The Ultimate Guide to Machine Learning Job Interviews

Get your dream job, from hunting to accepting the offer.

Table of Contents

Introduction

The rapid success of Springboard's Machine Learning Engineering Career Track has reinforced our belief that there isn't enough easily accessible education about this exciting and fast-growing field. This tactical career guide is one way we can help.

While working with machine learning experts to design the course and talking to aspiring learners, we found that there was a mishmash of resources that discuss ML job interviews, but no complete guides. There were individual profiles and collections of interview questions, but no comprehensive resource with solutions, no profiles that spoke to: how do I actually get this job?

The goal of this guide is to help you navigate the entire process, from A to Z, to find and secure an interview in a machine learning job, whether as an engineer, analyst, product manager, data scientist, researcher, or whatever role you determine is right for you.

To develop this ebook, we wanted to talk to both hiring managers and job candidates, people from both sides of the table, to outline what this experience looks like -- and what the job market looks like—right now. We wanted to talk to recruiters who source candidates, hiring managers who conduct interviews and make offers, and successful candidates who have made it through challenging machine learning interviews.

At Springboard, we've taught thousands of people the data science and machine learning skills needed to switch careers and land fulfilling jobs. We have a large network of students, alumni, and mentors, putting us in a unique position to provide a realistic perspective of what the machine learning interview process is like.

The investment in a machine learning career is significant and it's not without challenges. But the return is incredible. AI and machine learning have been hailed as the breakthrough of the century. AI has the power to completely change finance, health, education, and countless other industries.

Becoming a machine learning engineer is a huge step toward future-proofing your career.

So, you're interested in AI and machine learning. But how do you actually turn that interest into a career? Let's get into it.

What Are AI, Machine Learning, and Deep Learning?

If you're getting into the field, let's say as a programmer or an analyst, but you want to brush up on your knowledge of key terms and definitions of AI, machine learning, and deep learning, this section is for you.

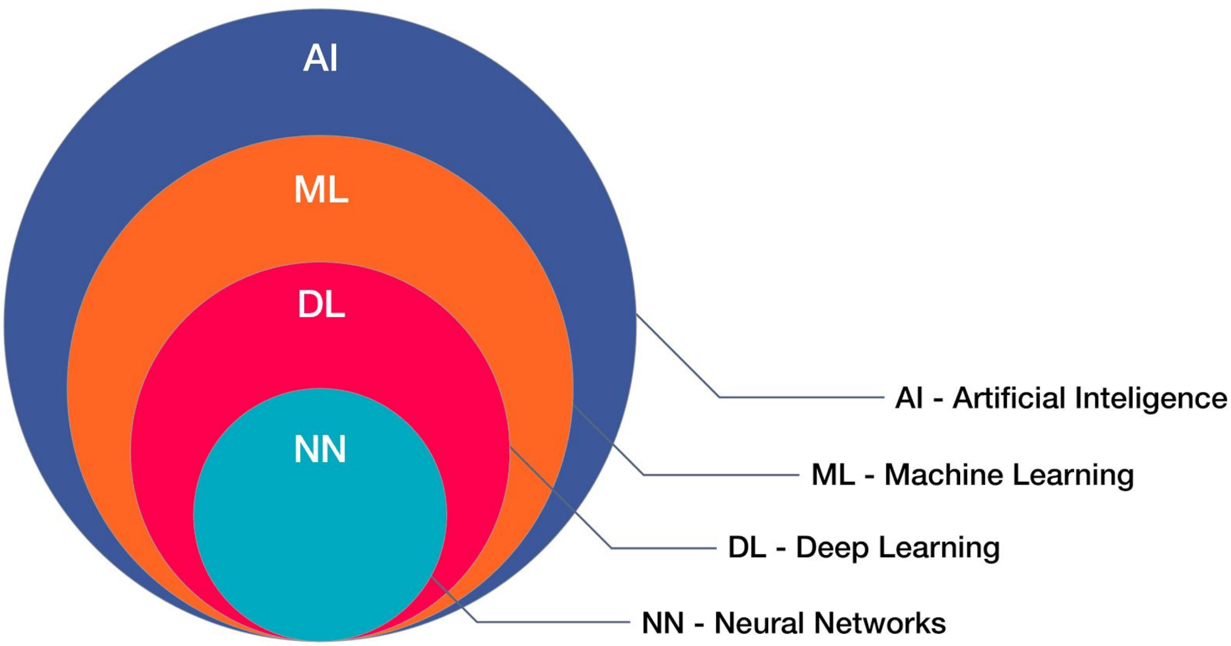
Artificial Intelligence

Artificial intelligence has been around since at least the 1950s, but it's only in the past few years that it's become ubiquitous. Companies we interact with every day— Amazon, Facebook, and Google—have fully embraced AI. It powers product recommendations, maps, and our social media feeds. But it's not only the tech giants that employ AI in their products. Now, startups, banks, consulting companies, and even governments are integrating AI solutions.

Simply put, artificial intelligence describes a machine that mimics human behavior in some way. AI can make the user experience similar to interacting with a human. The human part is the output. The input is huge amounts of data. That's what allows the AI to learn and adapt. It takes in reams of information and data and processes it. If it encounters a problem, it learns from the situation and recognizes a pattern.

There are many different terms being used, sometimes interchangeably and sometimes incorrectly, to describe artificial intelligence. AI is an umbrella term encompassing several different forms of learning. The main buckets are machine learning, deep learning, and neural networks.

Here's a simple visual for you to keep in mind:



Machine Learning

Machine learning is a subset of AI. It is a set of techniques that give computers the ability to learn without being explicitly programmed to do so. One example is classification, such as classifying images: in a very simplistic interpretation, for example, a computer could automatically classify pictures of apples and oranges to go in different folders. And with more data over time, the machine will become better and better at the job.

Andrew Ng, one of modern AI's pioneers, offers this helpful table on what machine learning can do:

What Machine Learning Can Do

A simple way to think about supervised learning.

INPUT A	RESPONSE B	APPLICATION
Picture	Are there human faces? (0 or 1)	Photo tagging
Loan application	Will they repay the loan? (0 or 1)	Loan approvals
Ad plus user information	Will user click on ad? (0 or 1)	Targeted online ads
Audio clip	Transcript of audio clip	Speech recognition
English sentence	French sentence	Language translation
Sensors from hard disk, plane engine, etc.	Is it about to fail?	Preventive maintenance
Car camera and other sensors	Position of other cars	Self-driving cars

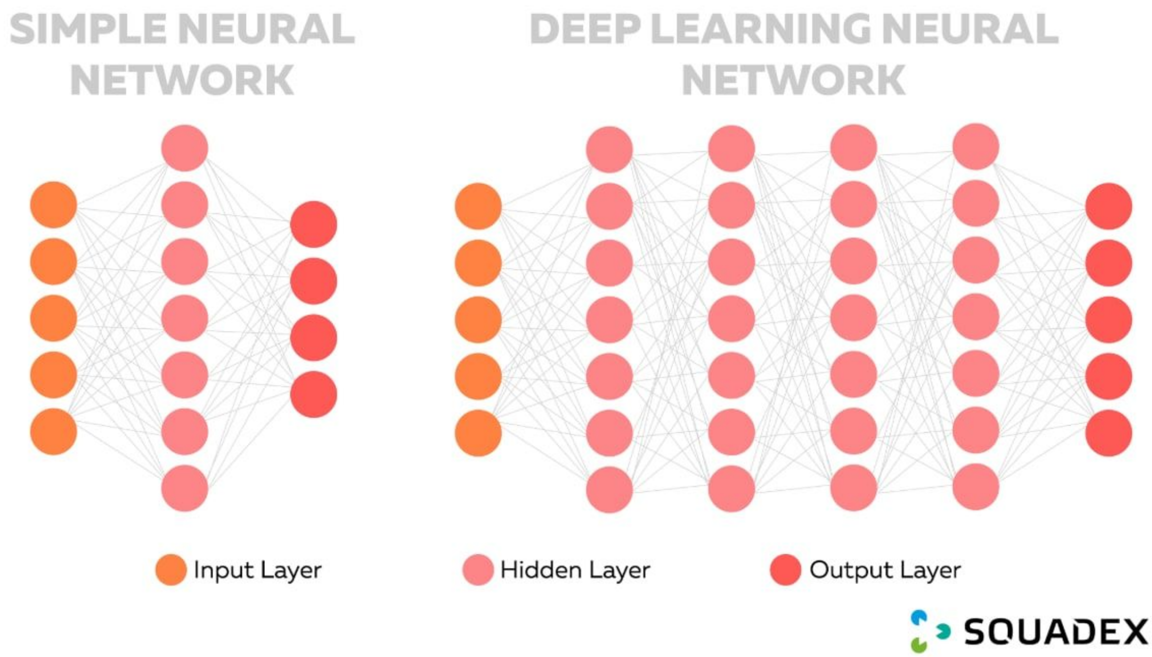
SOURCE ANDREW NG

Deep Learning and Neural Networks

Deep learning is a further subset of machine learning that enables computers to learn more complex patterns and solve more complex problems. One of the clearest applications of deep learning is in natural language processing, which powers chatbots and voice assistants like Siri. It's the recent advent of deep learning that has really been driving the AI boom.

Deep learning is based on neural networks, which is the idea that machines could mimic the human brain, with many layers of artificial neurons. Neural networks are powerful when they are multi-layered, with more neurons and interconnectivity. Neural networks have been explored for years, but only recently has research been pushed to the next level and commercialized.

Conceptually, here is a [comparison]() of a simple neural network to what a multi-layered neural network in deep learning may look like:



It's important to keep in mind that these are general, simplified definitions. Different organizations could have different definitions of each of these terms. And they could also have varying ideas about the depth they are looking for from an applicant. This can be difficult for candidates because there may be dramatically different job and interview requirements.

My recommendation is to supplement the technical section of this guide by searching for information about the company you're interested in to identify what you'll need to know. It's completely reasonable to do the following:

- Read as much as possible about the company to understand what kind of AI it uses and how deep a technical product it is.
- Look on Glassdoor to see if any employees have shared information about job interviews and/or the general nature of the business.
- Search LinkedIn to find ML-related employees and consider how experienced they are. For example, if they all have a Ph.D. in computer science or statistics, the company may have a complex, in-depth take on artificial intelligence.
- Reach out to the company directly, perhaps to someone in HR or a current machine learning team member. There's no harm in asking for additional insight.

