it's awesome.

Michael Mauboussin: I'm thrilled to introduce our next speaker, Pedro Domingos. Pedro is a professor of computer science at the University of Washington and a leading expert in machine learning, artificial intelligence, and big data.

Last fall, I had a series of meetings with senior investment managers in London. What struck me was that in every conversation, the topic of machine learning came up unsolicited. Determined to learn more about it, I read Pedro's book, *The Master Algorithm*. I found the book useful and illuminating on multiple levels, and recommend it highly. By the way, the money page is 240. Make a note of that. You'll hear more about the book in a few moments.

As I mentioned a moment ago, you can listen to Pedro for a primer on machine learning and to get a sense of where these capabilities may take us. But you can also listen to understand how the various approaches and challenges in machine learning apply to everyday thinking. Machine learning and AI represent a new and very exciting means of attaining knowledge.

Please join me in welcoming Professor Pedro Domingos.

Pedro Domingos: Let me start with a question: where does knowledge come from? Until recently, knowledge came from just three sources. The first one is evolution. That's the knowledge that's included in your genes. The second source of knowledge is experience. That's the knowledge that's included in your neurons. And the third source of knowledge is culture. It's the knowledge that we acquire by talking with other people, reading books, and so on.

Now, what's new in just the last few decades is that there is a new source of knowledge on the planet, and that's machine learning. It's computers. Computers are discovering new knowledge from data.

I noticed that the emergence of each of these new ways of discovering knowledge was a major landmark in the history of life on Earth. I mean, evolution is life on Earth itself. Learning from experience is what distinguishes mammals from insects, and culture is what makes humans as successful as they are. It's what makes us who we are.

I think that computers as a source of knowledge are going to be every bit as momentous as every one of these three, and we are just getting started. Already, we see a lot of the impact. Notice also that each of these new sources of knowledge, operated orders of magnitude faster than the previous sources when they first appeared on the planet. So learning from experience is orders of magnitude faster than learning from evolution, and learning from culture by just hearing something that somebody tells you again is a lot faster than learning from experience.

And learning from computers is going to be even faster. Computers can discover knowledge at a rate that is unimaginable for human beings. Corresponding to that greater speed, you also discover orders of magnitude more knowledge with each of these new ways than you did previously.

In fact, Yann LeCun, who is a well-known machine learning researcher and now the Director of AI Research at Facebook, says that in the future, most of the knowledge in the world will be discovered by computers and will reside in computers.

So I think we're at a point where all of us need to understand, not at a necessarily very detailed level, but conceptually, what machine learning is and what it does. That's what I am going to try to do in this talk.

Here is machine learning in one slide. In traditional programming, there have really been two stages in the Information Age. The first stage was where we programmed computers to do things. When we want a computer to do something, we have to write down an algorithm in painstaking detail explaining how that computer is

supposed to do it.

And then, this is what happens: there is the computer, in goes the data, in goes the algorithm, and the algorithm does something to the data to produce the output. For example, in goes an X-ray, in goes an algorithm to diagnose cancer. Then it says oh, there's a tumor here, and the output is either, yes, there is a tumor here, or no, there isn't one. This is how everything in the Information Age has been built until recently.

Machine learning turns this around. What happens in machine learning is something that is very strange at first sight but makes a lot of sense. Hopefully it will to you after this talk. The output is actually now going in, and what comes out is the algorithm. So what we give to the computer is the data that we want it to operate on and the output that we would like it to produce. The computer figures out how to turn this data into this output, and then it produces that in the form of an algorithm.

Then that algorithm goes and does its usual job. For example, this might be a bunch of X-rays and this might be the diagnosis for each of those X-rays. The diagnosis says yes, there was a tumor here, or no, there wasn't a tumor here, and the machine learning figures out how to decide whether there is a tumor or not by looking at the pixels in the image. That goes on and on, and then a bunch of new patients come in and it starts doing this automatically, at a fraction of the cost of a human pathologist and better. An algorithm that learns to do this in half-an-hour does better than someone who was in med school for many years.

The amazing thing is that in the old way of doing things, for every different thing that you wanted to do, you needed to write the new algorithm. If you wanted the computer to do the medical diagnosis, you had to program it to do the medical diagnosis. If you wanted the computer to drive a car, you had to program it to drive a car.

But in machine learning, the same learning algorithm can do an infinite array of different things depending on what data you give to it. A learning algorithm is a master algorithm in the sense that it's an algorithm that makes other algorithms.

In principle, the same learning algorithm, if it's powerful enough, can learn absolutely anything that you want it to learn provided you give it a sufficient amount of the right data. So how does this actually happen? Well, there are a number of different paradigms in machine learning, a number of different ways of learning new knowledge, of extracting programs and models from data.

One other thing—machine learning, in addition to being very useful and economically important these days, is also a lot of fun and very fascinating because the main ideas in machine learning all come from different fields.

In fact, there are five main schools of thought in machine learning. Each one of them has its roots in a different field, and each one has its own version of how to do things. Each one has its own master algorithm: an algorithm that if you give it data from any problem, in principle, it can then learn to do what needs to be done for that particular problem.

The first tribe that we're going to look at are the symbolists and their origins are in logic and philosophy. They are the most linked to computer science of the five tribes, and their master algorithm is something called inverse deduction, which sees induction as being the inverse of deduction.

Then there are the connectionists whose idea is that we're going to learn by reverse-engineering the human brain. Your brain is the greatest learning machine on Earth, so let's figure out how it works and do that on the computer. They're called connectionists because it's all based on this idea that your knowledge is included in the connections between your neurons, and their master algorithm is something called backpropagation.

Then there are the evolutionaries who say that the master algorithm is not your brain, but rather evolution. They

want to figure out how evolution works and simulate that on the computer, and their master algorithm is genetic programming.

Then there are the Bayesians who have their origins in statistics, and their biggest concern is with the uncertainty of learned knowledge. All knowledge that is learned from data that is the part of induction is necessarily uncertain, so the idea of Bayesians is to quantify that uncertainty, and so, their master algorithm is probabilistic inference. It's computing the probabilities of different hypotheses based on the evidence that you have.

And finally, there are the analogizers who actually have their roots in many different fields. The most important of these fields is probably psychology. The idea here is that most of the learning and reasoning that we do is by analogy. It's by finding similar situations to the ones that we are in now, and then trying to extrapolate from one to the other. Their most widely used algorithm is something called a kernel machine, also known as a support vector machine.

So let's start by visiting the symbolists and seeing what they have to propose. Here are some of the most prominent symbolists in the world: Tom Mitchell at Carnegie Mellon, Steve Muggleton in the U.K., and Ross Quinlan in Australia.

The basic idea behind this type of learning, that learning is induction, actually goes back to a 19th century philosopher and economist named William Jevons. Induction is going from specific facts to general rules whereas the inverse is deduction, which is going from general rules to specific facts.

Some of us can figure out how to do induction in the same way that mathematicians, for example, figured out how to do subtraction because it was the inverse of addition, or how to do integration because it's the inverse of differentiation, and so on.

In mathematics, this has a very long and distinguished history. So for example, addition gives us the answer to the question what is two plus two? It's four, of course. That's not the deepest thing I'm going to say in this talk.

Subtraction gives us the answer to the inverse question, which is what do I need to add to two in order to get to four? And so, my idea of inverse deduction is actually to do the same thing but with induction.

For example, deduction gives us the answer to a question such as, "If I know that Socrates is human and that humans are mortal, then what can I infer about Socrates?" And of course, it's that Socrates is mortal. We know how to do this very well. Deduction has been very well understood in logic and computer science and philosophy for decades or even centuries.

Now, the tricky problem is induction. Induction is the answer to the question, "If I know that Socrates is human, what else do I need to know in order to be able to infer that he's mortal?" The answer, of course, is that humans are mortal.

But you try and figure that out. If you can fill in this gap, now you have a new general rule that you can go and apply to many other things in combination with other rules to answer questions that you may have never thought of.

By this process of filling in the gaps in your knowledge using inverse deduction, you build up a knowledge base of rules that is very powerful. In particular, this idea that you can combine different rules in different ways is something that only the symbolists can achieve. None of the other schools of machine learning actually have that capability and it's a very important one.

Now, I wrote all of this in English. Of course, computers don't understand natural language, so in a computer, this is actually usually done using a formal language, such as first order logic. But the idea is the same.

Symbolist learning is a bit like emulating the scientific method. It's saying: here's some data, let me formulate some hypotheses to explain that data, and then test those hypotheses against new data, and then maybe refine them or discard them, and so on.

What we're really doing here is just automating the scientific method, except we're doing it much faster and on a much larger scale. And one concrete instantiation of this is what you see in this picture. The biologist in this picture is actually not the guy in the lab coat. The guy in the lab coat is a machine learning researcher by the name of Ross King.

The biologist in this picture is actually this machine. This machine is a complete robot biologist in a box based on the principle of inverse deduction. They started out with a robot named Adam, and now the name of this robot is Eve. There is only this one in the world so far, but of course, what's interesting is that once you have a robot like this, nothing stops you from making millions of them, and it can make progress in science millions of times faster.

Eve looks at, for example, the biology of a particular cell type. It starts out with data, formulates a hypothesis to explain that data by inverse deduction, then it designs experiments to test those hypotheses, and then it carries out the experiments using gene sequencers and DNA microarrays, which is what's going on here. And then, it repeats the process. So it really is a complete robot scientist. In 2014, Eve discovered a new malaria drug. So this is the kind of thing that you can do with this type of learning.

Now, the connectionists are skeptical about all of this. They say this type of learning is too abstract, too clean. The way most learning happens is not the way a scientist or a logician or even a philosopher works. It's messier. It involves making mistakes and being embodied and all sorts of things like that. The idea of the connectionist is that our competition in machine learning is the human brain, and we are far behind the competition.

So what you do in tech when you're behind the competition is reverse engineering. You start out by copying it. You open up their chip and you see what the circuit is, and you figure out how to do the same thing. The connectionists try to do that with the brain. The brain is inside the skull and it's got circuits and whatnot, and we can try to figure out how it works. And this has indeed been a very productive approach to machine learning.

The most famous connectionist in the world is Geoff Hinton. He actually started out as a psychologist in the '70s, and these days, he's more of a computer scientist. He actually splits his time between the University of Toronto and Google.

Jeff believes that the way the brain learns can be captured in a single algorithm, and he has spent the last 40 years trying to discover that algorithm. In fact, he tells the story of coming home from work one day very excited saying, "I did it! I figured out how the brain works!" And his daughter replied, "Oh, Dad, not again." [Laughter]

Now, he is the first to say that he's had his ups and downs, but his quest is starting to pay off. In particular, he is one of the inventors of this backpropagation algorithm which is used everywhere. Like, if you have an Android phone, for example, it's what's doing the speech recognition, and the variety of things that backprop is used for is truly mindboggling. One of its killer applications in the '80s was in finance, predicting stock fluctuations and foreign exchange fluctuations and whatnot. Two other prominent connectionists are Yann LeCun, who I already mentioned, and Yoshua Bengio.

So let's see in a nutshell how this all works. Biologists know roughly how neurons work, and we don't need to know more than roughly how they'll work in order to do what we want. A neuron is a very interesting, it's a very unique type of cell; it's a cell that looks like a microscopic tree. The trunk is called the axon, the branches are called dendrites, and the roots are also called dendrites.

The thing that's interesting about neurons that makes them different from trees is that the branches of one neuron make contact with the dendrites of other neurons. The points of contact are called synapses, and the synapses can be more or less efficient.

Neurons build up a charge in their body, or soma. If the charge exceeds the threshold, then the neuron fires what is called an action potential down the axon, which is quite literally a little lightning bolt. Your brain at work right now is a symphony of these little lightning bolts going all over the place.

Then the charge goes down the axon and it goes to the synapses, and the more efficient synapses transmit the charge better using a chemical process that is not important to discuss now, but the point is neuroscientists believe that everything you've ever learned is encoded in how strong the synapses are.

Roughly speaking, when two neurons fire together, the synapse between them grows stronger, meaning that next time around, the first neuron will have an easier time firing the second neuron. And so, we're going to build a mathematical model of a neuron, run it on the computer, build a big network of these neurons, and this is how we're going to learn things.

Here's our mathematical model of a neuron. Notice that there's a one-to-one correspondence between the blocks in this diagram and the image of the neuron that I had before. This is the cell body, these are the dendrites coming in, and this is the axon. So let's suppose that we have a neuron, and this is your retina right here. The inputs are just pixels. In general, there could be other layers of neurons, but let's say they're pixels.

Now, each one gets multiplied by a weight. Some will get multiplied by a larger weight than others, and these weights are where the learning is going to happen. Learning in neural networks is basically just twiddling those weights.

If the sum of these weighted inputs exceeds the threshold, then the output is one. Let's say I'm looking at an image of a cat and the neuron is doing its job. Some of the features, which are features of cats, exceed the threshold so the neuron fires. Otherwise, it says, well, no, this actually doesn't look like a cat, and the neuron doesn't fire.

This part is easy. When things get interesting is when we have a big network of neurons like this. How do we train it? If you think about it, this is a very hard problem with no obvious answer. I have a huge network of neurons and there is some little neuron here with some weight and the error is happening at the output. The network is saying that this is a cat but it's not a cat. Who is wrong? What weights need to change?

People first thought of neural networks in the '50s, but they didn't know how to solve this problem, and so, things kind of died out.

But in the '80s, they figured out this backpropagation algorithm, which is in essence a way to solve this problem. The way backpropagation works is conceptually very simple. All it's really doing is tweaking each weight in turn saying if I increase this weight, will the error at the output go down or not?

Let's say like this was a cat, this should have been firing, it should have been one, but it was just 0.3. So the error is 0.7, and I need to reduce that error. I need to make the output higher. If I change this weight over here, if it goes up a little bit, does that make my error go down? Or maybe if this weight goes down a little bit, that will make the error go down.

Now, of course, doing it like this one weight at a time would be ridiculously inefficient, so what backpropagation does is handle things in layers. So here is my input, and these purple circles are the neurons. Each neuron computes its value and then the next layer of neurons can compute their values all the way to the output. When

we get the output we compare it with what it should have been: did you say that it was a cat and was it a cat?

And then we see by what amount we were wrong and determine how much the weight should change. So what I do is I propagate the errors, that's the yellow circles, backwards through the layers, and I compute how much each weight needs to change.

First I compute how much these weights need to change, and then based on the errors here, I compute how much these weights need to change all the way back to the beginning. So I'm propagating the errors back through the network in order to decide how much the weights need to change, and that's why this algorithm is called error backpropagation, or backprop for short.

Backprop turns out to be an incredibly powerful way of doing things, and just in the last few years, it's caused a revolution in things like, for example, computer vision, object recognition, video understanding, and speech understanding.