Different Roles Within AI and Machine Learning

There are many different roles within the machine learning industry. In this book, I'm going to highlight the key areas you could work in and the most commonly associated roles:

- Engineering: Most commonly, a machine learning engineer. You handle the bulk of coding applications. You create systems that move data and implement algorithms designed by data scientists.
- Data Science: You could be a data analyst or a data scientist. Your job is to design machine learning models that create distinctions in data.
- Business Intelligence: You help query and present business implications, typically from data-driven insights.
- Research: While the other roles tend to be more applied, a research scientist is typically pushing the boundaries of artificial intelligence through new discoveries rather than by applying existing algorithms and models.

- Product: Finally, there are roles like a machine learning product manager or lead. This person brings together tech, business, and design in order to create the product. They're often responsible for facilitating the product development agenda and even managing profit and loss of a product.

Industries With AI and ML Careers

Before we talk more about roles, let's explore the industries that employ people in AI and ML careers.

Industries often focus on different areas of knowledge and have specific needs, language, and data types. For example, a software company will be focused on different metrics than a banking company. One of the key methods to keep in mind is code switching, which is not a reference to computer programming, but to the language and key terms that you express yourself in. Different industries have different jargon, and often, using it can help you converse properly and find a role at a company. Specifically, it can help you pass screening tests.

If you are looking to get into a certain industry, you'll want to optimize your resume and LinkedIn profile with keywords that are more common in that industry. It's important for screening purposes.

According to LinkedIn's Emerging Jobs Report, the largest hiring industries for machine learning, artificial intelligence, and data science talent are software, higher education, and consulting/finance companies. It turns out that these industries also often pay the most for machine learning talent.

Different industries also focus more on certain types of roles. For example, software, medicine, and telecom companies are typically the largest employers of data scientists. On the other hand, aerospace and information technology companies hire more engineers. And analysts tend to be hired by healthcare as well as consulting and banking companies.

It's important to be aware of the industry your potential employers will be in so you can learn more about their needs and also how they express themselves.

How Companies Think About AI and Machine Learning

Different companies can have very distinct interview processes.

In general, we can split companies into three rough categories:

Early Stage, Building a Machine Learning Product

The dream: a Silicon Valley paradise, working with a small team, raising millions of dollars, and changing the world. Besides the glory, there are major opportunities in working with an early-stage startup. For one, you may be able to work the fastest within one, and potentially see a large amount of success in a short period of time.

One important thing to keep in mind if you join an early-stage company is that your job description probably will not be static. You'll likely be required to do a number of different things across a number of different areas. Also, you'll probably be short on resources, so you'll need to be self-motivated, resourceful, and flexible.

Another element to keep in mind in a startup is that the front-loaded nature of your efforts might be quite significant. You typically won't have a lot of existing data to work with. And if you're looking to use other people's data for your own machine learning products, you're going to have to demonstrate a high level of reliability, security, and utility early on, all things that can be difficult for a smaller company (versus a larger company that has a lot of its own data or pre-existing relationships with customers). Also important to consider is whether the co-founders are unique and pioneering individuals within AI.

In a startup, you could have the opportunity of a lifetime, not only in terms of learning but in the potential financial windfall. But there's also a high risk that it could fail, as most new businesses do.

Examples of this company type: Kite, Looker, Nudge.ai

Sample job postings: Machine Learning Engineer, Data Analyst

Springboard

kite

Software Engineer, Machine Learning

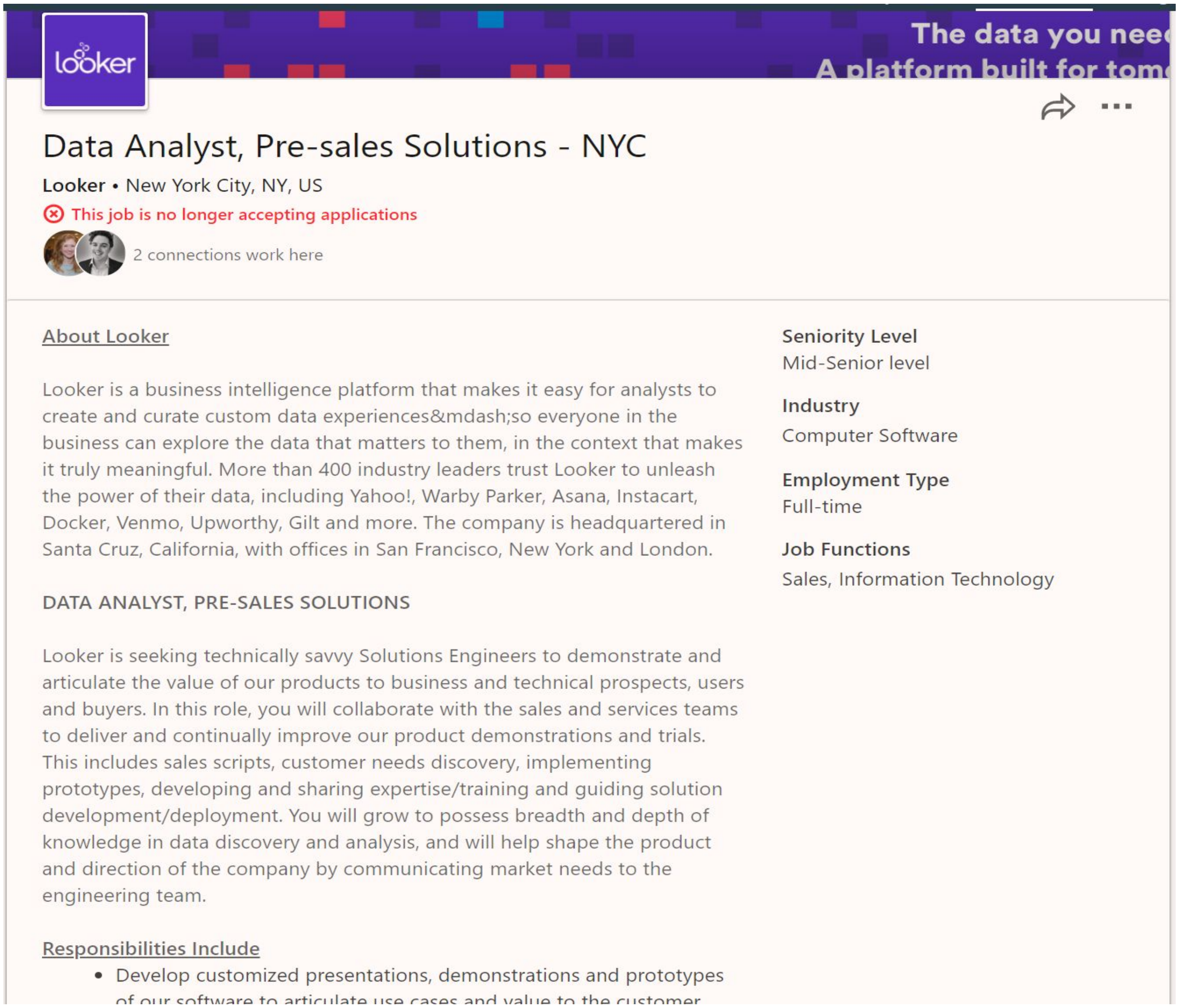
APPLY FOR THIS JOB

SAN FRANCISCO ENGINEERING - MACHINE LEARNING FULL-TIME

Programmers spend too much time doing repetitive work — copying and pasting from StackOverflow, fixing simple errors, and writing boilerplate code. We're building an AI code engine that does this work for you. Programming using Kite is faster and more fun.

Kite is well-funded by top investors in Silicon Valley, including the founders of PayPal, Stripe, Palantir, and Dropbox to name a few. We are looking to expand our 15-person startup with talented individuals who are interested in joining an early stage startup. The ideal candidate is excited to help guide the direction of our product and company. They will have a significant amount of ownership of critical technical components. Our team is growing rapidly and we hope you'll grow with us too!

We are looking for a talented Software Engineer with experience writing production level code and has the infrastructure background to implement our cutting-edge models. If you have a strong background in software engineering and either have experience with machine learning or a willingness to learn, come join the team that's changing the way people code.



Data Analyst, Pre-sales Solutions - NYC

Looker • New York City, NY, US

This job is no longer accepting applications

2 connections work here

About Looker

Looker is a business intelligence platform that makes it easy for analysts to create and curate custom data experiences—so everyone in the business can explore the data that matters to them, in the context that makes it truly meaningful. More than 400 industry leaders trust Looker to unleash the power of their data, including Yahoo!, Warby Parker, Asana, Instacart, Docker, Venmo, Upworthy, Gilt and more. The company is headquartered in Santa Cruz, California, with offices in San Francisco, New York and London.

DATA ANALYST, PRE-SALES SOLUTIONS

Looker is seeking technically savvy Solutions Engineers to demonstrate and articulate the value of our products to business and technical prospects, users and buyers. In this role, you will collaborate with the sales and services teams to deliver and continually improve our product demonstrations and trials. This includes sales scripts, customer needs discovery, implementing prototypes, developing and sharing expertise/training and guiding solution development/deployment. You will grow to possess breadth and depth of knowledge in data discovery and analysis, and will help shape the product and direction of the company by communicating market needs to the engineering team.

Responsibilities Include

- Develop customized presentations, demonstrations and prototypes of our software to articulate use cases and value to the customer

Seniority Level
Mid-Senior level

Industry
Computer Software

Employment Type
Full-time

Job Functions
Sales, Information Technology

Mid-to-Large Companies Looking to Leverage Their Data

AI and machine learning are still relatively new on the hype cycle, and there are a number of companies that have built sizeable data sets that can incorporate ML into their business. That way, they can leverage those data sets to improve their existing products and potentially to develop new products.

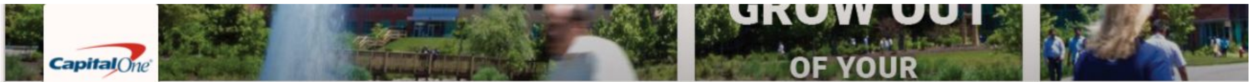
Every company with data is realizing that taking advantage of it is a top priority. Many companies are still figuring out how to do this, though. A common strategy is to create a startup team within the company that can turn data into insights and eventually new products for the business. Most companies realize that leveraging their data has become essential to remaining competitive. So, that means if you're seeking out a machine learning opportunity with a midsize to a larger company, you know that you have a strong case to make.

Mid to large companies will be more rigid in their culture or in the systems that they have set up, which can make it harder to innovate. But you'll have data that you can use to build machine learning models based on millions of data points. And you also know that you have a chance to immediately make more of an impact at scale, given the number of users and customers that you're likely to have.

While some of these companies may not be on the cutting edge of machine learning or AI innovations, they still offer a fantastic opportunity to learn, and you'll have solid compensation and benefits, as well as an overall solid foundation to stand on.

Examples of this company type: Capital One, JP Morgan, Morgan Stanley, Coca-Cola, Walmart, General Motors

Sample job postings: Software Engineer, Machine Learning, Big Data Engineer



Springboard

Machine Learning Engineer

Capital One • Vienna, VA, US

Posted 4 days ago • Be among the first 25 applicants

4 connections work here

Save Apply

Towers Crescent (12066), United States of America, Vienna, Virginia

At Capital One, we're building a leading information-based technology company. Still founder-led by Chairman and Chief Executive Officer Richard Fairbank, Capital One is on a mission to help our customers succeed by bringing ingenuity, simplicity, and humanity to banking. We measure our efforts by the success our customers enjoy and the advocacy they exhibit. We are succeeding because they are succeeding.

Guided by our shared values, we thrive in an environment where collaboration and openness are valued. We believe that innovation is powered by perspective and that teamwork and respect for each other lead to superior results. We elevate each other and obsess about doing the right thing. Our associates serve with humility and a deep respect for their responsibility in helping our customers achieve their goals and realize their dreams. Together, we are on a quest to change banking for good.

Machine Learning Engineer

Responsibilities

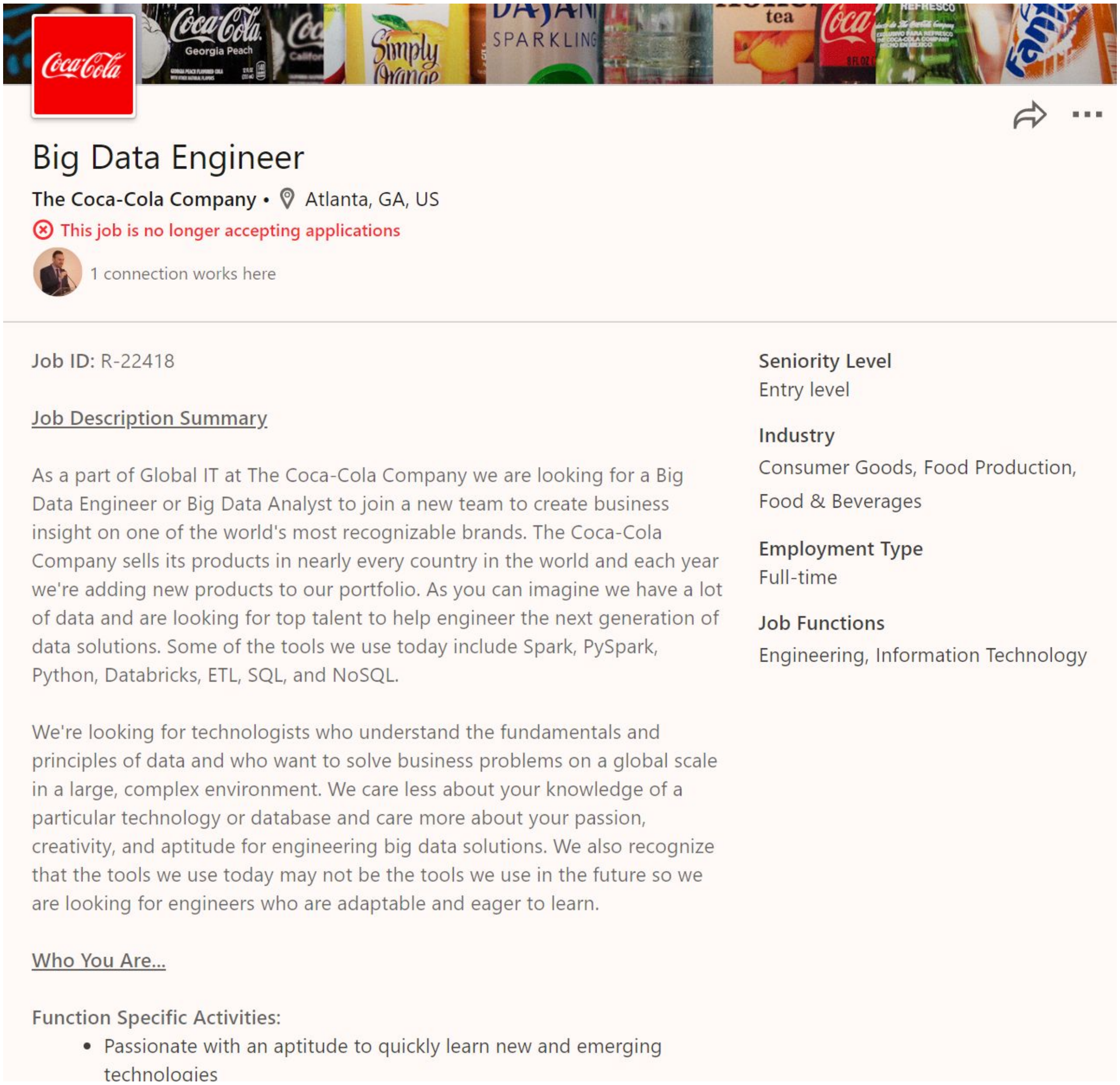
- End-to-end/full stack software solutions development and deployment of our Machine Learning (ML) and Artificial Intelligence (AI) algorithms.
- Write, design, code, test, implement, debug, and validate applications; document design decisions and develop modular software components; monitor system performance metrics, and

Seniority Level
Associate

Industry
Banking, Financial Services, Investment Banking

Employment Type
Full-time

Job Functions
Information Technology, Engineering



Big Data Engineer

The Coca-Cola Company • Atlanta, GA, US

This job is no longer accepting applications

1 connection works here

Job ID: R-22418

Job Description Summary

As a part of Global IT at The Coca-Cola Company we are looking for a Big Data Engineer or Big Data Analyst to join a new team to create business insight on one of the world's most recognizable brands. The Coca-Cola Company sells its products in nearly every country in the world and each year we're adding new products to our portfolio. As you can imagine we have a lot of data and are looking for top talent to help engineer the next generation of data solutions. Some of the tools we use today include Spark, PySpark, Python, Databricks, ETL, SQL, and NoSQL.

We're looking for technologists who understand the fundamentals and principles of data and who want to solve business problems on a global scale in a large, complex environment. We care less about your knowledge of a particular technology or database and care more about your passion, creativity, and aptitude for engineering big data solutions. We also recognize that the tools we use today may not be the tools we use in the future so we are looking for engineers who are adaptable and eager to learn.

Who You Are...

Function Specific Activities:

- Passionate with an aptitude to quickly learn new and emerging technologies

Seniority Level
Entry level

Industry
Consumer Goods, Food Production, Food & Beverages

Employment Type
Full-time

Job Functions
Engineering, Information Technology

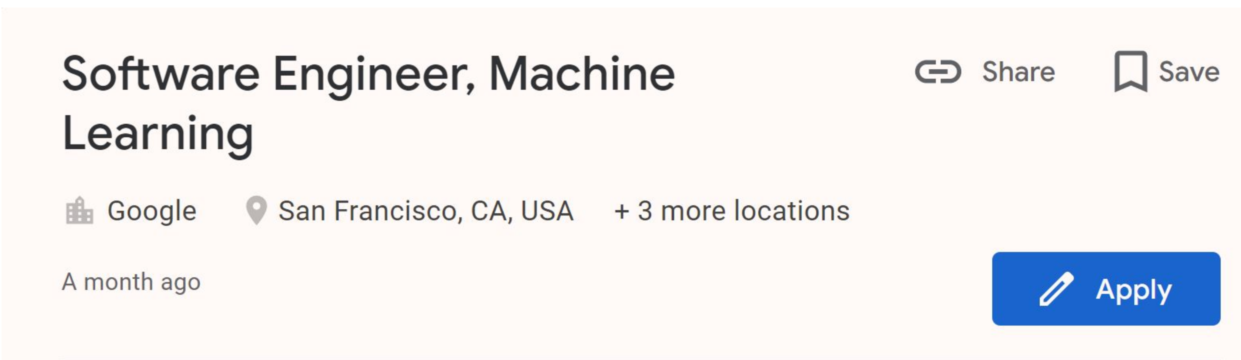
Large Tech Companies With Strong ML Capabilities

Some large technology companies already have very established machine learning teams. These are typically large software companies because they are the ones that pioneered these fields. They often have big technical teams and some of the most talented employees in the world. You'll often be working on complicated problems that require very innovative thinking.

If you want a challenge and some of the best training in the world, this is the target. You will have access to immense amounts of data. You likely won't be able to move as fast as some of the early-stage startups, but you'll have a good balance between those and a traditional company, with an impressive compensation package, and often a very well recognized brand on your resume when you decide to move on.

Examples of this company type: Google, Microsoft, Uber, Airbnb, Facebook

Sample job postings: Software Engineer, Machine Learning, Data Scientist, Strategic Analytics



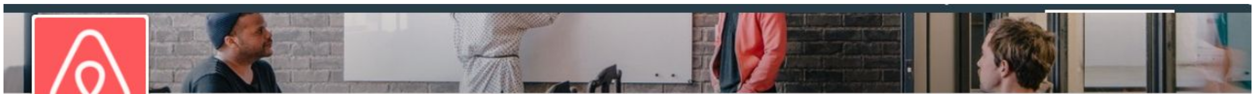
Minimum qualifications:

- BA/BS degree in Computer Science or related technical field or equivalent practical experience.
- 2 years of work or educational experience in Machine Learning or Artificial Intelligence.
- 1 year of relevant work experience, including software development.
- Experience with one or more general purpose programming languages including but not limited to: Java, C/C++ or Python.

Preferred qualifications:

- MS or PhD degree in Computer Science, Artificial Intelligence, Machine Learning, or related technical field.
- Experience with one or more of the following: Natural Language Processing, text understanding, classification, pattern recognition, recommendation systems, targeting systems, ranking systems or similar.

Springboard



Data Scientist - Strategic Analytics, Experiences

Airbnb • San Francisco, CA, US

This job is no longer accepting applications

5 connections work here

Airbnb Experiences are handcrafted activities designed and led by local experts. We launched Experiences in November 2016 with the ambitious goal of taking Airbnb from an accommodations business to an end to end travel platform. Since then, the Experiences team has been growing the new product exponentially, helping our hosts build new businesses and our guests experience travel in completely new ways. In the past year, we've seen a wide variety of Experiences succeed, everything from cooking in the Tuscany Countryside to flying a plane over Los Angeles to a Wolf Encounter.

The Data Science team plays a crucial role in realizing this mission by understanding our users and fueling actionable data insights to drive huge impact in a wide range of areas from influencing business strategies, product direction to operational excellence. As the Data Scientist- Strategic Analytics, you'll have the potential for massive impact, influencing key strategic decisions by bringing data to the table when it matters most.