Microsoft has a system, in Skype to be more precise, where it does simultaneous translation for you. You can be speaking English on the phone with someone in China and they're hearing Chinese, and vice versa. This is done by a bunch of neural networks using this type of learning.

Companies like Google, Microsoft, Amazon, and Facebook use these types of networks not just for object and video recognition but also to choose search results, to choose ads to show you and whatnot. In the press, this is often called deep learning these days.

Why is it called deep learning? Because it's training networks with many layers. In the '80s, people figured out backprop but they couldn't really train networks with one so-called hidden layer, which is a layer that's neither the input nor the output. But now people actually know how to train networks with more layers.

Perhaps the best-known example of deep learning is what has come to be known as the Google cat network, which was on the front page of the *New York Times* a couple years ago. What the Google cat network does is learn to recognize all sorts of objects from watching YouTube videos. It literally watches hours and hours and hours of YouTube videos, so maybe it should be called the couch potato network.

Some people actually think that all it recognizes is cat, but no, it recognizes cats and dogs and mice and people and whatnot. The reason the reporter picked cats as the example is that this is the category on which the network does best. This is because, I don't know if you know this, but people really like to upload videos of their cats. And so, there is more data on cats than on any other entity.

Now, the evolutionaries say, well, sure, backprop might be good for tweaking the weights of the brain, but what made the brain was evolution. Evolution didn't just make the brain, it made literally all life on Earth. So evolution, not backprop, is the master algorithm.

The evolutionaries simulate evolution on the computer except that instead of evolving animals and plants, they evolve programs. The person who first ran with this was John Hall, and he died actually just last summer. For a long time, when he started out in the late '50s, early '60s, people used to joke that the school of evolutionary computing consisted of just John and his students and their students. But then in the '80s, things took off and a lot of different people started doing evolutionary computing in all sorts of different areas.

And then John Koza actually developed this version of it called genetic programming that we're going to meet shortly. And then, Hod Lipson is another person doing very interesting things with this type of learning today that we will also look at.

This is the basic idea of this type of learning, which John Hall called genetic carbons, because they're algorithms

that imitate what genetics does.

At any given time you have a population of individuals, and each of these individuals is defined by a genome. Except that in our case, the genome isn't going to be the DNA base pairs, it's just going to be bits. In some sense, DNA bits are the DNA of computers, so they're just going to be bit strengths.

So each of these bit strengths defines a program, and then that program goes out into the world and does its thing. It tries to perform whatever task we want it to perform, and it gets a fitness score. Again, this is very close to biology.

The programs that do better get a higher fitness score than the ones that don't do so well. And then, the fittest individuals actually get to produce the next generation. You literally do crossover between the genome of a father program and the mother program and you get genomes of their children.

And then, on top of that, you do random mutations again just like in real evolution, and you have a new population. The amazing thing is that you can start this process with a population of random individuals and after, say, a few thousand generations, they're actually doing very useful and very non-obvious things.

For example, people in this area have been able to evolve things like radios and amplifiers starting literally from piles of components. And along the way, they've actually amassed a lot of patents. They've gotten patents for devices that were invented by genetic algorithms. They have, for example, amplifiers, low pass filters that work better than the ones that were designed by human engineers.

But John Koza's idea was that representing programs as bit strings is too low level. When I do crossover, I pick a random point and then I use one genome up to that point and then the other genome after that point. This is how nature does things. But it's very messy. It's very likely that I have something that is already a pretty good program and then I cut it at a random place and it doesn't do anything useful anymore.

So John Koza's idea was a program. At the end of the day, we're trying to evolve programs and a program is really a tree of subroutine calls all the way down to simple things like doing additions and multiplications and and's and or's. So he invented genetic programming to actually use the program tree itself as the genome.

Here is a very simple tree of operations. At the root, there is the multiplication of C by the square root of something else, and let's say I have picked the highlighted note from my crossover. One of the child trees is going to be the tree with the white notes.

That tree is actually one of Kepler's laws. It's the law that gives the average duration of a planet's year as a function of its average distance from the Sun. It's actually going to be proportional to the square root of the cube of the distance.

A genetic algorithm can variously induce this from something like Tycho Brahe's data that Kepler used, but it can also induce much, much more complex things. It can induce whole robot routines, and it can induce programs that do very nontrivial tasks.

And in fact, these days, the evolutionaries are doing things like evolving not just programs but real hardware robots. This little spider here is actually a mechanical spider from Hod Lipson's lab that was evolved.

What happens is that the robots start out as random piles of components in simulation. Once they're doing well enough, they get 3D-printed and they start to walk and crawl in the real world. There are spiders, there are dragonflies that fly, there are things that look like nothing that you ever saw before, but they actually crawl and walk and recover from injury and so forth. And in each generation, the fittest robots get to program the 3D printer to produce the next generation of robots.

So this is exciting and maybe also a little scary, right? If the *Terminator* comes to pass, [Laughter] maybe this is how it's going to happen. Of course, these little spiders are not ready to take over the world, but they've come a long way from the random pile of parts that they started out as.

Now, most machine learning researchers actually don't believe that imitating biology is the way to get to the master algorithm. So the evolutionaries emulate evolution, the connectionists emulate the brain. But most machine learning researchers have the attitude that, well, nature did things that way, who knows why and who knows how good it really is. Let's just try to figure out from first principles how we can learn optimally. Bayesians are very much in this paradigm of figuring out what the optimal way to learn is, and as I said Bayesians have a long history in statistics but also in machine learning.

More recently perhaps, the most famous Bayesian is Judea Pearl, who won the Turing Award, the Nobel Prize of Computer Science in 2011, for inventing something called Bayesian networks. This is a very powerful type of Bayesian model that is used for many different things now. Two other prominent Bayesians are David Heckerman and Mike Jordan.

Bayesians take their names from Bayes' theorem. Bayesian learning is all based on Bayes' theorem, and in fact, Bayesians love Bayes' theorem so much that there is a Bayesian machine learning startup that actually had a neon sign of Bayes' theorem made and hung outside its offices for the whole city to see. So they really, really believe in Bayes' theorem.

Bayesians are actually known in the machine learning community as being the most fanatical of the five tribes. They really have enormous religious attachment to their paradigm and they're the first ones to say so.

They have to be because for 200 years in statistics, they were a persecuted minority. Statistics was dominated by Frequentism, and the Bayesians had to get very hardcore in order to survive, and it's a good thing they did because they have a lot to contribute. These days with computers and better algorithms, they're actually in the ascendant even within statistics.

So what is Bayes' theorem and Bayesian learning all about? The idea is that Bayesians, above all, are concerned with the problem of uncertainty. They are obsessed with the fact that nothing I know do I ever know for sure. Anything that isn't used from data, I can never be completely sure is right.

So we need to quantify the uncertainty in the way we quantify probability. Then, we have a spate of hypotheses that we're considering and as we see evidence, we're going to update the probabilities of each hypothesis.

Roughly speaking, the hypotheses that are consistent with the data will become more likely, the hypotheses that are inconsistent with it will become less likely and eventually, there will be a winner. But there may not be a single winner, and then you just have to average the hypotheses weighted by the confidence that you have in them.

Bayes' theorem is really just the little piece of math that tells you how to do this. It's actually so simple that it's barely worth being called a theorem except for the fact that it's so important. What it does is to help compute the posterior probability of each hypothesis, which is how much I believe in that hypothesis after seeing the evidence.

But I start with my prior probability, which is how much I believe in each hypothesis before I even see any evidence. And this is what makes Bayesianism very controversial. Most statisticians, in fact, most scientists will say, well, you have no basis to make up these prior definitions. You're just pretending that you've quantified something that you know nothing about.

The Bayesian answer to that, however, is that you have to make those assumptions one way or another. You

can make them implicitly or you can make them explicitly, and we at least are going to make them explicit, and that's healthy.

So you start with the prior, and then as the evidence starts coming in, the question that you ask is if my hypothesis is true, then how likely am I to see this? And if my hypothesis makes the evidence likely, then conversely, the evidence makes the evidence likely. My model is likely if it makes the world that I am seeing likely. This quantity, the probability to the hypothesis of the data given the hypothesis, is called the likelihood and, it's also what frequentist statisticians use and what we all learn in Stats 101.

And when you do the product of the two, the prior and the likelihood, you get the posterior probability. There is a normalization constant to make sure that everything adds up to one, but it's not too important for our purposes. You can do all sorts of amazing things with Bayesian learning, one of which is driving cars. Your first self-driving car is probably going to have a vision network inside it. Google uses a massive vision network with hundreds of millions of connections to decide which apps to show you.

One application of Bayesian learning that we are all familiar with is spam filters. In a spam filter, the two hypotheses are: this email is spam or this email is not spam. You start out with a prior probability that, let's say, 90 percent of emails are spam.

Then, the evidence is the contents of the email. So for example, if the email contains the word "free" in all capitals, that makes it more likely to be spam. If it contains the word "Viagra," that makes it even more likely to be spam. [Laughter] And if it contains "free Viagra" with four exclamation marks, then it's almost certain to be spam.

[Laughter]

On the other hand, if it contains the name of your best friend on the signature line, that makes it a lot less likely to be spam. And so finally, after looking at the evidence, you get a probability that the email is spam or isn't, and then you use some threshold of probability to decide whether to throw out the email or put it in the user's inbox.

David Heckerman had the idea of doing this many years ago, and this was literally a grad student's summer project when he interned at Microsoft Research. These days, people use all sorts of different machine learning algorithms for spam filtering, but vision learning is still one of the most widely used methods and one of the best.

Finally, we have the analogizers, whose idea is that all learning is analogy. The way we learn, the reason we're able to do the right thing in new situations is that we notice there are similarities to our previous experiences, and then, based on what we did or what worked in those experiences, we figure out what to do in this new case. The analogizers are a less cohesive tribe than the other five. They're really just a bunch of different people that all do learning based on this idea. But this is a very central idea in machine learning.

The most important analogizer is probably Vladimir Vapnik. He invented support vector machines, also known as kernel machines, which until the height of deep learning, were the dominant machine learning algorithms. Even today support vector machines are still the best method for tackling a lot of problems, not deep learning.

Peter Hart was one of the people who started the very earliest form of analogy-based learning, called the nearest neighbor algorithm. We're going to see what that algorithm is shortly. And then, there are famous analogizers like Douglas Hofstadter, the author of *Gödel, Escher, Bach*. He actually coined the term analogizer.

He says he's an analogizer, and Einstein was an analogizer, and that all these great discoveries and things all happened through reasoning by analogy. So he very much believes that analogy is the master algorithm. In fact, his most recent book is 500 pages arguing that all of learning, all of intelligence, is just analogy and nothing else.

Interestingly, *Gödel, Escher, Bach* is really a book that has more to do with the symbolist type of machine learning, with logic and whatnot. But the entire book is an extended analogy between Gödel's theorem, the music of Bach, and the art of Escher. So the whole learning by analogy was already latent in his thinking, even back then.

So how does learning by analogy work? Let me explain it by way of proposing a simple puzzle to you. I'm going to give you a map of two countries, and I'm going to fancifully call them Posistan and Negaland because one is going to have the positive examples and the other is going to have the negative examples.

So for example, when we're learning to recognize cats, we call pictures of the cat the positive examples and pictures of dogs and everything else negative examples. And what I'm going to tell you is where the main cities in Posistan are on the map. So here's a Posistan city, here's another one, there is the capital, Positville. And the same thing for the main cities in Negaland.

Another question that I'm going to ask you is where is the border between these two countries? I just told you where the main cities are, and of course, you can't know for sure where the boarder is going to be because the cities don't determine the border.

But if I give you a piece of paper with these things marked on it, you can probably roughly put down where the frontier should be. And the nearest neighbor algorithm is really just using the following idea to do this: I am going to assume that a point on the map is part of Posistan if it's closer to a city in Posistan than to any city in Negaland.

So I'm going to break up the map into the neighborhood of each city. The neighborhood of a city is the points that are closer to it than to any other. For example, the neighborhood of this example, here is this area. And then, the region of the positive class is just going to be the union of the neighborhood with the positive cities.

Even though it's a really simple algorithm, notice that at learning time the nearest neighbor algorithm consists of doing exactly nothing. You do no work. Sometimes this is also known by the name lazy learning. It's like, oh, I'm lazy, I'm not going to study for the exam, and then, when I see the questions, I'll make something up. Like, your Mom told you that procrastination is bad, but actually, in machine learning it can be very powerful. It can be very powerful because this frontier that is only implicitly being formed can actually get very, very intricate.

In fact, what Peter Hart did was prove back in the '60s that you can learn any function in the world just by using the nearest neighbor. To be more precise, if you give this enough data, it can learn absolutely anything.

Now, the nearest neighbor has a couple of shortcomings, one of which is that if you look at it, this frontier is kind of jagged. The real frontier is probably smoother than that.

The other is that if you think about it, I'm actually wasting a lot of time and space here by remembering cities that I don't need to. For example, if I took out this city and, in fact, if I took out positive itself, if I just erased them from the map, nothing would change.

The reason nothing would change is that this neighborhood would just get absorbed by the neighborhoods of the nearby cities and the frontier itself would not change. The only thing that I really need to remember are the examples that keep the frontier as it is, where it is. For example, if I took this out, then the frontier would move.

Those examples are called the support vectors—vectors because examples in machine learning are usually represented as vectors, and support vectors because they're supporting the frontier.

Vladimir Vapnik invented support vector machines, which in essence solve both of these problems. They figure out exactly which examples you need to keep, and they also learn a smoother frontier. The frontier can come

from much more general classes of curves than just a piecewise straight line.

The way support vector machines do this is also quite intuitive. Suppose I told you to start on the south end of the map, and you have to walk all the way to the north, always keeping the positive cities on your left and always keeping the negative cities on your right.

We know how to do this. We start walking, we go all the way up there. But there is a twist. You have to give all the cities the widest possible berth. Imagine that the cities were mines and this whole thing was a landmine. You wouldn't just walk anywhere. You would stay as far away from the mines as you could while still doing the job.

You would try to maximize your margin of safety, and this is, in fact, how support vector machines work. They try to maximize the margin between the frontier and the examples of each class. By avoiding going close to here, I actually avoid going into a region that perhaps actually is negative even though I am not sure.

Analogy-based learning has been used for all sorts of things. It's one of the oldest and best established types of learning. But one that we are all familiar with is recommender systems. Netflix, for example, needs to decide which movies to recommend to you.

In the early days, people tried to recommend things based on the properties of the audience. It says, well, you like action movies but you don't like movies with Arnold Schwarzenegger but you like movies with this director, etc. This turned out to not work very well because taste is subtle. Whether or not you will like a movie is not a simple function of its properties. Using other people as a resource works much better.