Responsibilities

- Recommend business strategy using deep understanding of data & what drives growth for our business
- Drive impact through compelling data backed recommendations to executives
- Frame & communicate your recommendations with a clear view into the assumptions you have made and the rigor employed in the analysis
- Investigate questions around user experience to understand the voice of our users at scale
- Leverage Airbnb's rich data and state-of-art data science

Industry
Hospitality, Internet, Leisure, Travel & Tourism

Employment Type
Full-time

Job Functions
Research, Science, Engineering

How to Look Into Companies

One of the most important parts of getting a good job in machine learning and AI is first identifying the quality of the company that you want to join.

Evaluate Their ML Technology and Approach

From a technological perspective, you'll want to consider what problems the company is trying to solve, its approach, its data, how it audits and monitors itself, and whether the company is thoughtfully applying machine learning.

I've quoted these strong guiding questions verbatim from Karen Hao's recent publication in the MIT Technology Review, because they are simply so spot-on:

1. What is the problem it's trying to solve? What does the company says it's trying to do, and is it worthy of machine learning? Perhaps we're talking to Affectiva, which is building emotion recognition technology to accurately track and analyze people's moods. Conceptually, this is a pattern recognition problem and thus would be one that machine learning could tackle (see: What is machine learning?). It would also be very challenging to approach through another means because it is too complex to program into a set of rules.

2. How is the company approaching that problem with machine learning? Now that we have a conceptual understanding of the problem, we want to know how the company is going to tackle it. An emotion recognition company could take many approaches to building its product. It could train a computer vision system to pattern match on people's facial expressions or train an audio system to pattern match on people's tone of voice. Here, we want to figure out how the company has reframed its problem statement into a machine-learning problem, and determine what data it would need to input into its algorithms.

3. How does the company source its training data? Once we know the kind of data the company needs, we want to know how the company goes about acquiring it. Most AI applications use supervised machine learning, which requires clean, high-quality labeled data. Who is labeling the data? And if the labels are subjective like emotions, do they follow a scientific standard? In Affectiva's case you would learn that the company collects audio and video data voluntarily from users, then employs trained specialists to label the data in a rigorously consistent way. Knowing the details of this part of the pipeline also helps you identify any potential sources of data collection or labeling bias (See: This is how AI bias really happens).

4. Does the company have processes for auditing its products? Now we should examine whether the company tests its products. How accurate are its algorithms? Are they audited for bias? How often does it re-evaluate its algorithms to make sure they're still performing up to par? If the company doesn't yet have algorithms that reach its desired accuracy or fairness, what plans does it have to make sure they will before deployment?

5. Should the company be using machine learning to solve this problem? This is more of a judgement call. Even if a problem can be solved with machine learning, it's important to question whether it should. Just because you can create an emotion recognition platform that reaches at least 80% accuracy across different races and genders doesn't mean it won't be abused. Do the benefits of having this technology available outweigh the potential human rights violations of emotional surveillance? And does the company have mechanisms in place to mitigate any possible negative impacts?

In my opinion, a company with a quality machine learning product should check off all the boxes: it should be tackling a problem fit for machine learning, have robust data acquisition pipeline and auditing processes, have high accuracy algorithms or a plan to improve them, and be grappling head-on with ethical questions. Oftentimes, companies pass the first four tests but not the last. For me, that's a major red flag. It demonstrates that the company isn't thinking holistically about how its technology can impact people's lives and has a high chance of pulling a Facebook later on.

So, use this fantastic five-question framework from Karen Hao to assess whether a company is really on the right track with their machine learning product.

An even shorter, no-messing-around list from a friend of mine in the startup space is:

- Competent and transparent leadership
- Smart and stupidly high-energy people
- Early stage, maybe first official product or engineering hire
- Validated product-market fit with lots of product work to do
- Extremely difficult problems that'll keep you up at night
- Solving a real fundamental need

Whatever your most important criteria are, put those first, and don't settle.

Evaluate the Company Culture

People talk all the time about culture and how important fit is, both for a candidate and for a company.

My ultimate recommendation is that this is best done during the interview process, specifically the behavioral portions. There is nothing like in-person time with a company to learn more about how it really operates. That being said, there are some simple ways that you can “screen” a company to find out more about its culture and people.

1. **Check out Glassdoor.** It’s probably the best online repository of information about companies, including reviews from people who have interviewed with the company as well as current and former employees. If you see a lot of bad reviews, it’s pretty simple: you should probably stay away. So, it's a good, quick reference check to evaluate a company’s reputation. However, the Glassdoor approach can be tricky, especially with a very small or very large company. With the former, it's possible that there won’t be any reviews at all. On the other hand, with a very large company, it's possible that none of the reviews pertain to the machine learning part of the business.
2. **Read up on recent company news.** One of the easiest ways to do this is to go to the company's website and check out their online press room to see what they've been putting out recently. But that’s just the company’s approved PR. You should also do at least some light Google searching. Obviously, if they've had some bad press recently, you might want to do additionally due diligence about the business.
3. **Look on LinkedIn.** Check out people who currently work at the company, both people you would directly work with and employees at the director level and higher. You might want to check out their backgrounds to see the schools they attended, activities they are

involved in, and of course their most recent work history and how they describe their career. If you see some similarities and some things that interest you, that could be a sign that the company is a good fit for you. You should also keep in mind that some companies simply hire differently. For example, some are looking for specialists versus generalists. Some are looking for fully formed candidates while others are looking for people that they can train. Looking at profiles of current employees can help you get a sense of what that company typically likes, and then you can consider whether that works for you.

Now that you've looked into companies a bit, let's look into getting an interview.

8 Paths to Getting a Job Interview

Scoring an interview is sometimes the hardest (and most frustrating) part of getting the job, even more than getting through the interview. So in this section, we're going to help you figure out how to land an interview in the first place.

I separated the section into both traditional approaches as well as newer proactive approaches, which could help separate you from the candidate pool at startups in particular.

Traditional Paths to Getting an Interview

1. Job Boards and Standard Applications

Most companies post jobs on their career site. You can always target a company and respond to a specific job posting or submit a general application expressing broader interest in the company through that jobs portal. You can also find machine learning job postings on websites like Indeed or LinkedIn. These are old faithfuls. Definitely invest some time there.

Recent job seekers have told me that they like Breakoutlist, AngelList, and Triplebyte.

There are specific job boards for the machine learning space, such as ML Jobs List, as well.

I've also heard that aggregators from VCs such as First Round Capital, Greylock, and Costanoa are quite good.

2. Recruiters

You'll typically work with a recruiter during the interview process, but you don't just have to wait for them to contact you. Sometimes you may want to reach out directly to recruiters, either at the company itself (the in-house recruiter) or third-party recruiters who put companies in touch with great candidates.

There are recruiters who specialize in machine learning and artificial intelligence. Very often they include this in their LinkedIn profile title, or they may list something more general, such as "technology recruiter." What's important about recruiters is that they may know about the jobs that aren't even posted online. Keep in mind that something like 50% of the jobs are not even listed publicly. Quickly searching LinkedIn will give you a sense of some recruiters near you who might be able to find you the most relevant companies.

3. Job Fairs and Trade Shows

Job fairs present a rather daunting perspective: who wants to be milling around with a whole bunch of other candidates trying to chase down company representatives at a booth? But major universities in particular can have decent job fairs. However, my real recommendation is that networking events and meetups within your local machine learning community will probably be better than a traditional job fair. (Read on for more on those!)

Proactive Paths to Getting an Interview

While the options above are pretty traditional, it's more and more common for candidates to take a different tack when it comes to getting a job interview. Often you'll need to hustle and demonstrate creativity and grit in order to get a position. Startups are one of the main areas of new jobs in AI and machine learning, and are known for pioneering a different type of interview style.

4. Attend or Organize an Event

This is often the best way to meet people interested in AI and machine learning in your community, and you may also learn about job opportunities from attendees. There are large conferences and smaller or more focused community meetups that you can target, depending on what you're looking for.

Conferences

International Conference on Machine Learning (ICML)

ICML is one of the leading international machine learning conferences in the world, with over 35 years in the business. Usually hosted in California, it's full of expert speakers on the current state and future of machine learning. You should consider this event if you're a machine learning engineer, but there's likely something for everyone connected to machine learning at this one.

Artificial Intelligence Conference

This conference focuses on the latest breakthroughs in AI and machine learning. It's an O'Reilly event that brings together science and business, featuring speakers from top software companies, training courses, and networking opportunities. It's a great place to learn about different applications of artificial intelligence. It's especially well aligned to product managers and business intelligence developers.

Neural Information Processing Systems Conference

Another long-time conference (since 1987), this one is probably the most focussed on theory and research into the latest developments in machine learning. There is a significant focus on computational neuroscience, so it would be perfect for machine learning researchers or individuals from theoretical backgrounds to consider.

Meetups

Sometimes it may be more in your interest to attend smaller community events where you're able to make a bigger impression and potentially even take on a leadership role after some time. It's also a way to get to know people in a local community. Often, large events are chock-full of vendors and people focused on bigger partnerships. Small events can give you easier access to hiring managers and help you connect with peers who could help you down the road. You also may be able to find an opportunity to snag a speaking gig or start to build your name within the community.

One of the best stops for meetups is very simply Meetup.com. You'll find all sorts of them on the site. Some are quite large; for example, the NYC Machine Learning Meetup has more than 13,000 members. But don't let that intimidate you; many people will register for a Meetup but not actually attend. Typically a smaller meetup is between 10-200 people.

Another alternative if you can't find a good meetup or one nearby: create one yourself! I know many candidates for roles that have gotten jobs because they started the relevant community group. It makes you a connector or an influencer in the community. And that's a great role to add to your CV.

5. Freelance and Build a Portfolio

There's no reason that you can't start doing machine learning work right way. One of the easiest ways is to start freelancing. This will likely be easiest for designers, engineers, and data scientists, though there certainly can be opportunities for researchers and product managers as well. Sites like Upwork or TopTal make it easy as a skilled professional to create a profile and find work in short-term contracts and ongoing projects, or even longer-term engagements.

A portfolio can help you build your brand and also be an online record of experience for your work. It can also give you some early references and testimonials that you can pass on to potential employers. And of course, it could allow you to do some genuinely interesting work that could inspire articles for blogs or other content that you can use to expand your profile.

Ultimately, freelancing may also simply be a good idea for you to validate the different types of work and industries that you are interested in, to help you narrow your job search and be more specific in the future.