What I need to do to recommend a movie to you is look for people who have similar tastes. And I know that their tastes are similar to yours because you've given similar ratings to movies. If there is someone that gave five stars when I gave five stars, gave one star when I gave one star, and gave 5 stars to a new movie that I haven't seen, the system hypothesizes that I'm going to like it as well. This is really just an application of the nearest neighbor idea in this particular domain, and it works shockingly well.

Three-quarters of the movies that people watch on Netflix come out of the recommender system. This is how important it is to their business. And Amazon, of course, also has a recommender system that you've all met. A third of what Amazon sells comes out of the recommender system.

This makes a huge difference to their bottom line and, in particular, how accurate this is makes a huge difference. Every e-commerce site worth its salt has one of these systems. These days people use all kinds of different algorithms to do this, but the earliest one was this type of similarity-based learning, and it's still one of the best.

Let's take a step back now. We've met the five main tribes of machine learning. We've seen that each one of them has a problem that it can solve better than the others, and each one has an algorithm it uses to solve that problem.

For symbolists, the problem that they really care about is learning knowledge that they can then compose in different ways. They can be very flexible that way, and they discover that knowledge through inverse deduction by filling in the gaps in deductive reasoning.

Connectionists emulate the brain, and the problem that they solve is called the credit assignment problem. It probably should be called the blame assignment problem because it's deciding who needs to change when something goes wrong. Their algorithm for doing that is backprop.

The evolutionaries discover structure. The connectionists have to start with a predefined architecture and then change the weights, but the evolutionaries actually know how to evolve that structure in the first place. The most

sophisticated way they do this is via genetic programming.

Then there are the Bayesians, who care about the problem of uncertainty. Their answer to that problem is probabilistic inference.

And then finally, the analogizers can reason by similarity. As a result, they can function in all sorts of situations where the others would completely fail. For example, if I just have one positive and one negative example, the other methods don't know what to do, but with nearest neighbor, you just put a straight line between them, which is quite sensible.

They can also generalize farther. For example, Niels Bohr's original theory of quantum mechanics was based on an analogy between the atom and the solar system, where the nucleus was the sun and the planets were the electrons, and so on. So with analogy, you can actually generalize much farther than the other methods can. And now, each of these tribes very much believes in its own master algorithm.

For example, these days, deep learning is really going like gangbusters and some of the connectionists think backprop is all they're going to ever need. But I think the truth is that precisely because each of these problems is a real problem, none of the tribes has the whole answer. The whole answer is one algorithm that actually solves all five. That's when we will truly have a master algorithm.

We need a grand unified theory of machine learning in the same sense that the Standard Model is a grand unified theory of physics because it unifies the different forces, or the central dogma is a grand unified theory of biology, and so on.

So what might that look like? I know a lot of us have been doing research on this for a while and we eventually made a lot of progress and are getting fairly close. The learning algorithms that I describe all look very different, so it seems very unclear how you could possibly unify them. In fact, some people have argued that it's impossible.

But it becomes a lot easier once you notice that all learning algorithms are really composed of the same three parts. And so, all we have to do is unify each of these parts in turn.

The first part is representation. It's the choice of language in which you are going to write the program that you learn. Now, human programmers will use languages like Java and Perl and whatnot. Typically machine learning people use more abstract languages like, for example, first order logic, but the principle is the same. It could be differential equations if you're modeling a physical system where you need to choose that. So that is the choice of representation.

A natural thing to do here is unify first order logic, which the symbolists already use. But we need to unify that with what the Bayesians do because we need to handle probability, which logic doesn't.

We have done that, so now we have this probabilistic logic. For example, the best known one is called a Markov logic network, which combines first order logic with graphical models. Examples of this are vision networks and Markov networks.

This language is essentially the same formulas as in first order logic, and a formula is just like a sentence in English. In fact, sometimes people call them sentences except it's stated more formally in a way that the computer can handle. But the twist is that we are now going to attach a weight to each formula. A formula that has a high weight is a formula that you really believe in, so if the world violates that formula, then its probability really takes a hit.

So you have all these formulas with all their weights, and the more formulas the world satisfies and the higher

weight they have, the more likely the world is. And with this, we can represent pretty much anything that we might want to represent in any field.

Now, the second part is evaluation. We need a score function to tell us how good a candidate program is. One of the score functions to use is posterior probability, which you already heard about, but more generally, the evaluation function actually should not be a part of the algorithm. It should come from the user. It's you, the user, who should tell the learning algorithm what it's supposed to be optimizing. So if you're a company, the evaluation function might be return on investment.

If you are a consumer, it might be some measure of your happiness. It's for you to say. And then, finally, what the algorithm has to do is optimize. Finding the program or the model in that big space defined by the language achieves the maximum score.

And now here, there's a very natural combination of ideas from the evolutionaries and from the connectionists. We need to discover the formulas, but a formula is just a tree of sub-formulas with conjunctions and disjunctions. We can use genetic programming to evolve our formulas.

Then, we can use backpropagation to learn the weights within the formulas. I have my big chain of reasoning that I use to explain the data, and with weights within the formulas in various places, I can just backprop through that to optimize my weights.

So we are actually pretty close to having a complete unification of the five paradigms. And some people say that or believe that that's all we're going to need. My sense, my intuition is that actually that's not the case. It's that even after we have successfully unified these five paradigms, there will still be key new ideas that somebody has to come up with.

There are some insights that we haven't had, and in some ways, someone who is not a machine learning researcher is better placed to have those insights than we in the field. Machine learning researchers are already thinking along the tracks of a particular paradigm, which makes it hard for them to see outside that paradigm.

One of my secret motivations in writing my book was to get other people interested in the problem because maybe they'll have those ideas that we're not having. So if you figure out how to do this, let me know so I can publish it.

Let me conclude by just mentioning some of the things that I think will be possible with the master algorithm that are not possible today. The first one is home robots. We would all like to have robots that do the dishes and do the cooking and make the beds and maybe even look after the children. Why don't we have them today?

Well, first of all, everyone agrees that you can't build a home robot without machine learning. We don't know how to program even a car to drive itself let alone a home robot. The second problem is that a home robot, in the course of an ordinary day, runs into every single one of those five problems, multiple times, which means that no single one of the five master algorithms is enough. If we unify them, then hopefully we will have what we need.

Here's another one: all of the major tech companies have a project to turn the worldwide web into a knowledge base that computers can understand and reason with. Google has the Knowledge Graph, Microsoft has Satori, et cetera. The idea is that I do not want to just type in keywords and get back pages. What I want to do is ask questions and get answers.

But for the computer to be able to do that, it has to understand the text that's out there on the web. It needs to transform that text into something like first order logic. On the other hand, the knowledge on the web is messy. It's ambiguous, contradictory, broken, and incomplete. It's going to be full of uncertainty, so you need the

probability as well. And so again at the end of the day, we're going to need to unify those five paradigms in order to be able to do this.

A very important problem that we hope computers will be able to solve one day is curing cancer. In a way, diagnosing disease is a perfect application for machine learning. And indeed, for most diseases, learning algorithms already do this better than doctors. In a way, what you need to do is recommend a drug for the patient in the same way that you might recommend a movie or a book except that the problem is much, much harder.

The reason we haven't cured cancer is that cancer isn't one disease. Everybody's cancer is different and the same cancer mutates as it goes along, so it's very unlikely that there will ever be a single drug that cures cancer.

What we really need is a program that takes in the patient's genome, the tumor's mutations, the patient's medical history, and other relevant information, and then suggests a drug for that particular cancer. Or maybe a combination of drugs. Or even designs a new drug.

There are already companies that are starting to do this and projects to pull together patient data and whatnot because without that data of the tumors, the drugs, and the outcomes, we can't do this. But at the end of the day, it's going to take modeling how living cells work, modeling how the gene regulation happens because it's when that goes awry that you get cancer.

And so, that's going to require a lot of data like microarray data and gene sequencing. But again, it's also going to require more powerful learning algorithms than the ones that we have today because all of those five problems keep turning up here.

Let me return once more to recommender systems. Today every company has its own recommender system. That model is just a little sliver of you based on the data that they have. So Netflix has a model of your movie tastes based on your movie ratings. Amazon has a model of what you buy based on what you did on their website. Facebook has a model of you to choose which updates to show you, and Twitter has one for tweets, and so on.

But this is not what I as a consumer really want to have. What I want to have is a single complete 360-degree model of me learning from all the data that I ever generate, and then to have this model help me with the decisions that I have to make at every stage of my life. Not just picking movies or books, but finding jobs, deciding where to go to college, finding a house, even finding a mate.

Most marriages in the world today, or I should say in America today, start online, and the matchmakers are learning algorithms, picking a potential list for people based on their profiles. So there are actually children alive today who wouldn't have been born if not for machine learning.

But the quality of the process is still very low because you can't predict whether two people are going to be a good match just based on their profiles. The more of their life you can see, the better you know them, the better you'll be able to do this.

We're going to need to have that data pulled together, and there are a lot of interesting issues such as who will I trust to do that? Will I trust one of these companies? What are the privacy considerations, and so forth?

Companies are all in a race trying to do this. Google has Google Now and Microsoft has Cortana. You see all of these things coming out, but even if you have all that data, the learning algorithms that we have today would actually not allow us to do this. The master algorithm would. Let me conclude there and take questions.

Question: In the field of empirical finance and historically, the concept is you have a hypothesis and then you

test it using the data. Some of us historically have used the outputs of that to help make investment decisions. There is sort of an egregious stand, if you will, that people who are thoughtful users talk about which is called data mining.

Data mining is derogatory in empirical finance. Among a number of things, it's overfitting models, it's reacting to outliers, it's looking at datasets that are not representative of the big picture, having nonstationarity that does process changes over time.

So not being as familiar with machine learning concepts and how that all works, how much of a problem is that in your world and what are the techniques used to prevent egregious errors based on overfitting?

Pedro: Yes, overfitting is the central problem in machine learning. Every time that you have a powerful type of learning like all the ones that I have described, this becomes a huge danger. You hallucinate patterns where there aren't any. And so you need to figure out ways to avoid hallucinating those patterns. And every one of these paradigms has their different ways of doing this.

The simplest way is to make sure that you don't believe your model until it makes correct predictions on data that it wasn't trained on. This is a really simple rule and breaking it is the number one mistake made by people who do machine learning. If you do that, your results will be crap because you can just memorize.

Think of the simple nearest neighbor algorithm. You can memorize the data and it's always perfectly accurate on the past, but the problem is when a new patient comes along, or a new situation, will you do the right thing there?

And so, one way to beat this is with a type of holdout testing or sometimes cross-validation as it's called, depending on how you do this. Every learning algorithm usually has at least one parameter that you can tweak to make a tradeoff between overfitting and being blind to patterns that are there.

You want to be able to see the patterns that are there without hallucinating patterns that aren't. In the case of nearest neighbor, there's a very simple parameter in the number of neighbors that you use. In general, people don't just use the single nearest neighbor; they use the k-Nearest Neighbors. For example, I find the k-Nearest Neighbors to that patient and they vote, and the majority wins. If I increase k, I become less likely to overfit. At the limit, if I use everybody, then I just predict the most frequent diagnosis.

Question: What is an optimal sample size for something like the nearest neighbor case?

Pedro: Well, it depends. We are in the days of big data. In fact, there are many domains where you have so much r-sample data that you don't even use all of it. There are, however, problems where you don't have a lot of r-sample data and, in particular, the most difficult problems are the ones where the phenomenon is continually changing. So if you gather a lot of data, you are outdated, but if you don't get a lot of data, you don't learn the phenomenon.

Those problems are hard and at the limit maybe you can't solve them, but one way you solve them is by working at multiple time scales and bringing in other knowledge. You really try to find what the constants are.

A lot of things are changing but some aren't. And in particular, in cellular biology, the way the cell works actually doesn't change. The particular cancer that you're seeing is new, but we know how to cope with that very well in machine learning.

Question: I think a lot of the people in this room probably have seen *The Matrix*, *The Terminator*, *Ex Machina*, etc. When computers reach singularity, this risk is real. What ethical guardrails are taking shape in your world?

Pedro: Yes, so it is very controversial whether computers will reach the singularity or not. Most computer scientists do not believe that that will ever happen, but singularity means you have a learning algorithm that can learn . . . imagine a learning algorithm that makes another learning algorithm. If it makes a learning algorithm that is better than itself, and that one makes a better one, then we have a runaway intelligence.

This is actually what the concept of the singularity is.

Now, even if this happens, it doesn't mean that each one is going to get better than the previous one. In fact, the singularities that you imagine get this exponential growth, can't actually go to infinity. Like all these technology curves that look like exponential curves, at some point, you start to see diminishing returns. They start out growing faster and faster, but then they taper off, and it's going to be the same thing with intelligence.

Question: Do you see AI advancing to a point where it becomes a danger?

Pedro: No, exactly but let's suppose that we will have very intelligent machines. Should we worry about them? Right now, they are a problem to people like Elon Musk and Stephen Hawking who said AI is an existential danger to humanity, which has gotten a lot of press and whatnot.

I don't know anybody who is actually an AI expert that takes these ideas seriously. The reason they don't take them seriously is that AI, no matter how smart, is still just an extension of us. People have this image of AIs as being agents that are going to have different goals from us and compete with us or wipe us out, but why would that happen if we are the ones designing them?

The thing about *Terminator* and *Ex Machina* and all these Hollywood movies is that in them, the AIs and the robots are always humans in disguise, because that's how you make an interesting movie.

But real AIs, real machine learning algorithms look nothing like humans in disguise. In particular, their goals are set by us. As long as their goals are set by us, the more intelligence they have, the better. Because if their set goal is to cure cancer, you want it to be as intelligent and as powerful as possible.

In fact, what we really need to be afraid of is not computers getting too smart but rather them being too stupid. People worry that computers will get too smart and take over the world, but the real problem is that they're too stupid and they've already taken over the world.

Computers are already making all these decisions about you and me, like who gets credit and who gets flagged as a potential terrorist. They're very fallible because they just lean on datasets and their programming.

So the dangers from AIs come from the EII (Enterprise Information Integration) not knowing enough to do the right thing: from it not having common sense and misinterpreting what you say. Look what happened to King Midas, who wanted everything he touched to turn into gold.

The way to avoid these things is to make the computers more intelligent, not less. So this idea that we should limit the intelligence of computers in order to be safe is exactly backwards. Computers are already flying airplanes, and soon they'll be driving cars. The way to be safe is to make them more intelligent, not less.

Question: So maybe a similar question asked differently, what parts of human experience are not reachable with the five that we have right now?

Pedro: That's a good question and let me preface my answer with the following. Ten years from now, there will be a lot more AI in the world than there is now. But the vast majority of that AI will look nothing like people, and we will be completely unaware of it. It will just be doing its job in some corner of the economy.

A small percentage may look like people because, for example, if I have a home robot, one that's very cute and

cuddly and humanlike is likely to sell better than one that's just a big hunk of metal. So these algorithms by themselves are not that close to people, but you can imagine some generations forward having algorithms that for their purposes do the things that people do.

Here the question would be, does that algorithm really have consciousness? This is actually something that people already ask about some of these bots. Because they look like they're conscious, people treat them like they're human.

And I think ultimately, we're never really going to know that answer. How do you know that I am conscious? You just assume that I am conscious because we're similar enough that since you're conscious, maybe I am, right? You're just doing analogical reasoning.

And people have this amazing ability to project human qualities onto things that behave even slightly humanly, so the same thing will happen with intelligent robots. We will treat them as being conscious, whether or not they are. I think it's an interesting question whether they truly are conscious, but I think that we will never know the answer to that question.

Question: You started off talking about the analogues between neuroscience and a lot of what we learned in neuro were from anomalous incidents in history. You know, a guy getting a pole stuck through his brain, and so on, and then someone discovers something.

There are a lot of positive anecdotes with AI and machine learning. To that, I think people are familiar with, the Target ad that predicted a teenage girl was pregnant from her buying patterns.

There's a startup now that is doing visual recognition, identifying images that we would mistake for pornography that, in fact, are not. Are there examples where the computer is getting the opposite, where they get it completely wrong?

Pedro: Oh, yes, there are examples galore.

Question: What are we learning from that?

Pedro: This question is closely related to the previous one although it doesn't seem to be. When we see one of these learning algorithms do something like, for example, recognize objects, we assume that it's doing it the same way that we do.

So for example, a computer learns to tell cats from dogs, and we assume that it kind of knows what a cat or a dog is, but in reality, it doesn't. It just picked up on some signal that allows it to distinguish cats from dogs.

One of my colleagues actually did the following experiment. He trained a neural network, one of these deep models, to discriminate dogs from wolves, and it was 90-something percent accurate, so he was doing a very good job.

But then they had this way of looking inside the network to try and figure out what it has learned, what parts of the image that it was keying in on.

Now, what do you think it was keying on? Maybe the snout, maybe the ears? What it was focusing on was long horizontal white patches in the image. Why was that? Because the images of the wolves were in snow for the most part, and the images of the dogs were not. So what it really learned was how to classify snow. [Laughter] And machine learning algorithms have an uncanny tendency to do stuff like this.

Now, one way to look at this is to say, oh my god, the algorithm is so dumb. But a lot of animals in the real world make mistakes like this, too. If you take a cow and you put a square of calfskin rubbing against its side, the cow

leaves its calf behind. Shocking, right? It's that easy to fool a cow, and I know there are more examples like this.

But I would look at this as evolutionarily, nobody was playing pranks like that on the cows, [Laughter]. Cognition has a cost. Cows use their energy in their intestine, whereas we use it in our brains. But even our brains are extremely energy-intensive, so using the least amount of energy that will get the job done is fine.

Now if you will always meet wolves in snow and dogs not in snow, then you're fine. The problem, of course, is that one day you see a wolf in your backyard and maybe this will get you killed.

Question: Thanks. We heard in Bill Gurley's talk before you the idea that all of the technological advancement in AI and so forth might cause disruption for individuals, but be a big positive that creates growth at the level of society. He cited the idea that 100 years ago, 99 percent of people worked in agriculture, and today, one percent do. People found jobs doing other things that are better.

Do you feel similarly or are you more concerned that this Fourth Industrial Revolution is a different kind of technological change that may leave fewer roles for human workers if we can recreate a lot of what humans do?

Pedro: This is actually a raging debate right now. There are the people who say that this is just the next stage of automation. We've seen this story before in every culture, and there is nothing to worry about. It's always easier to see the jobs that disappear than the ones that appear.

The biggest effect of automation is that things become cheaper. People are now going to buy other things with the same money. Or, things that were completely impossible become possible, and you have new jobs.

For example, there are millions of app designers in the world today. That job didn't exist ten years ago. Farmers in the 19th century worrying about the destruction of farmers' jobs couldn't possibly have imagined that one day there would be app designers and other things like that, and all of us in this room for that matter.

But now, the counter to that is people who worry that this time is different. And the argument usually runs that the Industrial Revolution automated manual work, but now we are automating intellectual work, and then there will be nothing left for the people.

Now, where do I fall on this? I think we really need to distinguish the short to medium term from the long term. In the short term, and by short term I mean the next decade, I don't buy the idea that this time is different, because AI is a very long road.

We've come a thousand miles but there are a million more to go, and they're thinking about this time as going the other million miles. That million miles will not happen in the near future. In the distant future, maybe we will have computers and robots that do everything better than people.

I think what's going to happen in that case is that we're all going to be independently wealthy. And then, people are already talking about these things like a universal basic income. But politically right now, it's impossible to do something like that because the great majority of us are producing and not receiving unemployment benefits.

But if you get to the point where more than half of the people are unemployed, and the democracy is in place, I don't see how people will not vote themselves very generous unemployment benefits and call it lifetime income. They would have all sorts of very good moral justifications for that stuff. I think that will happen.

In the short term, I think what's going to happen is there is going to be a lot of displacement. A lot of jobs will disappear, and this is a real concern. For example, if I was a truck driver, I would be really looking at getting another job right now because sometime within the next several years — and truck driver is the most frequent occupation in the U.S. — a lot of these jobs are going to disappear. So we need to worry about what happens

with the people who lose those jobs. It's not that we have to retrain those people to be data scientists. Again, the interesting thing that is happening today is that on the one hand, there are people who don't have jobs, but on the other hand, there are all these companies who are desperately short of qualified people.

So we need to train these people to be qualified for those jobs. Bill was saying we need more computer scientists and more data scientists, but also, I think we need to empower people to retrain themselves and to find the next job that they're going to do. The unemployed truck driver isn't going to become a data scientist.

Say the cost of trucking goes down, the costs of transportation go down, the costs of goods goes down, so a lot of people have more money. Maybe now they buy better houses so there are more jobs for construction workers. So maybe the truck driver becomes a construction worker, and construction work is very difficult to automate.

One of the lessons that we've learned in AI painfully and that everybody should be aware of is that we used to think 30 years ago that the easiest jobs to automate were going to be the blue collar ones. We thought that the white collar jobs, which require education, were going to be hard to automate.

That has it exactly backwards. The hardest jobs to automate are things like construction work because they require dexterity, moving around, not stumbling, seeing things that we take completely for granted. These are tasks that took hundreds of millions of years to evolve.

On the other hand, doctors, lawyers, analysts, engineers, scientists, these are hard because we didn't evolve to do them. We have to go to college to learn how to do them. But that also means that the computers can learn that much more than we can because we are evolving them for that purpose. So there are already a lot of white collar jobs that have shrunk or disappeared and more of them will.

I think what's going to happen in this ten-year future is that some jobs will disappear, but the majority of jobs will not. They're just going to change.

The way I do my job will change, and what I need to think about in my job is: what in my job can be automated by machine learning algorithms? Could a machine learn to do what I do by observing me? If it is the case that everything I do can be automated, then I need to get another job quickly.

But what will be the case for most people and in particular, most white collar jobs, is actually much more interesting. It's that some parts of my job can be automated but others can't.

So what I want to do is I want to automate my job. The way to keep my job safe is to automate it myself, and then I spend my time doing the higher value things that only I can do on top of what the computer has now automated. People often frame this in terms of a race between humans and machines, but the real race is between a human with a machine and a human without a machine.

A very good example of this is when Deep Blue beat Garry Kasparov. People figured computers were now the world chess champions, end of story. But actually the best chess players in the world today are not computers because for the time being, humans and machines have different strengths.

What you want to figure out is how do I use the computer to do my job better than I alone could do it or better than the computer could do it? And I think for most occupations, this is what people should be preoccupied with.

Michael: We're going to break for lunch, but I do want to ask one thing. Two years ago in this conference the theme was prediction. And a number of our speakers felt that computers beating humans at Go would take five to ten years. But in just the last couple months, we saw AlphaGo. Can you explain how AlphaGo works? Were you surprised by how quickly they were able to do what they did in the game of Go?

Pedro: Yes, that's a great question. So indeed, if you had asked me a year ago how long it would take for computers to beat humans at Go, I would have said I wasn't sure. I would have thought probably a long time. So what did DeepMind do that was so amazing?

Demis Hassabis is the guy who created DeepMind, and its initial success was in playing computer games, in playing Atari games, using these deep learning algorithms that we just saw. Now, what Demis decided to do at one point was ask why Go was so hard compared to chess or checkers.

Why was chess solved 30 years ago with no machine learning involved, just using search? The problem is that within Go, the space of possibilities is way larger. And the problem with picking a good Go move is almost more like a visual pattern recognition problem.

The way most game players work is they have these evaluation functions of the board. They say this is a good board position because I have a piece advantage, control of the center, etc. I can enumerate those things. And going back to the '50s, what people did was they tried these things and they put weights on them. This doesn't work for Go. You ask a Go expert why they made that move and they can't explain.

So Go is almost more of a pattern recognition problem. Pattern recognition is what deep learning is good at: vision, speech, and whatnot. So DeepMind took these neural networks and plugged them into a classic Go-playing search process, which again is a little different from the ones that people used before. There is a thing called Monte Carlo tree search that Go players use to go from being a complete disaster to being as good as a human amateur. That was already there, but the pattern recognition wasn't. When you combine the Monte Carlo tree search with the neural network's thought by recognition, you get what DeepMind produced.

The other interesting aspect of it is that DeepMind got to where it is by initially learning from a big database of all the Go games that it could find. That thing spent three months burning thousands of servers just playing against itself.

And in fact, this is one of the oldest ideas in the whole field: self-play. The first known occurrence, at least known to me, of the term machine learning is in a paper by a guy at IBM (International Business Machines) Research in the '50s called Arthur Samuel. He taught a computer to play checkers by playing against itself until it was as good as a human being.

At the time, Thomas J. Watson who was the President of IBM, said that when this paper is published IBM's stock will go up by 15 percent. And it actually did because people's perception of what a computer could do changed. This is what's happening with companies like Google today. It's a combination of very new ideas, semi-new ideas, and very old ideas that actually made this all happen, coupled with a lot of computing power.

Michael: I think on that, we'll break. Thank you very much, Pedro.



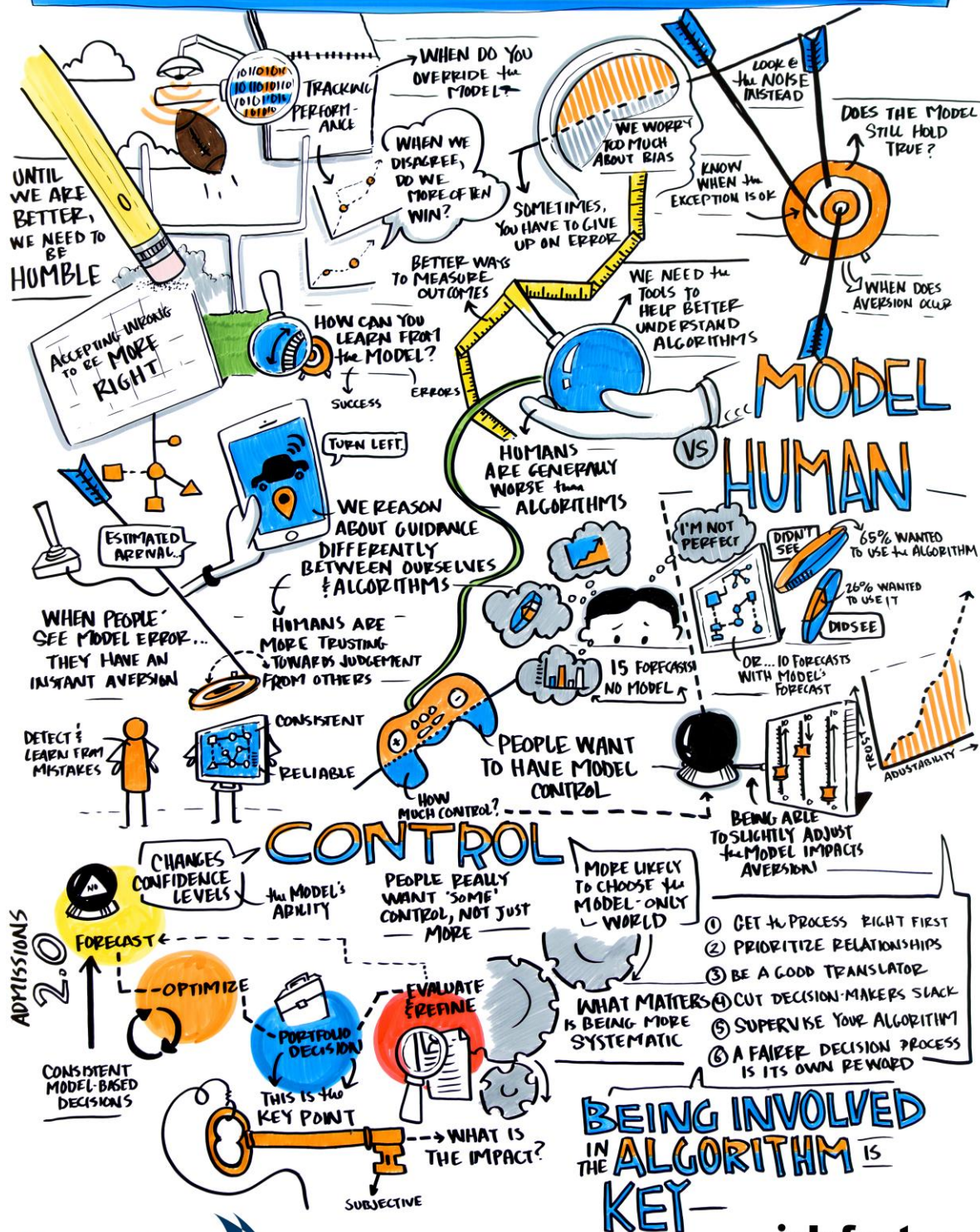
Cade Massey
University of Pennsylvania

Cade Massey is a Practice Professor in the Wharton School's Operations, Information and Decisions Department. He received his PhD from the University of Chicago and taught at Duke University and Yale University before moving to the University of Pennsylvania. Cade's research focuses on judgment under uncertainty—how, and how well, people predict what will happen in the future. His work draws on experimental and “real world” data such as employee stock options, 401(k) savings, the National Football League draft, and graduate school admissions. His research has led to long-time collaborations with Google, Merck, and multiple professional sports franchises.