6. Get Involved in Open Source

Another way to make connections in the machine learning community is to get involved in open-source projects. These are non-proprietary code bases and repositories that are worked on by distributed communities and are typically not meant for profit or owned by a company. They are often in open-source repositories on Github. This includes the Natural Language Toolkit project, which helps deal with human language as a data source, and the various libraries that make up the Python data science and machine learning toolkit.

Companies hiring engineers are often particularly known to hire based on open-source contributions, and sometimes will find you through what you wrote. It's similar to the portfolio effect. People will often look you up online and want to see what you worked on.

7. Participate in Competitions / Hackathons

If you prefer to use your skill set in a more confined or time-limited environment, perhaps a competition or a hackathon is the way to go.

There are machine learning competitions like Kaggle and many hackathons that allow you to work quickly on real business or social problems. It's a great way to put your skills in machine learning and artificial intelligence to use, and you will be able to meet people as well as showcase your ability to make a difference.

8. Informational Interviews

The final path in, or step that you can take, is a classic, but it may be irreplaceable. Ultimately, relationships are what can start you on the path to a job, and help you close the deal. More than half of jobs aren't even posted on job boards and sometimes the only way through a company's seemingly impenetrable shell is to start meeting people from that company and build strong relationships with them.

One of the best ways to network is to request only a little bit of someone's time. A quick coffee date is ideal. Meet them on their schedule, at a place of their choosing. Reach out via email or a LinkedIn message with a very short note. All you need is one sentence about why you're unique as a candidate. You could also use this great framework from Steve Blank.

If you're successful in getting coffee, treat it as an opportunity to seek advice and information from people in the field. And if you're good at growing your network, you'll really get a sense of how the industry works.

Building Your Profile for Recruiters

How to Use References and Your Network

Let's go deeper into networking. One of the most powerful sources of information for companies is a strong referral, especially if that referral is coming from someone who's already part of the company. If you have someone looking out for you on the inside, they'll ensure that your application gets looked at, and sometimes you can even jump ahead in the interview process.

What's important to know is that these referrals don't need to be friends. It could be as simple as having basic name recognition with a person and making a good impression. They might put in a good word and say, "Yeah, I met Rob once, he was great." That could help you make it past the initial screen.

The important thing is to build relationships before you need them. You want to take a long-term view on this and regularly be looking to build new relationships and nurture them. That way, when an opportunity comes along you'll already have great relationships or places to turn to get the right referrals. You won't be as successful if you're only invested in these relationships when you need help. You need to invest in them as well.

If you find yourself in a place where you need a referral right away, you could use something that is known as the informational interview technique. This technique is about reaching out to people in the field to get a sense of what they are working on. If you approach in the right way, people can be very generous with their time and offer to help you.

My advice is to target people on networks like LinkedIn and AngelList. Once you've found the people that you're interested in talking to, send them a LinkedIn message or find their email and send them a note.

Here's an example template that you could use:

> *Hi [name],*
>
> *I am very interested in the problems that Google is working on using machine learning. I've been aspiring to break into the field, and being a passionate follower of the Think with Google blog, I regularly read updates on how Google is advancing the frontiers of AI and machine learning.*
>
> *Based on my background in engineering and design, I might be able to help come up with some creative ideas on how to help Google's latest projects.*
>
> *I'd love to take you out to coffee and get a greater sense of what problems you are working on in your role. And perhaps I can help! Would you have some time in the coming weeks?*

Cheers,
[your name]
[your LinkedIn or email]

Take a shot with something like that and see how far you get. One thing to keep in mind: don't get discouraged too easily. It may simply be a numbers game, because people are quite busy. Don't feel personal disappointment if someone doesn't respond to you.

One of the most helpful ways that you can use LinkedIn is to check out your second-degree connections. You might be able to identify some mutual connections that you can mention to the person you're targeting.

If you do meet up with someone for a coffee or an informational interview, do your homework beforehand. Do some deeper research on the company as well as the person you're talking to. That way you can make sure the conversation flows.

Finally, you may also find from the interview that you are not interested in working at that company or on that specific team, which will save you valuable time!

CV vs. LinkedIn

Particularly if you're coming from an academic background, this is an important section for you. When you're looking for a job, people are not necessarily interested in perusing your CV—at least not at first. Perhaps for a highly specific research role, but otherwise you need to present yourself in a more comprehensible form, often through LinkedIn. Even if you have an impressive CV, LinkedIn is a golden standard for recruitment. Having a well-designed and clear LinkedIn profile will allow potential employers and recruiters to discover you, evaluate you, and potentially find you a job opportunity.

Everyone is on LinkedIn. If you're not standing out on LinkedIn, you may already be losing out to other candidates. Resumes are often considered overrated, but I don't believe that LinkedIn profiles are. It's more than just a list of your accomplishments. It should tell a story, and often you using your LinkedIn profile for different activities can often help you be seen.

There are many people I know who have regularly receive job and speaking offers thanks to the content and the comments that they post on LinkedIn regularly. One of my strategies is to follow people who matter in my industry and comment on their posts and like what they do. Then intentionally send them a message after you've established a presence through these engagements.

Those coming from academia tend to value publication. But when searching for these kinds of jobs, it's all about being succinct and talking about the impact and metrics that you've driven within your accomplishments. Recruiters tend to run through these very quickly, so you're going to want to be as concise as possible. Use keywords for the industry. Use meaningful numbers. And don't sell yourself short.

Cover Letters 2.0

The way you were probably taught to do this in school is totally different from how you should actually do it. And it is very important, especially for smaller companies and the hottest startups. They care about who you are as a person and how you'll fit within the culture.

Springboard

Email the hiring manager and include a couple of key paragraphs or some bullets points about your experiences and interests. Keep it short, sweet, and personalized. Highlight why your background would be valuable to their company and what you hope to get out of the experience. That really might be all you need.

Here is a more in-depth guide with some great examples of how to make a cover letter better.

Preparing for an Interview

Once you secure an interview—or several—you'll be invited to start the process, typically on a screening call with a recruiter.

In this section, I will outline what a typical interview process looks like in AI and machine learning careers.

What to Expect

An interview in the machine learning industry typically combines behavioral questions with an array of challenging technical questions. It's important to remember that every company is unique, and hiring processes for different positions could be very distinct. Some companies will want to focus on in-depth, highly technical challenges. Others will focus heavily on culture fit and the behavioral questions. Odds are that you will have a mix of both, so it's good to prepare thoroughly and holistically.

For example, one of the companies that we highlight later on, Integrate.ai, focuses heavily on the behavioral interview, and even if a candidate has the best technical interview, they won't move forward if they aren't a cultural fit.

A typical and thorough process will look like this:

Phone Screen

Phone screens are about filtering out candidates who don't meet the base parameters of the job. They are also about validating whether a candidate's claimed experience is legitimate.

There's a good likelihood that the interview will be conducted by someone from human resources rather than the hiring manager, in order to save time. These interviews are typically more about culture fit and seeing how you work with teams. And it's an important opportunity to showcase your communication skills

During the call, you'll answer questions, but you'll also want to ask your own, such as:

- What kind of problems is the machine learning team facing?
- What are the company's business priorities?
- What are the main values of the company?
- Whom would I report to?
- Are you able to share an expected compensation range for the role?
- What would the interview process look like if I were to move forward?
- If I were to move forward, how should I best prepare for the next stage?

And consider other thoughtful questions that could help you to understand their business and the space that they operate in, as well as the role itself.

Take-Home Assignment

After a phone interview, companies will often give you an assignment and a deadline, usually a few days or a week. This is typically a second screening stage that companies use to ensure that you have a minimum level of technical skills and understanding for the role, as well as some reasoning power and problem-solving ability. It also screens out candidates with commitment issues.

So, what are some examples of take-home assignments? It could be anything from deep analysis on a specific data set to the deconstruction of a machine learning algorithm to demonstrate your understanding of it. It may also be a coding assignment to construct an application. One of your goals in this process should be to see what kind of problems you might be working on.

The important thing to remember is that most interviews consider your approach and problem-solving methods rather than solely focusing on the results or whether you've gotten the correct solutions. It's often OK to fail—if you exemplify some level of creativity and problem-solving ability. If you need to code extensively, write it clearly and comprehensively so that an interviewer can easily follow your thought patterns and understand what you did.

A post on KDNuggets recommends that you:

1. Prepare an ML template with reusable functions.
2. Make an API on top of SciKit-Learn and Matplotlib so that you can quickly perform EDA and build basic models.
3. Consider stacking different models or using one model prediction in the other, in order to raise the eyebrows of an interview.

For case studies, consider reading blogs of major companies like Google, Facebook, Twitter, and others as you can often get a better sense of how these companies tackle business problems with machine learning.

Something to note for take-home assignments is that you'll typically be given a timeline; for example, it may take 5-6 hours and you may have to return it within a couple of days. The goal is of course to keep within those bounds and be honest. Go as far as you can and if you're edging past the expected boundaries, write some quick bullet points about what you would continue to do if you had more time to work on the problem.

Phone Call With the Hiring Manager

If you make it past these first couple of screens—some people have suggested these can remove 50% of candidates—you'll likely be headed to a call with the hiring manager (i.e., the person who is actually hiring you).

The likely focus of this interview will be your technical skills, and it will probably be the last screen and phone call before you go on-site. Often this is split into two or three different components, usually taking place over one long call, but sometimes during three shorter phone calls of 30 minutes each.

Coding

This part of the interview is the most common, especially for a machine learning engineer. You'll likely be evaluated on your ability to solve a coding challenge by presenting pseudocode, or in tougher interviews, compile-ready code. If you're applying for a data position, it will probably be more about asking how to query data with SQL. The questions you will be asked are likely in the programming scripting languages that you said you're experienced in already—that could be Python, Java, Ruby, or whatever language you work in.

Your interviewer may use some sort of online whiteboarding software to evaluate you online, or ask you to share your screen. Alternatively, they may ask you to directly collaborate with them on a text editor and have you type in your solution. Be ready for these scenarios and train with tools like HackerRank or Collabedit if you can.

For coding interviews, there are myriad online resources available, everything from Cracking the coding interview to Interview cake. Take advantage of them.

Mathematics and Statistics

You may have a call that screens for key mathematical and statistical concepts. This is particularly relevant for those applying for data science roles. Web companies will tend to focus on your knowledge of A/B split testing, your understanding of how p-values are calculated, and what statistical significance means. Energy companies may touch more heavily on regression in linear algebra. The key in any of these interviews is to share your entire thought process.

For example, if you're asked about A/B tests, describe the process in detail, emphasize what to watch out for, and voice your experiences running experiments. Treat these questions like mathematical proofs and showcase your ability to statistically reason. And also, don't hesitate to tell a story about why this matters and what insights you could share with a company based on the result.