Cade's research has been published in leading psychology and management journals and has been covered by *The New York Times*, *The Wall Street Journal*, *The Washington Post*, *The Economist*, *The Atlantic*, and National Public Radio. He has taught MBA and Executive MBA courses for 15 years, receiving teaching awards from Duke, Yale, and Penn for courses on negotiation, influence, organizational behavior, and human resources. Cade is faculty co-director of Wharton's People Analytics Initiative, co-host of “Wharton Moneyball” on SiriusXM Business Radio, and co-creator of the Massey-Peabody NFL Power Rankings for *The Wall Street Journal*.

ALGORITHM AVERSION: MASSEY

WHAT IT IS & HOW TO DEAL WITH IT



Michael Mauboussin: It's my pleasure to introduce our next speaker, Cade Massey. Cade is a Practice Professor at the University of Pennsylvania's Wharton School in the Operations, Information, and Decisions department. Cade's work focuses on judgment under uncertainty, including overconfidence, optimism, underreaction, and overreaction.

First of all, I have to say that I love talking to Cade. If you're interested in business, sports, or investing, his work provides countless lessons about how to think better and decide more effectively. What I love about Cade is that he straddles both the experimental world and the real world. He has also collaborated with some of the largest companies in the world, such as Google as well as numerous sports franchises.

In many ways, Cade stands at the intersection of all of the discussions today. You're going to hear about algorithm aversion, inefficiencies in sports markets, and even ways that we can introduce more rigor to certain processes that are largely subjective, such as graduate school admissions.

The last thing I'll mention is that Cade is very involved, as a faculty co-director, in Wharton's People Analytics Initiative. I've had the pleasure of participating in the annual conference over the last couple of years and the work is very exciting. Think "*Moneyball* meets HR (Human Resources)."

Please join me in welcoming Professor Cade Massey.

Cade Massey: Thank you, Michael. Appreciate it. Thank you. I'm delighted to be here. We've had our conference for three years. Michael has made huge contributions in each of the last two, and it is much appreciated.

So Michael, I took seriously your theme of what being wrong can teach us about being right. I'm going to open the discussion this afternoon with a little story about being wrong. I also decided to title it a little bit more poetically, and again consistent with your message: "Accepting Error to Make Less Error." I didn't coin that phrase. You'll see where that comes from momentarily.

But I want to start with a story about the Massey-Peabody Power Rankings. This is a collaboration with a former student of mine, Rufus Peabody, who is a professional sports gambler. We publish the football rankings, starting out just with the NFL (National Football League), and now spanning to college. We've done it I think for six years now in the *Wall Street Journal*.

When I was at Yale, the *Wall Street Journal* asked if I would put together a power-ranking system for them, and I said I would as long as I could get my former student to actually do this with me. They were up for it, and we've been publishing for the last three years.

These days, it's more easily found online. I'm going to show you various clips from this project. It's been kind of a garage project. This is just something we do on the side. Rufus is a full-time gambler. He uses these inputs a little bit. It's certainly consistent with what he does in his professional life.

But for me, it's been a way to learn a little bit more about judgment under uncertainty, and it also gives me a platform to talk about judgment under uncertainty, because people are generally more interested in hearing about football than the latest experiment that we've run on campus. Lucky for you, you're going to get both today.

The system is set up so that we rank teams from 1 to 32. These are NFL teams. And we quantify the ranking, so this is the number of points we'd expect them to win by or lose by to the average team on a neutral field. In this ranking, that's the Broncos at plus eight all the way down to the Jaguars at minus 8.39.

So if these guys were to play each other on a neutral field, we would favor the Broncos over the Jaguars by

something like 16.4 points. And we built it that way so that you could use it to bet in actual NFL games, if you wanted to do that. And then, his being a professional gambler and my being a professor, we actually track our performance.

We want to see how we're doing, and it has turned out that we've done well in this role. We kind of surprised ourselves in how well we've done. Each week, we designate which are our "Big Plays." Those are the ones we're most confident about. The other plays are still bets but they're not as confident. Then, just to add a little flavor, we throw a few extra games in there.

We track performance on these Big Plays. In 2014 or so, we had 10-and-a-half games right, five games wrong, and five-and-a-half games tied. But we track this as we go through the season. So we're always making sure we know where people are communicating and how we're doing.

And then, at the end of the year, we want to track how we've done. I'm building up credibility a little bit here, this is over the first four years. I ran this midseason in 2014 for another talk and the breakeven line in the betting markets is 52.37. That's 50 percent plus a little bit of the vig. You have to do better than that to make money.

Across all of our seasons, we've been above that breakeven line. We're calibrated in that our Big Plays do better than our Other Plays. I can tell you that we didn't end up 2014 this well. Regression to the mean hurts sports gamblers as well. This came down some. In 2015, we had a very good year. We were again profitable. We started doing college football a few years ago and we've done really well in that as well.

Let's do the calibration exercise that says here is the edge and here is our difference with the market—starting with perfect agreement with the market to increasing difference with the market. This is a three-point difference with the market. This goes out to six. The histogram here is just frequency of gains, so you see that we very often agree with the market and in fact generally stay pretty close to the market.

But then, what we want to know is, the more we disagree, do we more often win? And we see this increasing line, so we've got good calibration. This is exactly what you'd hope for. But this is just an aggregate, and of course there are exceptions. I told you this was a story about being wrong, so I'll tell you the wrong part of it.

A few years ago, Texas came into their big rivalry game against Oklahoma, and they were having a bad year at the University of Texas. The previous two years, Oklahoma had beaten them by an average score of 59-to-19, so Texas fans were hurting and not expecting much better. At the beginning of the year, the futures line on this game was a pick 'em, but five weeks into the year, it was a 14-point line.

So at this point, Oklahoma is expected to win by 14. Our model kicks out its predictions for the game. Rufus sends me the predictions, and we have a Big Play on Texas, which is fine. Four-and-a-half years into the model, we've never changed anything. Trouble is I'm a Texas fan. I've lived through those 59-to-19 games the last couple of years. I feel like I know more about the University of Texas football team than anybody else out there, and there is no way the Longhorns should be favored even getting 14 points in this game. The system thought that they would lose by 9 or 10 instead of 14. That's a Big Play in our world.

I thought Rufus was pulling my leg. I literally thought he was joking with me. I said, no way the model likes Texas and he says, absolutely, it's a Big Play. I said, well, we're not going to run it. [Laughter] And he's like, we've never overridden the model, ever. [Laughter] I said, we're going to do it this time.

And so, he said, okay, well, if we're going to do that, you're going to have to bet with me. He and I, we had just done some software work, and so we bet between us the bill for that software work just to make it interesting.

And he said, that's fine against the line, but what if Texas wins outright? And I'm like, well, whatever you want

because that's not happening. And I'm the person who studies overconfidence.

So the bet was that if Texas won outright, that we would be Peabody-Massey for the next week. [Laughter] None of this Massey-Peabody stuff. It would be Peabody-Massey for a week.

You might guess where this is going. The Longhorns win, they win big, it was a glorious day for Longhorn fans, and the next week, in all of our outlets—our website, our Twitter account, the *Wall Street Journal*—we went out as Peabody-Massey.

It remains the only time we've overridden the model. And I feel like this was a lesson learned, and I'm happy to have paid that small price to learn that lesson of not overriding the model. The wiser way to go is the title from this Hilly Einhorn paper, which is where I get the title for the talk today: "Accepting Error to Make Less Error."

Hilly was a researcher at the University of Chicago. He was one of the first behavioral researchers at the University of Chicago. Before Dick Thaler was there, before Josh Klayman was there, Hilly was there arguing with all of those new classical economists about the rationality of man.

And I didn't know Hilly, but I'm told that he was pugnacious and he was a good fighter. He has a ton of great papers, and he has this very short paper that you guys would all enjoy, "Accepting Error to Make Less Error." And the idea is this distinction between clinical judgment and actuarial judgment, and the virtue and supremacy of actuarial judgment.

But in there is this notion that you have to accept some error in order to have that superior system. In order to have a superior model, you have to give up on being perfect, essentially. So that's a nice idea and it's an elegant paper, but what does that look like in practice?

It is harder in practice. We've gotten a little better over the years, and we've had relatively outstanding performance in this world, but it's still a very noisy world.

So the trouble with Einhorn's idea of accepting error to make less errors is you have to accept every one of those deviations as being as good as you can do. And so, that algorithm is the prediction. You're not allowed to deviate from that prediction for any given game. You're not allowed to chase those errors that you think you could actually improve.

As you go through the season living that, Einhorn's idea is nice but it's tough. And I feel like I have learned that more from having worked on this Massey-Peabody project for the last six years than I could have ever running experiments or just talking about things in class.

I assume this is very close to your world. I think my sense is the people who get this kind of thing very well are sports gamblers and folks who are involved in financial markets because you've got some model, you've got some trading strategy, and you don't always know on any given day whether that model still holds. It's always this test of confidence of whether it holds as you see these deviations from the model.

Einhorn is going to strongly come in, and everything I'm going to talk about today is going to talk about the virtue of staying with that algorithm, and importantly, the psychology of the desire to depart from the algorithm.

Ultimately, we want to get to the question, if people do want to depart from algorithms, what can we do to make them soften their position? What can we do to make them more amenable to algorithms?

So what I want to do in part two and kind of in the body of the talk is to speak a little bit about whether we have any research that can inform these questions.

I've got a couple of papers and an ongoing project with two co-authors. The first paper was out last year in the

Journal of Experimental Psychology. Both papers are with Berkeley Dietvorst and Joe Simmons. Dietvorst just graduated last weekend from our PhD program. He has taken an assistant professor position at the University of Chicago. I'm very proud of him.

Joe Simmons, a longtime friend and collaborator of mine, has been right at the heart of the replicability crisis in psychology. So if you guys have read any of that stuff, if you know about the storm that has brewed there over the last six, seven years, Joe's paper with Uri Simonsohn and Leif Nelson was one of the key initial papers in that work.

So we've got this initial paper, and then we came in behind that and we've got a paper under review now that is about overcoming algorithm aversion. What I want to do in this section is give you basically the paradigm we've been using, but also two experiments.

There are three experiments in the first paper, four in the second, but I'm just going to give you one from each, kind of the greatest hits from each of these two papers to give you a sense of what we're doing in this research.

The idea we're investigating is that the forecasts of evidence-based algorithms outperform human forecasts. So algorithms are better than humans. This has been long established.

It goes way back to the psychologist Paul Meehl. Robyn Dawes is famous for it in psychological circles as well. And in these circles, there is no question that you should be using algorithms. Humans don't consider things that the algorithms do, which is one of the four main reasons they do worse than algorithms.

They don't include everything that they should in the model. They do include things that they shouldn't. They don't know how to weigh each attribute. They don't know the right weights to put on these things.

And then finally and probably most importantly, whatever they have in the model, they're not using it consistently. They're applying it with one set of weights in the morning and a different set of weights in the afternoon, or with three factors on Monday and three different factors on Tuesday. They're not consistent in using whatever algorithm they're using implicitly. So this is what we understand about why humans are worse than algorithms. We haven't known much about why they are resistant to algorithms. That's what we wanted to know.

There has been research on the fact that they do resist them. People have run horse races essentially, they have given people the opportunity to use them, people don't avail themselves to that opportunity. So we do have evidence that they don't use algorithms, but we haven't had much evidence on why they don't use these algorithms.

That's where we're coming from. Let me tell you where I'm coming from on the project. My motivation on this project came again from the NFL. I did some research a few years ago on the NFL draft and because of that, I've gotten pulled into consulting to football, baseball, and basketball organizations. But my main work has been with the NFL organizations.

And since about 2005, I feel like I have had truth on my side with how teams should allocate their draft capital in the NFL draft. It's not that I know which quarterback to pick, but I do have a good sense of the value of the first pick versus the value of the 32nd pick and what that should mean for their draft strategies.

At this point, we have very robust results, and I've got long-term relationships with multiple NFL teams, but it's a little depressing how little progress we've made with changing decision making in the NFL. Despite having, quote, truth on my side, and a better regression than the next guy, we don't move the needle very much.

That experience led to this project. That experience led me to realize we don't have the tools for opening people up to algorithms. Even though I'm not selling, I am a little bit selling an algorithm. More generally, I'm selling a

way of thinking about things that these guys are averse to, and we needed better tools for winning that argument.

So that's where I'm coming from, trying to better understand this not just for dry academic reasons, but to actually make progress. But the more I've done this, the more I realize this applies way beyond NFL organizations. This applies to many of our students who are going out and working in quantitative fields and trying to get a seat at the table where decisions haven't historically been made based on quantitative evidence.

So the research questions we're going to ask are, why do people choose humans over algorithms, and then, how can we get people to choose algorithms over humans?

We've done a number of studies in this area. As you've seen, we have a couple of papers. We still feel like we're just getting into it. We can say confidently what I'm going to say today, and then what we say beyond that is still speculative.

We're continuing this research and we have more hypotheses we want to pursue. These two, I feel like we can give at least.

So for this first question, the example I would give is one that is common to all of our experiences now, especially with tools like Waze. Say you're driving. To what extent do you use algorithms when you drive?

If you decide to change your normal route as you're driving, and you place yourselves in the middle of a traffic jam, how do you respond?

You may be unhappy, but I suggest that you don't lose much confidence in your judgment. It's probably not very hard for you to explain away what went wrong on that particular occasion.

What happens on the other hand if Waze or some other GPS (Global Positioning System) tells you to change route and you end up in the same tracking channel? Very often, it's a very different attribution. You kind of want to fire the GPS.

In fact, this was exactly my experience with my wife where I first started trying to get her to use Waze. We had the misfortune of having a bad experience with Waze right up front and it took probably three more months before she finally was convinced that it was superior to our judgment.

So this is the idea that we reason about guidance from ourselves differently than we reason about guidance from algorithms. And in particular, when we see algorithms err, we are much harsher on them than we are on individuals when we see them err. So why is it that people choose humans over algorithms?

One, they see that it is almost inevitable, and in the domains that we're studying, it is inevitable that algorithms are going to make errors. We're not looking at domains where perfect predictions are possible.

In fancy terms, these are aleatory uncertainties. There is some kind of irreducible uncertainty. If you were going to ask somebody, "is that coin going to be a head or a tail if I flip it, is that dice going to be a one or a six if we roll it," there is no way of knowing for sure. And in those cases, it's inevitable that an algorithm will err. So that's part of the setup. We think that captures many domains.

Second is that people will be more tolerant of errors made by humans and that they'll lose confidence in algorithms after seeing it. They will not necessarily lose confidence in humans after seeing it. So that's where our hypotheses are. To be clear, see algorithm err, lose confidence in algorithm, choose human instead.