Qualitative Discussion

The final aspect of the phone call with the hiring manager will focus on how you communicate and whether you'll mesh with the rest of the team. (This could be a separate call from the technical phone screens.) The goal of this call is for the hiring manager to get a feel for who you are: your character, your motivations, your fit with the team, and also a general sense of your intelligence. The goal is to showcase who you are and why you're the right person for the job—not just for your skills, but your personality and traits.

The way to prepare for this part of the interview is to think about the problems the hiring manager is facing and the kind of person they're looking for. They may already have a mental model of what kind of qualities that person has. Your goal here is not to be a chameleon, but you can do some tailoring of your conversation and focusing in on specific traits of yours. Another helpful element to think about is if you would pass "the airplane test." If you were sitting next to each other for several hours on an airplane, would you enjoy the experience?

On-Site Interview With the Manager

If you've made it through the screens, it's now time to meet your hiring manager in person. They'll be judging you from both a technical and non-technical perspective. This will go deeper into evaluating who you are as a candidate.

On-site is also where things get more intense. I've documented a full set of questions below that you might engage with in this interview or in the next several portions.

Technical Challenge

If you don't encounter a technical challenge during your first on-site interview, it's likely that you'll have another one, especially for an engineering role. You could be asked to whiteboard and write down how you would implement certain algorithms or how to think about certain business problems.

Brushing up on your technical skills and knowledge of key terms is important for this. If you don't succeed or at least show potential in the technical portion of the on-site interview, you're unlikely to get the job.

Executive Interview

If you make it through an interview or two with the hiring manager, you're likely near the end of the process, and around that time you'll likely meet executive team members. If it's a startup, this could even be a founder and/or the CEO.

Good job if you've made it this far. Typically, it means you've already "passed" the other portions, and now is the time for final decisions and choosing between star candidates. The key now is to make it clear why you are the best person for the role. Basically, keep doing what you've done, and don't let nerves get to you.

The executive interview usually won't be technical. Odds are, it's about solidifying their choice based on fit and how you get along with the strategic vision of the company. It's also a good opportunity to discuss how you could see yourself growing with the organization, not just how you would fit in for a specific role.

Key Interview Questions, Prep, and Solutions

I have divided this broadly into two categories: behavioral and technical. I've further divided the technical portion into several sub-categories (e.g., algorithms, probability, and more).

One general recommendation I've been given is to review the entirety of chapter five of the MIT Press “Deep Learning” book, which focuses on machine learning basics. You can access it for free here.

Below, I've done my best to pull together tactical examples, proofs, and resources that you can directly apply toward verified questions in machine learning interviews.

Behavioral

These questions are used to evaluate an applicant’s qualitative skills and fit, including past work situations and scenarios, as well as teamwork skills.

Past work

Can you tell me about an AI / machine learning project that you have done in the past?

Intent: The intent of the question is to understand your depth of knowledge and contributions from past experiences. It tests your ability to tell a story around your work and whether you can tie it to impact on the company you worked with.

Springboard

How to answer the question:

■ Try to describe a project that demonstrates both product and engineering experience. For example, if you identified a machine learning model that could solve a business problem, you should explain how these topics furthered company growth.

■ Go into detail about your specific contribution and the outcome from a business goal perspective. The interviewer wants to know what you specifically did while trying to understand the overall goal of the project.

■ Rehearse your experiences many times. This is a very common question, so have two or three go-to projects about which you can get into significant detail.

What have you liked or disliked about your previous position?

Intent: The intent of the question is to identify whether the role you're interviewing for is suitable for you, and to identify why you're moving on from a previous position.

How to answer the question:

■ Understand the role well. If possible, before the interview, use your HR contact to get as much information as possible about the role and its challenges. The HR person can be a treasure trove of information about the role, team, history, and key immediate business goals.

■ Avoid talking about issues you had with specific people, and be professional when talking about what you disliked. Introspect carefully and talk to what makes you passionate. For example, discuss solving a machine learning problem in an actionable way as something you enjoy. You could also talk about learning new technologies that make machine learning applicable across an organization. You might dislike how the organization is not placing AI/ML at the center of its strategy or that the company has had significant attrition at the management level and the direction of the team is unclear. Keep it positive and away from personal situations.

- Bad: “I didn't like that management didn’t have a clue what the company direction was!”

- Good: “I realized I wanted to work in a company where AI and ML is part of its core strategy and where the company has a clear direction.”

Situational

Tell me about a time when you had to convince others to take your position on a specific matter. **What was the outcome?**

Intent: The intent is to find out how good are you at defending your position and your ability to make change within a team.

How to answer the question: Try to find an example where you were successful at making the change and then discuss how the change was quantifiable in its impact. If possible, use a machine learning or AI example. It’s important that you demonstrate your communication and leadership skills here.

I often use [a framework to describe situations and outcomes]() that has its origins at prestigious business consulting firm McKinsey:

The Situation-Complication-Resolution framework

- Situation - the framing of the important, recent context the audience already knows and accepts as fact.
- Complication - the reason the situation requires action.
- Resolution - the action required to solve a problem (or capture an opportunity).

Bottom line is, you want to describe a situation that was happening as normal, a trigger event or complication that messed things up, and then how you pushed the team to come to resolution. This will show that you don't simply react instinctually, but that you think about how to solve problems.

Note: this is as general framework that you can use for many situational questions.

General Project Workflow for ML projects

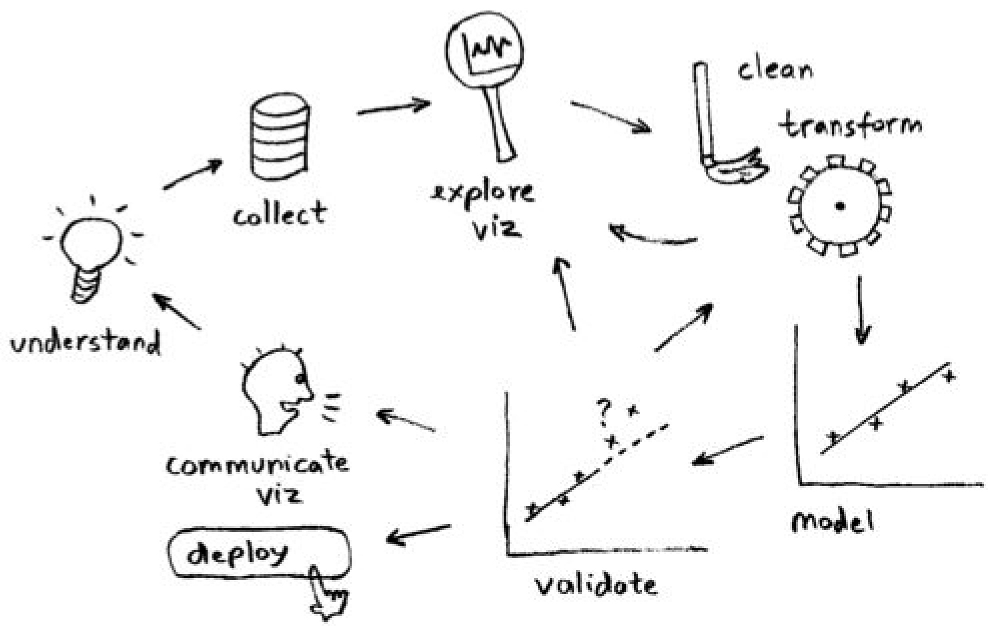
This is one of the most helpful frameworks that I've seen on how to approach an ML project in general. Here it is:

- Specify the business objective. Are we trying to win more customers, achieve higher satisfaction, or gain more revenue?
- Define the problem. What is the specific gap in your ideal world and the real one that requires machine learning to fill? Ask questions that can be addressed using your data and predictive modeling (ML algorithms).
- Create a common-sense baseline. But before you resort to ML, set up a baseline to solve the problem as if you know zero data science. You may be amazed at how effective this baseline is. It can be as simple as recommending the top N popular items or some other rule-based

logic. This baseline can also serve as a good benchmark for ML algorithms.

- Review ML literature. To avoid reinventing the wheel and get inspired on what techniques / algorithms are good at addressing the questions using our data.
- Set up a single-number metric. What does it mean to be successful (high accuracy, lower error, or bigger AUC?) and how do you measure it? The metric has to align with high-level goals. Set up a single-number against which all models are measured.
- Do exploratory data analysis. Play with the data to get a general idea of data types, distribution, variable correlation, facets, etc. This step would involve a lot of plotting.
- Partition data. The validation set should be large enough to detect differences between the models you're training; the test set should be large enough to indicate the overall performance of the final model; for the training set, needless to say, the larger the merrier.
- Preprocess. This would include data integration, cleaning, transformation, reduction, discretization, and more.
- Engineer features. Coming up with features is difficult, time-consuming, and requires expert knowledge. Applied machine learning is basically feature engineering. This step usually involves feature selection and creation, using domain knowledge. It can be minimal for deep learning projects.
- Develop models. Choose which algorithm to use, what hyperparameters to tune, which architecture to use, etc.
- Ensemble. Ensemble methods can usually boost performance, depending on the correlations of the models/features. So it's always a good idea to try out. But be open-minded about making tradeoffs—some ensemble methods are too complex/slow to put into production.

- Deploy models. Deploy models into production for inference.
- Monitor models. Monitor model performance and collect feedback.
- Iterate. Iterate the previous steps. Data science tends to be an iterative process, with new and improved models being developed over time.



Technical Questions for Machine Learning Interviews

There's a plethora of technical questions that you can encounter in a machine learning interview and they could vary greatly based on the role and the company. Your interview could include algorithms and theory, mathematics and probability, your programming skills and application of theory into code, and ultimately your general understanding of AI and machine learning. Because the field is moving so fast, it's critical to stay on top of the latest trends. There will also likely be company- or industry-specific questions that will challenge you to apply your general knowledge into actionable insights for the business.

Note: I directly quote verbatim questions and solutions from many sources and have done my best to give attribution and links everywhere that I can. I exclude quotations to keep it streamlined. This is a compilation of the excellent work of others, and I've gathered the best and most concise solutions possible.

These include many parts of Springboard's machine learning questions list by Roger Huang, as well as these key sites:

Data Science Q & A

Cracking The Machine Learning Interview (this is the longest set of questions that I've seen)

Popular Machine Learning Interview Questions to Assess Candidates

Other specific sources are listed inline within the sections.

Mathematical Skills

Linear Algebra

What are scalars, vectors, matrices, and tensors?

More reading and solutions from Quantstart

The two primary mathematical entities that are of interest in linear algebra are the vector and the matrix. They are examples of a more general entity known as a tensor. Tensors possess an order (or rank), which determines the number of dimensions in an array required to represent it. Scalars are single numbers and are an example of a 0th-order tensor. Vectors are ordered arrays of single numbers and are an example of 1st-order tensor. Vectors are members of objects known as vector spaces. Matrices are rectangular arrays consisting of numbers and are an example of 2nd-order tensors.

What is Hadamard product of two matrices?

More reading and solution from [Medium]

Hadamard product of two vectors is very similar to [matrix addition]; elements corresponding to the same row and columns of given vectors/matrices are multiplied together to form a new vector/matrix.

$$\vec{g} \circ \vec{h} \circ \vec{m}$$

Hadamard product of vector g, h, and m

The order of matrices/vectors to be multiplied should be the same and the resulting matrix will also be of the same order.

$$\overset{G}{\begin{bmatrix} 3 & 5 & 7 \\ 4 & 9 & 8 \end{bmatrix}} \circ \overset{H}{\begin{bmatrix} 1 & 6 & 3 \\ 0 & 2 & 9 \end{bmatrix}} = \overset{N}{\begin{bmatrix} 3\times 1 & 5\times 6 & 7\times 3 \\ 4\times 0 & 9\times 2 & 8\times 9 \end{bmatrix}}$$

Hadamard product of Matrix G and Matrix H (both of order 2x3), gives another Matrix N

$$N = \begin{bmatrix} 3 & 30 & 21 \\ 0 & 18 & 72 \end{bmatrix}$$

Matrix N is of the same order as input matrices (2x3)

Hadamard product is used in image compression techniques such as JPEG.

What is broadcasting in connection to linear algebra?

Solution from Machine Learning Mastery

Broadcasting is the name given to the method that NumPy uses to allow array arithmetic between arrays with a different shape or size.

Although the technique was developed for NumPy, it has also been adopted more broadly in other numerical computational libraries, such as Theano, TensorFlow, and Octave.

Broadcasting solves the problem of arithmetic between arrays of different shapes by, in effect, replicating the smaller array along the last mismatched dimension.

The term broadcasting describes how numpy treats arrays with different shapes during arithmetic operations. Subject to certain constraints, the smaller array is "broadcast" across the larger array so that they have compatible shapes.

— Broadcasting, SciPy.org

NumPy does not actually duplicate the smaller array; instead, it makes memory and computationally efficient use of existing structures in memory that in effect achieve the same result.

The concept has also permeated linear algebra notation to simplify the explanation of simple operations.

In the context of deep learning, we also use some less conventional notation. We allow the addition of matrix and a vector, yielding another matrix: C = A + b, where Ci,j = Ai,j + bj. In other words, the vector b is added to each row of the matrix. This shorthand eliminates the need to define a matrix with b copied into each row before doing the addition. This implicit copying of b to many locations is called broadcasting.

— Page 34, Deep Learning, 2016.

A quick toolbox and reference for linear algebra

More on linear algebra for machine learning (includes resources and refreshers)

More on Hadamard product

Linear and Logistic Regression

More on linear regression and logistic regression

What is linear regression and logistic regression?

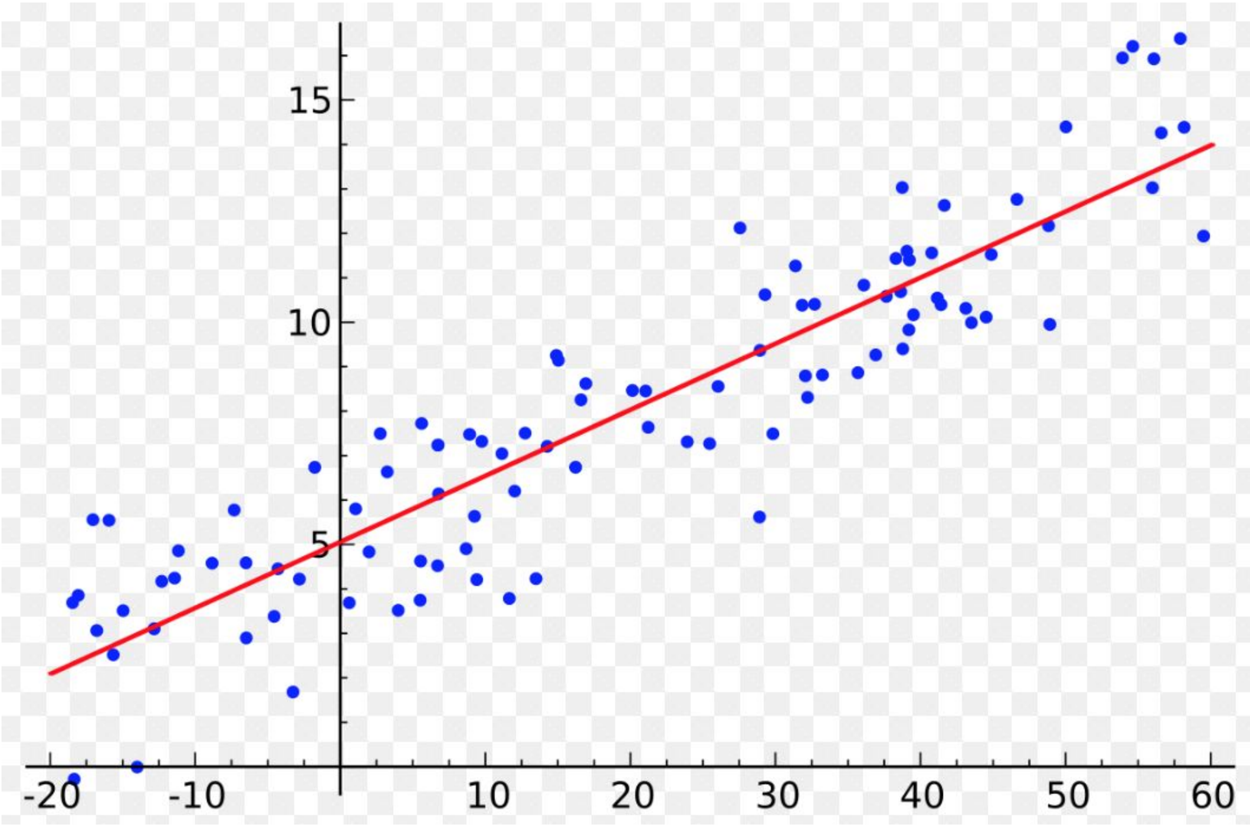
Linear and logistic regression are commonly used for ML algorithms. You can expect at least one of these questions in an interview.

A linear regression models the relationship between the dependent variable Y and the independent variable X.

A logistic regression models a binary dependent variable.

Analyze a data set and give a model that can predict this response variable.

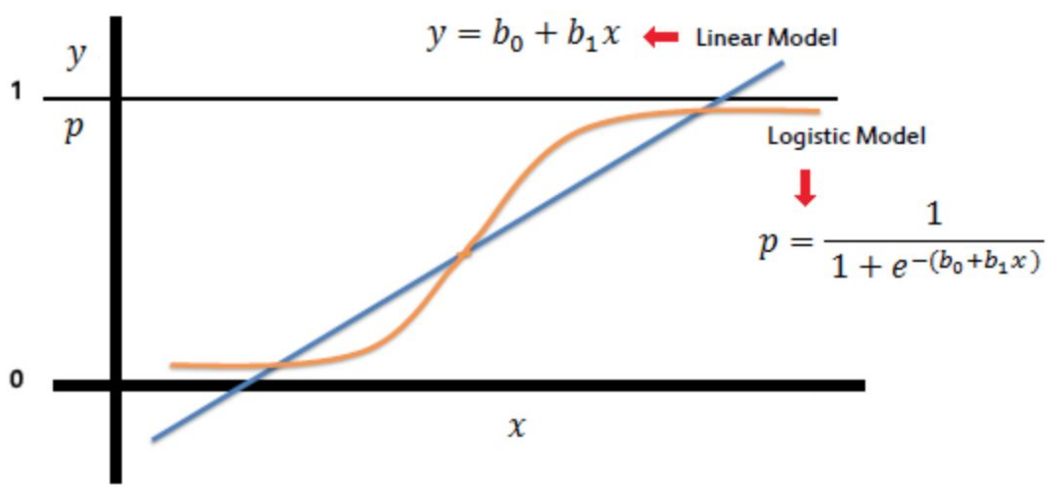
Example from [Vimarsh Karbhari]



Source: Wikipedia

The cost of one pen is x$. The cost of ten pens is 10x$. This is the most classic layman's form of linear regression. The simplest form of the regression equation with one dependent and one independent variable is defined by the formula y = c + b*x, where y = estimated dependent variable score, c = constant, b = regression coefficient, and x = score on the independent variable. In our pen example, c=0, y is the cost of pens and x is the number of pens. If we know the unit cost of one pen b we can calculate the cost of any number of pens. A complex form of linear regression is used in housing price predictions.

For any scenario-based problem in an interview, it is an easy mistake to start with a complex ML algorithm. Most interviewees make the mistake of starting with something that the problem resembles. They may start with neural networks or SVMs. ALWAYS start with linear/logistic regression if possible. This helps you level set on the most basic benchmark performance for the solution. Approach that question like a [programming interview]() where you start with a benchmark and you proceed to a more optimized solution.



Source: [Logistic regression]()

Springboard

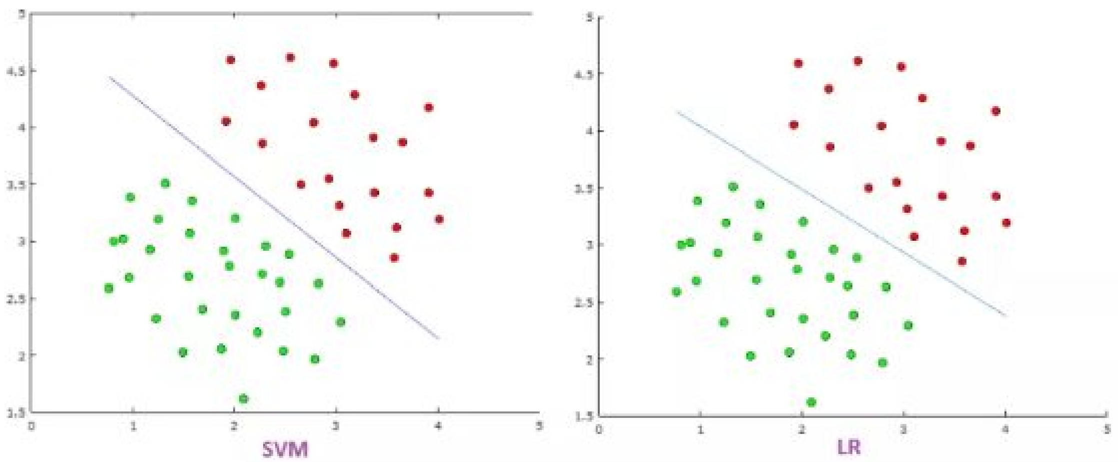
Linear regression is used for continuous targets while logistic regression is used for binary targets as sigmoid curve in the logistic model forces the features to either a 0 or 1.