Let me make a confession here. We pursued a number of different hypotheses when we first started studying this. We discovered in kind of the purest sense that it was this seeing error that is driving things, and I want to

give you some evidence on that. So let me give you a couple of experimental details. In study 1, we have 361 subjects. These are lab participants. They are estimating the success of MBA students.

We have 115 real MBA students, and we know how they did long-term measured in a number of different ways: how fancy a company they went to work for, how much money they made, ratings by their peers on graduation, GPA (Grade Point Average).

We combined all that into student success, and we ask the lab participants to forecast that essentially. How good given these inputs do you think these MBA students will be long-term?

So the question is whether they want to rely on their own estimates or a statistical model's estimates. That's always going to be the question. You can do this yourself or you can lean on a model to do it.

The model is the algorithm, obviously, and to do this, they're going to ultimately have ten forecasts. We're going to put in incentives where they can earn more money or less money depending on how well they do.

We tell them that they will estimate the actual percentile ranks of real MBA students based on their application, a statistical model will also estimate this performance, and then we give them a description of variables.

These are what the variables look like. These are the inputs: undergraduate degree, GMATs [Graduate Management Admission Test], essays, these are basically application materials.

These are things that MBA admissions departments know. In some ways, we're replicating the admissions decision. The design is that they're going to make 15 forecasts with no incentives and they get feedback on what actually happens.

This is the training set, if we're going to follow Pedro [Domingos]. They're going to learn on one set and then they're going to be tested on a different set. So this is the learning opportunity, and then what we manipulate during this learning opportunity is if they themselves make forecasts, and if they have access to the algorithm's forecast?

We're going to cross both of those things, so we have four experimental conditions. You are in one of these four conditions. Are you making your own forecasts with feedback, yes or no, and are you seeing the model's forecast with feedback, yes or no?

Again, this is the training phase. You're in 14 of these, you're either making forecasts or not, and you're either getting the model's forecast or not.

We have a human condition which is just making your own forecast and getting feedback; we have a model condition which is not making your own forecast, just observing the algorithm; and then we have the two diagonals which is model and human; and then the control condition where you're not going through the learning phase at all.

All of the manipulation happens during this learning phase. The subjects are doing 15 trials in one of these conditions, and it's just whether or not they're making predictions and getting feedback and/or whether they're doing the same thing with the model.

This is an example of the stimuli they get. Here is an MBA. Here are some basic demographics on the MBA. How well do you think this person is going to do in their MBA career? You might make this judgment yourself.

What percentile would you put them in? Start cranking it through your own model. You might also think about your confidence in that model. Would you like the support of a statistical algorithm? I hope most of you would.

And in this case, the average prediction was the 75 percentile, the model's prediction was 28, and I mean, the students' actual percentile, the answer essentially was two. So that may not be a super representative one. We naturally sampled this space. You're going to get some very noisy outcomes.

Just in the example of the stimuli they go through, they get the profile of demographics, they make a prediction or not, the model makes a prediction or not, and then they get this feedback.

This is an example from the human and model condition. And what I want to show you is what happens obviously when they are in these different conditions.

So they're going to get paid a \$1 bonus each time the estimate is within five percentile, so they've got some money riding on this. This isn't a lot of money, obviously, but for our experimental subjects, it adds up, and this is their basic choice.

After they have done the 15, they face this choice. They're going to make 10 more of these forecasts, and they have to decide if they want to use the statistical model or if they want to use their own judgment?

And this is really the most important question for us. They've had a chance to learn about this environment, they've had a chance to either exercise their own forecast or observe the model's forecast. They've done this 15 times with no incentives, just a learning trial, and now they're asked, okay, ten more times for money: do you want to use yourself or do you want to use the model?

The model in this case is from the same data. In this case, we didn't do the proper thing and model him on one dataset and test on another. So it doesn't have deeper but it does have the same. The model is working in-sample in this case. We fixed that in the second study.

The outcome is that the participant's forecasts had 15 percent more error than the model, which is reliably more error than the model. They would have earned 29 percent more money had they used the model's forecast.

This is how the students performed. This is how the model performed. The key bit is the percentage who chose to use the model. I'm breaking it down by the four conditions.

The control condition is where they didn't do this at all. The first question they got was, okay, you're doing ten of these forecasts, do you want to use your own or do you want to use the model? They didn't have any learning. What happens? About 65 percent of the time, they want to use the model.

This is a little bit like your experience. Would you rather use the model or your own judgment? This is important because sometimes our paper gets misconstrued as saying that everybody hates algorithms. We don't say that people hate algorithms. In fact, they come in naively here and about two-thirds would actually prefer the algorithm.

The next one is the human condition, where they made their own judgments, they got feedback, and they never saw the model do anything. What happens? Again, about two-thirds would actually prefer to use the model. So they realized they are not perfect in this and would rather use some statistical help.

The other two conditions actually saw the model perform, and I just reported to you how the model performed relative to how they perform. The model is better. The model is demonstrably better. Everybody who saw it saw it do better.

But their actual interest in using the model drops dramatically. Only 26 percent of the people who only saw the model perform chose it. They didn't have the humility of making their own predictions. So maybe this is the worst case.

But in fact, when they saw both, it's still just a quarter. And I actually oversee this. Because we're naturally drawing these 115 and because we have real subjects making real predictions, it's not the case that everybody in the study is outperformed by the model.

Some people actually do as well as the model just by chance, but even those who saw the model outperform themselves, and this is the majority of our people, still prefer to choose their own judgment 69 percent of the time over the model's judgment.

You can start to see where we're getting our conclusion. It's seeing the model err that leads to the inversion. It's not an inborn aversion to the algorithm, it's that in this world where prediction is tough, error is inevitable. When people see a model err, they punish the model.

We have some additional measures to give some insight into what's going on here. For example, we ask about confidence in the model, and we see that confidence doesn't change much when they see their own selves perform. You wouldn't expect it to. But the confidence gets hurt whenever they see the model perform.

So confidence goes exactly as the choice of model goes, in other words, confidence mediates their choice in the model. You might ask, well, what about confidence in humans? Did that play a role? That doesn't play a role.

Confidence in humans is constant across all four conditions. It doesn't matter whether they saw themselves perform poorly or not. They don't lose confidence in themselves. Back to the GPS example. Somehow they rationalize how their mistake leading to the traffic jam isn't a permanent feature of their judgment. That distinction in confidence is one of the clues to what's going on here.

We have done this in a few different conditions, a few different tests. We have done different tasks. We always need more of these tasks. There are only so many of these tasks you can run in the experimental setting, but we have tried forecasting various economic indicators, different kinds of student performance. We have manipulated the extent to which algorithms outperform the humans.

You might reasonably guess that when the algorithm is only a little bit better than the humans, some algorithm aversion might be more rational. So we put them in situations where the algorithm is a lot better than the humans, and yet, this algorithm aversion in the face of error remains robust.

And then we run studies where the choice isn't between a model and your own judgment, which introduces a lot of egocentric biases, but between a model and somebody else's judgment. And we see almost as much algorithm aversion. It's mitigated a little bit. There is, it seems a role for egocentric biases, but the bigger effect still is the aversion to algorithms over just human judgment in general.

So we have some exploratory measures on why this is. We reported these in our first paper and we ask, we literally just ask, what are some things that you might worry about if you're trying to compare judgments of models and humans. What do you think? Do you think humans are better? Do you think models are better?

So for some of these factors, they believe models are better than humans. For some, they believe humans are better than models. Models are better than humans, according to our subjects, at weighing information consistently, weighing attributes appropriately, and avoiding obvious mistakes. This is what our participants say. This is part of the psychology for their preference.

They say that humans are better than models at finding underappreciated candidates, detecting exceptions, learning from mistakes, and getting better with practice.

So for us, this comports quite a bit with our intuition. It's this possibility of getting better. It's the static feature of at least most algorithms, and laypeople's impressions of them, and the dynamic feature of humans where we

can learn and improve over time.

There are also some other interesting details. It isn't clear to me at all that avoiding obvious mistakes should be in that category. I'll give you another example of that later in the talk.

But this idea of detecting exceptions I think is also very close to the heart of it. It goes back to me showing you that graph of all the NFL games and the line through it, all kinds of exceptions.

Games deviate dramatically from our line. If you could detect those exceptions, fantastic, you could definitely improve on the algorithm. The trouble is believing you can detect those exceptions. And so, this I think is real close to the heart of the bias here, a preference for human judgment over the models.

Okay, that's study one, and the main takeaway is that people are much more likely to abandon an algorithm than a human after seeing them err. And again, despite the first two words of the title, algorithm aversion, we're not saying that people hate algorithms. It's that when they see algorithms err, which in many domains that we're interested in is inevitable, they punish algorithms more than they do humans.

The second paper is more on this question of what can we do about it? Are there ways we can mitigate that? We have brainstormed many ideas, we've run a number of studies, and we're sure we only have one of the answers. There are more to be pinned down, but I do want to share what we have on one of those.

So how can we get people to use algorithms? The idea here is that people want to retain some control over their forecasts. They are not ready to cede forecasts and decision making to these black box algorithms.

The idea is that they'll be more willing to use an algorithm when they can modify its forecast. Even when the ability to modify is quite constrained.

You might wonder what difference it would make if you were just given a modicum of control over that algorithm. Would it make a difference or do you need a lot? You just want a little control or would it be a lot before you would actually start ceding some decision rights to it? And what is your intuition for how other people will respond to algorithms? This is what we're going to investigate here.

So broadly, the setup is going to be the same. We're going to use a different dataset. We're going to fix this in-sample problem I mentioned earlier. We're going to use some online participants this time instead of lab participants.

One of the main learnings from the replicability crisis is that psych studies have been underpowered using far too small a sample. Doing work with Joe Simmons has lead me to use very large samples. Now we try to get 200 a cell generally, where this is going to be a four-cell experiment.

Eight hundred sixteen online participants, and the task in this case is to estimate students' standardized math test performance. We have a real dataset, real high school students, and again, they have to decide for these estimates, and we're going to incentivize them. But for these estimates, do they want to rely on their own judgment or do they want to rely on a model? Broadly the same design as we had before.

Finally and most importantly, the manipulation is how much they're able to modify the model's prediction. Here is the introduction they see: you will estimate the actual percentile ranks of 20 real high school students on a standardized math test, here are some independent variables, and here is a model that has estimated all students' percentiles using the same information you have. Oh, and by the way, the model is wrong by 17 ½ percentiles on average. We're just telling them upfront this is the model performance.

What we're saying here is look, it's hard. We usually say something like "informed, thoughtful statistician" or

“informed, thoughtful modeler.” We pimp the model a little bit.

But here, we also wanted to give them a truthful take on what the performance is. So we’re saying this is a hard domain, it’s noisy, the model is imperfect, it’s a good model but it’s imperfect, okay? That’s the setup. And we ran the training sessions again and then we gave them this choice.

So these are the variables they have, kind of doesn’t matter but just this is what they’re working with. Again, it’s a challenging task, it’s *prima facie* challenging. We all kind of thought, why wouldn’t anybody just go straight to the model? It turns out that they really don’t like to go to the model, but you can see that there is a lot of similarity to study one.

These are the incentives. We tried to crank it up a little bit though. We were only able to pay one of the four conditions on this bonus schedule. And then, here is the interesting bit, the four experimental conditions.

They were randomly assigned to one of these four experimental conditions. One is they can’t change the model. They have to choose between their forecasts or the model, and if they choose the model, they have to go with the model 100 percent.

The second is that if they choose the model, they can adjust the model by up to ten percentile. So if the model predicts 37.5, they can move it down to 27.5 or up to 47.5, or adjust by five, adjust by two . . .

They were randomly assigned to one of these two conditions. We were curious how that would impact the take-up of the model. We’re giving them a fair bit of control, a little bit of control, and almost no control over the model to see what the impact is on their interest in using the model.

This I think is the most interesting of the paper’s four studies and kind of captures the spirit of it most closely. This is something that we happened into as we started seeing what they were doing. We were pushing the idea that they like control, and we wanted to see how little control we could get away with.

You can adjust the model by ten percentiles. Now, you’re going to make a number of predictions. Do you want to use the model and adjust it or do you want to use your own judgment?

This is what we found. Again, I’m reporting to you what I reported to you for study one, which is the percentage of people who chose to use the model, and I’m going to show it to you by these different conditions. And for the can’t-change condition, about 47 percent chose to use the model, so it’s kind of in-between our two conditions before.

Remember, we told them that it errs, so they haven’t experienced the error firsthand, but they know that it’s noisy. They know that it has 17.5 percent error. They look at that and about half of them say, yes, I’ll use the model. About half of them want to use their own judgment.

What happens when you allow them to adjust the model by ten? Significant uptake in their interest in the model. Now we’re back up there above two-thirds of participants willing to use the model if they can tweak it a little bit.

And then, what happens if you let them tweak it but don’t let them tweak it as much? For 5 percent, it’s 71 percent of participants. And then, most interesting, what happens if you crank it all the way down and you can only tweak it by two percent, what happens? Sixty-eight percent.

We were really struck by this and this is the main point of the second paper. Yes, it’s kind of intuitive, people will like the model more when they can move it, but it’s counterintuitive as you don’t have to let them move it very much to get them to accept the model’s help. So we get dramatic uptake and model usage even though we’ve tightly constrained them.

People have demonstrated an inability to improve the model consistently, so we like being able to keep them very close to the model, so if we can do that without costing us uptake, then we're ahead of the game.

And so, that's in fact what we see. These people went about 50 percent with the model and about 50 percent on their own, and their average error was 22 percentile points.

If you can adjust the model by ten, now we have more than two-thirds of the people using the model. Because more people are using the model and they're not all actually adjusting it by that much, they have less error. Constrain model adjustments by five, you get even less error. Adjust the model by two, less error.

So you get this better performance. There is noise in here so it'd be nice if this were each one improving because we know that we're keeping them closer to the model. About the same number of people are using the model. You would expect this to come down. It's noisy.

We sampled truthfully, and so, sometimes the Bayesian performance or the model's performance doesn't do as well as it should. But the idea here is that by giving them these adjustment choices, in all cases, they outperform not giving them the control at all.

If you don't give them control, if you don't give them the option of some of the control, they don't use the model and they don't do as well. Give them a little room for control and they're much more interested in using the model. When they use it, they don't push it around that much and they end up performing better.

There are additional downstream consequences. We like this because we've made them perform better, but we're also really struck by the other downstream consequences. So we collected additional measures.

Things that we have learned are that they are dramatically more satisfied with the process when they can have some control over it. That's not terribly surprising.

We have a couple other which are a little more surprising. Being involved with the model, having some control in the model changes their beliefs about themselves and it changes their beliefs about the model. In particular, they become less confident in their own ability.

They learn some humility basically by engaging with the model in this way, and kind of profoundly, they get more confident in the model's ability.

So if you compared people who didn't have the chance to work with the model, these are people who were randomly assigned to these conditions, those who have had a chance to work with the model actually increase their confidence in the model's ability.

And then, finally, they are three times more likely to choose model-only. We give them a downstream task where they again get to choose, and this time, instead of randomly assigning them to these different conditions, we let them choose which condition they want to be in.

Again, we're trying to get closer to the real world. We're trying to get closer to your world where they get to choose whether or not to use algorithms or not, whether they impose algorithms on their employees. They're much more likely to choose the model-only world, which is the optimal world for prediction here, if they've had a chance to play with the model and if they've had some control in the earlier rounds.

So people are much more willing to use the model. It doesn't matter how much you constrained them to the limit, to our two percent limit even, and there are these positive downstream consequences from using the model.

Our main takeaway is in deciding whether to choose to use the model, people were insensitive to the amount by

which they could adjust the algorithm. It seems people want to have some control over the algorithm, not necessarily greater control.

The research questions we started out with were why do people choose humans over algorithms? And how can we get people to choose algorithms over humans? Our answers are they are less tolerant of algorithms' mistakes. This is why.

And then, how can we modify that? We can modify it by letting them modify the algorithm even if just a little bit. So those are the research parts. I wanted to share those, too. I tried to hew it down to just two experiments, give you a sense of the spirit of those two papers.

I want to close with a real world example, and I was doing this example concurrently, which is the admissions process at Wharton. It's not that we did this research and then I went out and applied it. I'm literally pursuing these two things concurrently.

How many of you have seen this movie with Tina Fey? I think it's called *Admissions*. She is at Princeton. As you can see from the orange in the background, she's the Princeton admissions officer. From what I'm told, this is actually a very representative take on what admissions is like.

A couple of years ago, our dean asked me to get involved with our MBA admissions. And so, for the last two years, I have worked very closely with those guys trying to bring the best of our world into that office. What Princeton does is no different than what all the Ivy League schools do, and we think of it as kind of Admissions 1.0.

Some very smart people back in the '70s decided they were going to improve on the legacy systems that had been the way people had been admitted for centuries, and they designed this system. It's been the model for not just these schools but for everybody in admissions around the country. And it's a very case-by-case, laborious, individual, all-the-details-matter kind of process. And we think that there are some ways we can improve it.

So what we've tried to do is what we so humbly call Admissions 2.0. Admissions 1.0, the Princeton model as depicted in that movie, is read the files, debate, and then make an individual decision.

What I mean by that is they are literally sitting around a conference table for a week with the stacks of files, and they are deciding one-by-one to go through 1,000 up-or-down, and they do this from first thing Monday morning until Friday afternoon when they're done.

You might worry about how systematic or how consistent they're being over that full day, and you might also wonder how do they make an optimal decision for the portfolio of whatever it is: 500 admitted students, 1,000 admitted students. How do you make the optimal portfolio decision if you're making sequential one-by-one, case-by-case decisions, which is what we worried about and which is what we've tried to address.

When the Dean first asked me to get involved, I think he thought that I would do some kind of forecasting thing as I have begun doing with some sports teams. But when I started looking at this, my interest was less in improving the forecast than it was improving the broader decision process.

So what we do is we do the forecasts up top, and then the key bit is this optimization in the middle. We're trying to make a portfolio, we're trying to optimize the portfolio as you guys would do in seven seconds as opposed to making case-by-case decisions for four-and-a-half days. And then we are trying, once we have this system, to evaluate and refine that system based on what we learn each year.

I will say that we're talking a little bit publicly here or else I wouldn't be here talking about it, but we're not pushing this story out there much right now. We hope to eventually.

We've actually admitted one class on this model. Well, actually, we have a full year of students under this model, and we've just admitted our second class. We will begin talking about it more publicly once we can say we just graduated the kids that we admitted under this model and it's actually a working system.

The key part is this portfolio approach. Everything that goes into the model is subjective. Every single input is subjective. So we're still using the readers, we're still using all the admissions experts, but once we have it in the model, basically we turn the crank and force the optimization to be consistent across everybody.

It's literally just a maximization model where we're trying to get as much of our objectives as possible, given the forecast of all the readers who have read these files. Everything is subjective, everything is human, all the aggregation happens systematically through the model question.

This is going to take us into more detail than I'm going to actually talk about right now. But we have our objectives. They're not very controversial. What we can see right now is what they do in their two years on campus.

Obviously, what we really care about is long term what they do with their careers, but what we see on campus is broader than just GPA. So we have a broader set of objectives. And so, we're going to forecast against those objectives.

We framed it to all of our readers as if it's a forecasting task when they read essays and letters of recommendation. It is a forecasting task. So give us your forecast, and then we have other considerations as well.

This is an important part of this story. This is Hilly Einhorn's "Accepting Error to Make Less Error." We are doing something at the portfolio level where we're applying all of our weights consistently, there is no difference whether you're the first file read, the 67th file read, or the 670th file read, you're getting the same weight. If you came in with an application this year, identically the same application the next year, you'll be treated the same. There is much more fairness, consistency. Everything is systematic now.

What we give up is this whole, "Is Joe from Highland Park in Dallas better than Ginny from Menlo Park in California?" We make that decision but we do it in the model and we do it in a very consistent way.

What we get back is four-and-a-half days where we can be more robust on whether it's the right portfolio decision to make. Now we can argue about what's the right percentage of women to have in the class? Not just argue about it, we can run sensitivity analysis asking what would happen if we cranked the percentage from this to that? And any other considerations you want, we can run them all and ask what's the impact? And what do we care about?

We have all these policy considerations. On the one hand, we're just maximizing these forecasts but we're also subjecting them to constraints on all other policy considerations. Because we do it this way, now we can talk about these policies. We can spend our time talking about the policy considerations instead of case-by-case.

The psychologists will tell you and the decision scientists especially will tell you that the biggest flaw on human judgment isn't not considering the right factors or even getting the weights right. The decision scientist will say just give them unit weights. The actual weights don't matter that much.

What matters the most by far is being systematic, being consistent in applying those same weights day-in, day-out. That's what we've tried to do here and that's what we're a year-and-a-half into exploring.

So what have we learned? I want to emphasize mostly a couple of things. Some of these are basic decision process stuff, getting the process right, prioritizing relationships, being a good translator. I want to emphasize two

things, numbers four and five.

Number four: cut decision-maker slack. As I said, I was doing this concurrently with the research on algorithm aversion. We got much more traction with the admissions office when we told them these aren't going to be binding decisions. Each round, we're going to make 1,000 recommendations to you, 1,200 recommendations to you. It's up to you on whether you use them or not.

For every applicant, we'll give you a one or a zero, admit or not, according to the model. By the way, it's your model and it's your input, so we're just turning the crank, right? We're going to give you 600, 1,200 ones and zeros, you decide what you want to do with them.

And in fact, we have learned that in the optimization, we have to build in the slack, so we don't actually allocate all 600 slots. We'll allocate some fraction of that knowing that we want to give them some room for subjective changes to it after the fact.

That has been critical. And you could think of it as strategic. It's our way of getting them to buy in. It's like the algorithm aversion experiment where you say you can adjust this model by ten or you can adjust this model by five. They will be more interested in the model if we give them that slack. But there is another reason: the other reason is number five, the supervise-your-algorithm reason.

And so, I'll close with a quick story on being wrong again. See, I'm book-ending it, Michael, with your theme. The need for supervision is real with these algorithms. And I know this is kind of a low-powered example relative to some of the things you guys do, but it's an important example. It affects a lot of people's lives. And I've been humbled by how dangerous algorithms are. It's like a power tool that does quick efficient work but you've got to be careful. You can do a lot of damage with the power tool.

So one really brutal example which I'll probably learn not to give on the record, but I'll give for the first time on the record. The first time we did this, this was literally making decisions on who to interview for MBA admissions round one.

Historically, the admissions folks had a ranking system, and one was the best you could get. You read an essay, a one is the best, four is the worst. You read a letter of recommendation, one is the best, four is the worst. So we collect all these data, we build a model, we're interacting with them all the time, we know that this is the scale, and yet, when we go to turn the crank, what do we do?

We maximize. We get into the meeting, we're presenting our results and we've got everything showing up, we're talking it through, and we are 15, 20 minutes into the meeting before we realize that we haven't figured out this distorted worst possible portfolio.

Because we ran the thing in the wrong direction. That's humbling and that's also why, playing with a power tool, you've got to have the safety glasses on, got to get the gloves on. It needs a little bit of supervision.

We've had a few other moments like that where you realize the beauty of not working with algorithms is that for all you guys who are going to make idiosyncratic error, you're unlikely to get one big thing wrong that hurts a lot of people. With an algorithm, error is not idiosyncratic anymore. You get one big thing wrong, you can take out a whole bunch of people in one go.

Okay, so I'll close with just a quick cartoon from "xkcd", the online comic strip. A lot of you guys probably know these guys. It's a clever little continuum of algorithms by degree of complexity, where our favorite cartoonist at xkcd says, "actually, in the world of complexity, forget anything Pedro [Domingos] does, his deep learning stuff, the most complicated algorithm is a sprawling Excel spreadsheet built up over 20 years by a church group in

Nebraska to coordinate their scheduling.”

What I love about this is it doesn't say that people are averse to algorithms. In fact, they will use algorithms, but they need to have some involvement in the algorithm. This may not be optimal, but you may be willing to go away from optimal if they are going to be committed to the algorithm. And I bet those Nebraska churchgoers are committed to that algorithm. Okay, guys, thank you and happy algorithming.

Question: With respect to your experiment where you allow plus or minus two, five, and ten changes, have you thought about or looked at instead, allowing the user to incorporate conditioning information, you know, X times in the trial? So two, five, ten times over the course of the period?

What I'm getting at is in thinking about it with respect to financial models. You like doing algorithms, but if there's an OPEC (Organization of the Petroleum Exporting Countries) meeting or there's a financial crisis or there's a terrorist attack, you might evaluate that the calibration of my models are probably inappropriate for this point in time. And that would help me avoid those really bad outcomes that can come, the errors that can be really bad. I'd imagine that if you allowed people control, you might see more buy-in.

Cade: It's a great observation. In fact, in one of our studies, in the first study in that paper, we had two different conditions. One was you can modify any given recommendation by a certain amount.

The other condition was you can strike any number of them and do your own thing, so it's closer to what you're talking about with the OPEC meeting or whatever. And we get the same comparable levels of uptake. They like that.

We just thought for further permutation, it was more interesting to play with the other. But as a modeler and a forecaster, I agree that it's critical but it's also just right back to the heart of the problem—when do you know that exception is okay and when is it not okay?

Question: How do you think about the “broken leg” problem from psychology, where you have the risk of clinicians overriding the formulas too frequently?

Cade: Right. There are way too many false positive. My collaborator on the NFL draft research is my advisor, Dick Thaler, who has some experience in the finance industry, and we have been working with the football teams for 10 or 12 years now. Early on, we had a sit-down with a coach and we were talking about game day decision making. We try to be humble in these meetings, we really do.

But this one coach in this one situation, we were talking about fourth down, I think, going for it on fourth down, and there's just so much data on this and it's been worked over in so many different ways. We're talking about this and the coach asks, what about the wind?

And you know that the wind is going to matter, of course it matters. But it was just that he asked a series of these questions: What about the wind? What about the left guard? There's always some other consideration he wants, he's never going to accept the model. So it's like an ongoing joke when we have these conversations with the “what about the wind type questions.” [Laughter]

Question: I'm just curious on the admissions process if the faculty feels there has been any change in the composition of the class or the behavior or the capability?

Cade: Yes, yes. So our admissions, the head of admissions, the vice dean of admissions who's been my partner in this all along, she came from the trading world, and so, she is pretty sophisticated on these fronts. And yet, she was a little worried.

So we had these events in the Spring, they were admitted over the Winter . . . we have this event in the Spring for all admittees. They come visit school, we kind of try to sell them on the school. They're going to our school, they're going to other schools, and she went to this first event the first year and she was worried.

She said afterwards she was worried that when she went in, it was going to be like going into the bar scene in *Star Wars*. [Laughter] You know, with all the freaks? She thought that's what we were going to have, which is misplaced. She's partly ribbing me but partly serious.

It's misplaced because all we've really done is codified the judgment that we pulled from what they've done the previous three or four years. All we've really done is do systematically what they've been doing a little less systematically in previous years.

And this troubled me for a little while because I come from the decision-making world, and we have obsessed for decades now about bias. And we're not fixing bias right now. We will move on to that kind of thing. We will tackle that challenge down the road.

But what we've done right now is we have just been more systematic, and my field was not focused on that in the past. Danny Kahneman has finally started talking about this some and Michael was there in the opening session of our conference this year.

It's possible that our field is worried too much about bias and not enough about noise. And this is a great example of worrying a lot about noise.

There has just been too much noise in the system, but even if we don't de-bias anything, even if we just codify what they've been doing, we're wringing a lot of noise out of it and everybody involved feels better about it because it's a more fair, just, systematic system. So that's a long way of saying they're the same students basically. We don't see any changes in class composition right now.

Question: When you gave your different groups different amounts of latitude in adjusting percentiles, did the ones with more latitude use it or did they not use it much? Were there any patterns of use based on how much they add?

Cade: So we had ten, five, and two, and you see ten's use it more than two's because they have more to use than two's, but they don't use it anywhere near as much as they could. And it varies by case, so they are paying attention to the details of the case.

I forget the exact averages but it's going to be something like four [percent], so it's really only a fraction of the discretion that they actually had, which was a pretty good clue to us that we could constrain them more. We ended up constraining them more than they were using. These should have been binding constraints, but they just didn't seem to mind very much.

Question: Hi, I have to say great presentation, but I had a visceral negative reaction.

Cade: [Laughter] Excellent.

Question: It feels like what you're doing is tricking people into accepting bad algorithms. And I want to go back to the football script which is out there right now. When I see that as a risk manager, what I want to say is, okay, show me the results conditional on the turnover differential. Show me your prediction, the score, if you replace all the field goal results with their expected value. I want to take the noise out until I can get to an algorithm that doesn't have a lot of noise. Now, there is still a lot of noise here because turnovers are pretty random and field goals are pretty random . . .

Cade: Yes. I'm sympathetic to that. I don't want to trick people into an algorithm until I'm very confident that it's the best option. Let me say one last thing on tricking. I am serious when I say that I'm more humble now about data than I've ever been in my life.

And I think the people I know who are best with models and data are actually the most humble. They end up getting humbled. You're talking with them a while and then on the other side of that, you actually get humbled about what data can do.

So I don't want to trick people. I honestly think it's better to have a little bit of involvement here and there. But I'm also in the persuasion game when I'm working with organizations. And so, if it's helpful then I'll use whatever tool is helpful.

The football example is a good one because that model is one that I've built over five or six years now, and I'm not giving up on getting it better. Every offseason, we tweak it a little bit.

But that model is as good a model as there is out there and people aren't going to beat it, they're just not. The odds of someone identifying the exceptions to that model are just exceedingly rare. And so, I'm very happy to kind of push it because I believe it's the best judgment. That's not to say it's going to be perfect because it's a very hard world.

I guess it's a fine line because I recognize that it can be improved and I want to work every offseason to make it better, but at the same time, it's about as good as it gets. I could give you the example from another football domain—the NFL draft where my early work was done. It's really hard to pick which quarterback is going to be better than the next quarterback in the NFL draft. It's really hard to forecast the performance of these college kids coming out.

It's not easy, and if you look at how much this has improved over time, this is a very humbling thing about this task. It's a very hard task. Take a correlation between how a player is evaluated coming out of college, say, by where he is drafted and some measure of his long-term performance: games started, career earnings, whatever, some correlation, okay? We could pick a lot of different numbers, just give me a correlation and ask how it's changed from like the mid-80s to the mid-aughts, 20 years' worth of work.

With the advent of computers, big data now, how has that correlation changed? Some correlation would be like 0.3. That 0.3 in 1985 is still 0.3 in 2005. There is just a degree of irreducible uncertainty.

So there is a limit to how much we can learn. But you talk about that with good draft guys and they'll say, yes, yes, yes, but we're getting better. And I believe it and I want to hold open that possibility because they are getting better. They absolutely are getting better. But until we're better, we need to be really humble and stay with the model.

Question: Can I ask in terms of the admissions process, one of the things that I like about the process is what happens to the debate in that it removes the discussion element of it and just in a sense aggregates the judgments of the individuals.

But what's the reaction then from the Admissions Committee about the absence of the data? I mean, I guess you're saying they do talk about it afterwards, but from my perspective, in some ways, talking actually can bias the discussion and judgments, assume people are more influential in a group. And if you remove that, it seems like you get close to kind of the wisdom of crowds. What's the reaction been in terms of not having those four-and-a-half days to talk about each individual candidate?

Cade: I think there was skepticism early and there has been broad buy-in having gone through this system,

even just having gone through it the first round the first year, and it's because we didn't remove debate. We just refocused the debate.

There is still debate on what the criteria are, how you judge that criteria from an essay, there is debate on what the right weights are across our objectives, there is debate on what's the right portfolio, the policy constraints, where we're pulling students from, and what the right mix of those are, we debate all those things.

There are debates on the exceptions. Whenever we look at that list of ones and zeros, 1,200 ones and zeros, we look at the people who are on the fence, we look at the people who if you shift the weights one way, they get out, if you shift the weight the other way, they get in. So for the marginal candidates, we go and look at them in more detail, and then they get debated. We get much more attention on the right places as opposed to spending all of this time where it's just not productive.

Question: I think it's great how this kind of defies one's decision making. The concern I have is about the outcomes. How do we measure the outcome?

Would it make sense to just run a sliver of things the traditional way against a sliver of things within the model and then you actually have somewhat of a natural experiment where over time, you can see that outcome might change, and you'll be able to go back and look?

Cade: Yes. Well, now I've got the opposite problem of this gentleman's instruction up here which is I'm going to rely on judgment that I know was noisy? Really, do I really want to do that?

Broadly, the spirit of the question, I agree with entirely. We want to run some experiments if at all possible. Back to the sample size issue, it's really hard to have enough in any given condition to draw much inference. And so, we've been humbled by what we're able to do and not do experimentally.

I want to say though about measuring outcomes, we know we're not getting that right now. And we don't know, we don't have the solution and we fully appreciate the difficulty of the task. The best thing to come of this process is that we're having the conversation now, that we're actually debating exactly how it is that we should be assessing our students' performance.

We were partly inspired to pursue this tack by Teach for America. You might not know it, but Teach for America are the most sophisticated hiring organization I've ever been around. They spoke at our first conference, the People Analytics Conference.

If you think about it, it makes some sense because they have 50,000 or 60,000 applicants every year for the same job. It's a homogenous job, more or less—tens of thousands of applications and they've been doing it now 15 years. They have really smart, quantitatively sophisticated people in that organization who have been refining their process for 15 years now.

They said this beautiful thing at our conference the first year. They said we're never going to be done. It's not a project to fix admissions. It's not a project to fix recruiting. It's an ongoing process. And they said they had this model inside, and they're never going to be done.

And so, Maryellen [Reilly Lamb] and I, the vice dean of admissions, have said that from day one. It kind of licenses us to say we don't really know right now how we're going to measure our outcomes because it's really hard, but we're going to start trying and we're going to have the conversation. And we're having the conversation now that we have never had before because of the process. But that's a very big and very difficult question.

Question: Thank you. Just in the first experiment, you were talking about how people have a very adverse

reaction to negative outcomes from algorithms. I'm curious with your college experiment, let's say you're solving for GPA and Nobel Prizes and wages and things, if that might make sense, but you also could suddenly get more axe murderers and other characteristics you may not be screening for. Wouldn't you say, oh, this is a problem, we can't do this because we have more people who might go to prison than we can measure.

Cade: [Laughs] I'm not worried about axe murderers per say. That's not something I'm worried about. But generally, this idea, I'm aware of something, more aware than anybody else because of the research I've done, I'm greatly privileged in getting this new system going in admissions, helping get it going because we won't see the outcomes for a couple of years.

Everything about this research says that people are getting more interested in algorithms if they don't get feedback, if they don't see the algorithm error. And so, it's like I've got this two-year window essentially to get things going in the most hospitable environment possible, and then, once we've started measuring, we'll learn that the algorithm is noisy.

We'll learn that those people that we ranked so highly sometimes don't turn out so perfectly and that will cause some people some concern, agreed. I hope we don't turn up axe murderers, but it will definitely be more challenging once we have hard outcomes that go against us. And there will be plenty because it's a hard task.

Question: I'm curious if you spend any time with your colleague, Phil Tetlock?

Cade: Sure.

Question: I mean, obviously, he's got his superforecasting approach and he's got all these myriad teams and some of those teams consistently outperform year after year.

My question is, have you thought about trying to participate and build algorithms to field your own team? And the second question is, do you think that there are elements of what those successful teams are doing that are algorithm-like in nature and that kind of coincide with what you're doing?

Cade: Certainly, I know and I thoroughly enjoy Phil, and Barb [Mellers] as well, and that's a phenomenal team and we do talk about these things. In fact, they've had a team interested in using data from admissions for one of their stimuli in some kind of study. So they're actively engaged in a way with Lyle Ungar, a computer science guy there.

For the second question, anything in what they're doing that's algorithmic is interesting. They don't very explicitly go down that road. There may be people who are individual forecasters who are working with algorithms, but that's not part of their shtick.

They have discovered some phenomenal things. They've discovered some qualities in good forecasters that differentiate good forecasters from bad forecasters. And probably as profound, have developed some training techniques that improve people's judgment under uncertainty.

And again, my field has looked at judgment under uncertainty all the way back to [Paul] Meehl. Really Meehl's not my field, but since Danny Kahneman and Amos Tversky did so pretty rigorously, no one has ever really improved upon it. And these guys have come along and come up with some techniques for actually improving their judgment. That's the kind of thing that we need to be incorporating.

We can train our admissions people, for example, using some of those same techniques, and we'd probably see better forecasts. But we're lucky to have Phil and Barb doing that work right there.

Michael: Well, we'll call it there. Thank you, Cade.



Paul DePodesta

Cleveland Browns

Paul DePodesta has made a career of evaluating, measuring, and assigning value to talent, as documented in Michael Lewis's book, *Moneyball: The Art of Winning an Unfair Game*. The Moneyball methodology has become a mainstay strategy for business leaders looking for new approaches for overhauling stagnant systems.

Formerly the Vice President of Player Development and Amateur Scouting for the New York Mets, Paul helped lead the team to the 2015 World Series for the first time since 2000. Mets GM Sandy Alderson said Paul was a "huge factor" in the Mets' success.

In January 2016, Paul joined the NFL's Cleveland Browns as Chief Strategy Officer. In this new role, he is responsible for assessing and implementing the best practices and strategies that will give the Browns the comprehensive resources needed to make optimal decisions for their players and team.

Paul is also an Assistant Professor of Bioinformatics at the Scripps Translational Science Institute.

Note: No transcript available.

FAILURE: ONE STEP CLOSER to SUCCESS

PAUL DePODESTA

"I HAVE SOME EXPERIENCE..."

A MEANS TOWARDS PROGRESS

SCOUTING REPORTS
RECORD THE ACTIONS
DATA

MEET EXPECTATIONS

EDISON BULB

WHY SOME SUCCEED?

THIS ONE DIDN'T WORK...

OPS

721-AVE
704 1ST
731 2ND
770 3RD

AT BAT W/ SAME PITCHER

WHY SOME PLAYERS FAIL?

DATA

JUST BECAUSE IT WORKS, DOESN'T JUSTIFY THE HIT

DRAFT

PROCESS & OUTCOME

learn & adapt

- 40 ROUNDS
- "JUST GET 2"

BUT I DON'T WANT TO FAIL!!!
OUTCOMES!

MARRY THE TWO CONCEPTS

WE GOT IT!!

BIAS

PROCESS

AFFIRMATION BIAS
SINGULAR ATTRIBUTE

TOO MUCH PRE-PITCH MOVEMENT

MVP
MIGUEL
FREE AGENT

"HE DOESN'T HAVE A CONTRACT"

I LOVE THIS HOUSE...

EMOTIONAL DECISIONS

DECISION DIARY

OBJECTIVE WAY TO MEASURE OUTCOMES

CONTINUOUS LEARNERS

0% PLAY TO THE PERCENTAGES

BUT STAY WITH ME!
outcomes

NORMAL to EXTRAORDINARY

ABSOLUTE & RELATIVE RETURNS!
EMBRACE VOLATILITY

inputs

ELIMINATE OPTIONS

1 2 3 4 NO 4s!

DON'T GET PHASED BY SOME BAD DRAWS
"IT'S A FLUKE"

FUTURE PICKS ARE DISCOUNTED

TEAM

MINDSET

ROOT FOR BROWNS!

TALENT & CHARACTER

GOOD NEWS FOR THE FUTURE TEAM

DOES IT FEEL RIGHT



FOOTBALL IS MORE THAN 1-1 MATCH-UPS

BETTER DECISIONS

CREDIT SUISSE

CLEVELAND BROWNS



ink factory
you talk. we draw. it's awesome.



This document was produced by and the opinions expressed are those of Credit Suisse as of the date of writing and are subject to change. It has been prepared solely for information purposes and for the use of the recipient. It does not constitute an offer or an invitation by or on behalf of Credit Suisse to any person to buy or sell any security. Nothing in this material constitutes investment, legal, accounting or tax advice, or a representation that any investment or strategy is suitable or appropriate to your individual circumstances, or otherwise constitutes a personal recommendation to you. The price and value of investments mentioned and any income that might accrue may fluctuate and may fall or rise. Any reference to past performance is not a guide to the future.

The information and analysis contained in this publication have been compiled or arrived at from sources believed to be reliable but Credit Suisse does not make any representation as to their accuracy or completeness and does not accept liability for any loss arising from the use hereof. A Credit Suisse Group company may have acted upon the information and analysis contained in this publication before being made available to clients of Credit Suisse. Investments in emerging markets are speculative and considerably more volatile than investments in established markets. Some of the main risks are political risks, economic risks, credit risks, currency risks and market risks. Investments in foreign currencies are subject to exchange rate fluctuations. Before entering into any transaction, you should consider the suitability of the transaction to your particular circumstances and independently review (with your professional advisers as necessary) the specific financial risks as well as legal, regulatory, credit, tax and accounting consequences. This document is issued and distributed in the United States by Credit Suisse Securities (USA) LLC, a U.S. registered broker-dealer; in Canada by Credit Suisse Securities (Canada), Inc.; and in Brazil by Banco de Investimentos Credit Suisse (Brasil) S.A.

This document is distributed in Switzerland by Credit Suisse AG, a Swiss bank. Credit Suisse is authorized and regulated by the Swiss Financial Market Supervisory Authority (FINMA). This document is issued and distributed in Europe (except Switzerland) by Credit Suisse (UK) Limited and Credit Suisse Securities (Europe) Limited, London. Credit Suisse Securities (Europe) Limited, London and Credit Suisse (UK) Limited, authorised by the Prudential Regulation Authority (PRA) and regulated by the Financial Conduct Authority (FCA) and PRA, are associated but independent legal and regulated entities within Credit Suisse. The protections made available by the UK's Financial Services Authority for private customers do not apply to investments or services provided by a person outside the UK, nor will the Financial Services Compensation Scheme be available if the issuer of the investment fails to meet its obligations. This document is distributed in Guernsey by Credit Suisse (Guernsey) Limited, an independent legal entity registered in Guernsey under 15197, with its registered address at Helvetia Court, Les Echelons, South Esplanade, St Peter Port, Guernsey. Credit Suisse (Guernsey) Limited is wholly owned by Credit Suisse and is regulated by the Guernsey Financial Services Commission. Copies of the latest audited accounts are available on request. This document is distributed in Jersey by Credit Suisse (Guernsey) Limited, Jersey Branch, which is regulated by the Jersey Financial Services Commission. The business address of Credit Suisse (Guernsey) Limited, Jersey Branch, in Jersey is: TradeWind House, 22 Esplanade, St Helier, Jersey JE2 3QA. This document has been issued in Asia-Pacific by whichever of the following is the appropriately authorised entity of the relevant jurisdiction: in Hong Kong by Credit Suisse (Hong Kong) Limited, a corporation licensed with the Hong Kong Securities and Futures Commission or Credit Suisse Hong Kong branch, an Authorized Institution regulated by the Hong Kong Monetary Authority and a Registered Institution regulated by the Securities and Futures Ordinance (Chapter 571 of the Laws of Hong Kong); in Japan by Credit Suisse Securities (Japan) Limited; elsewhere in Asia/Pacific by whichever of the following is the appropriately authorized entity in the relevant jurisdiction: Credit Suisse Equities (Australia) Limited, Credit Suisse Securities (Thailand) Limited, Credit Suisse Securities (Malaysia) Sdn Bhd, Credit Suisse AG, Singapore Branch, and elsewhere in the world by the relevant authorized affiliate of the above.

This document may not be reproduced either in whole, or in part, without the written permission of the authors and CREDIT SUISSE.