Support Vector Machine

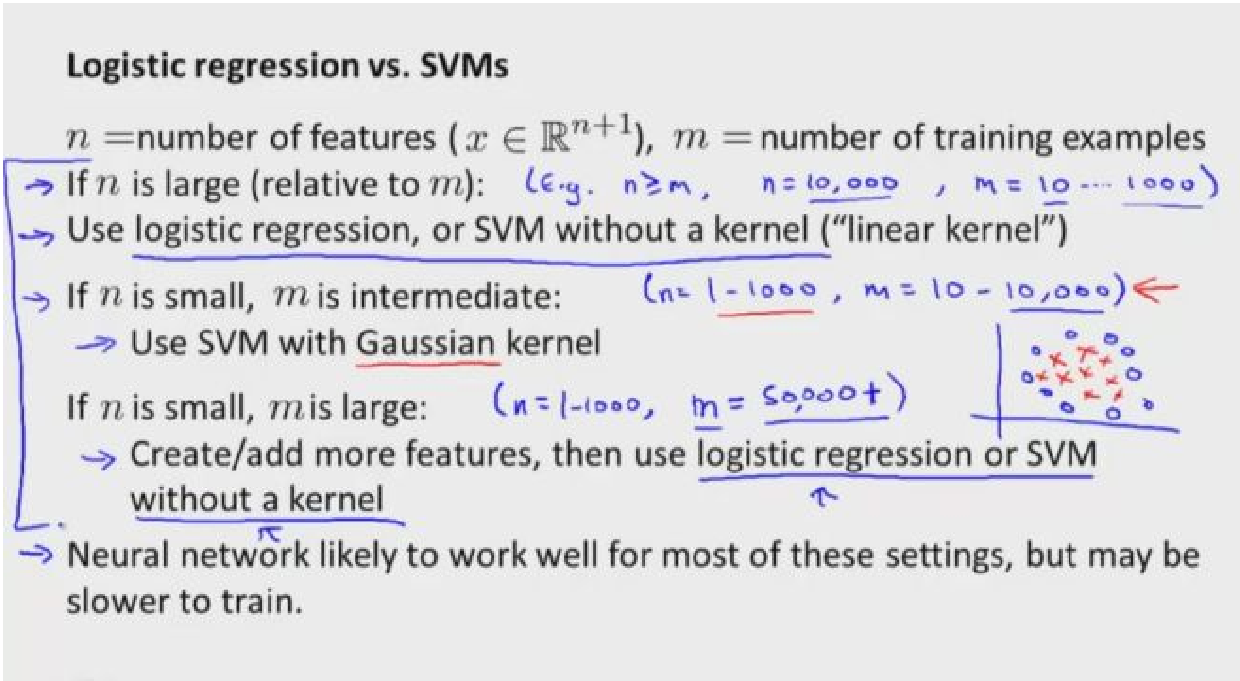
Logistic regression vs. SVMs: When to use which one?

Solution from [Towards Data Science]

SVM tries to maximize the margin between the closest support vectors while LR the posterior class probability. Thus, SVM finds a solution which is as fair as possible for the two categories while LR does not have this property.



Try logistic regression first, because it is simpler. If logistic regression fails and you have reason to believe your data won't be linearly separable, try an SVM with a non-linear kernel like a Radial Basis Function (RBF).



More on [support vector machines vs. logistic regression]

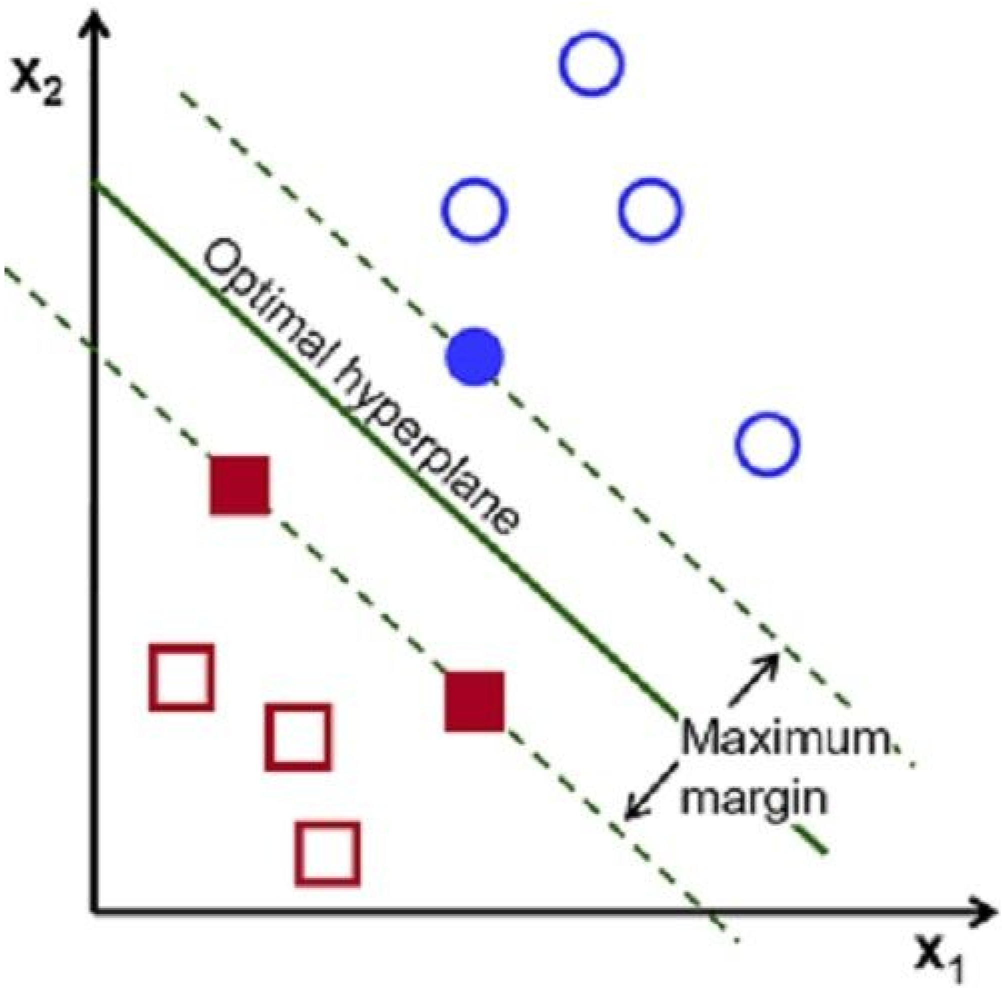
How can the SVM optimization function be derived from the logistic regression optimization function?

This one is a bit too long to include in its entirety here, so check out [this resource], particularly slides 5 and 10.

More on [Demystifying the math of support vector machines]

What is a large margin classifier?

Solution from [Quora]



- SVM is a type of classifier which classifies positive and negative examples, here blue and red data points.
- As shown in the image, the largest margin is found in order to avoid overfitting. That is, the optimal hyperplane is at the maximum distance from the positive and negative examples (equally distant from the boundary lines).

- To satisfy this constraint, and also to classify the data points accurately, the margin is maximized, that is why this is called the large margin classifier.

Numerical Optimization

According to Jason Brownlee, the core of machine learning models are an optimization problem. Each is really a "search" for terms with unknown values needed to fill an equation.

Ordinary Least Square and Maximum Likelihood

A solution from Zarantech describes both the ordinary least square and maximum likelihood methods to reaching these values.

OLS is to linear regression. Maximum likelihood is to logistic regression. Explain the statement.

Answer: OLS and maximum likelihood are the methods used by the respective regression methods to approximate the unknown parameter (coefficient) value. In simple words, ordinary least square is a method used in linear regression which approximates the parameters resulting in minimum distance between actual and predicted values. Maximum likelihood helps in choosing the values of parameters which maximizes the likelihood that the parameters are most likely to produce observed data.

Maximum likelihood

More on optimization

More on considering numerical problems as a search

What is underflow and overflow?

From Quora (Richard Urwin)

Overflow is when the absolute value of the number is too high for the computer to represent it. Underflow is when the absolute value of the number is too close to zero for the computer to represent it.

You can get overflow with both integers and floating point numbers. You can only get underflow with floating point numbers.

To get an overflow, repeatedly multiply a number by ten. To get an underflow, repeatedly divide it by ten.

If the variable x is a signed byte it can have values in the range -128 to +127, then

1. x = 127
2. x = x + 1

will result in an overflow. +128 is not a valid value for x.

For floating point numbers, the range depends on their representation. If x is a single precision (32-bit IEEE) number, then

1. x = 1e-38
2. x = x / 1000

will result in an underflow. 1e-42 is not a valid value for x.

Statistics & Probability

What is Bayes' Theorem? How is it useful in a machine learning context?

More reading: [An Intuitive (and Short) Explanation of Bayes' Theorem (BetterExplained)]

Bayes' Theorem gives you the posterior probability of an event given what is known as prior knowledge.

Mathematically, it's expressed as the true positive rate of a condition sample divided by the sum of the false positive rate of the population and the true positive rate of a condition. Say you had a 60% chance of actually having the flu after a flu test, but out of people who had the flu, the test will be false 50% of the time, and the overall population only has a 5% chance of having the flu. Would you actually have a 60% chance of having the flu after having a positive test?

Bayes' Theorem says no. It says that you have a (.6 * 0.05) (True Positive Rate of a Condition Sample) / (.6*0.05) (True Positive Rate of a Condition Sample) + (.5*0.95) (False Positive Rate of a Population) = 0.0594 or 5.94% chance of getting the flu.

Bayes' Theorem is the basis behind a branch of machine learning that most notably includes the Naive Bayes classifier. That's important to consider when you're faced with machine learning interview questions.

What is the difference between Type I error and Type II error?

More reading: [Type I and type II errors (Wikipedia)]

Type I error is what is referred to as a "false positive," or the incorrect rejection of the null hypothesis. Type II error is what is referred to as a "false negative," or the incorrect acceptance of the null hypothesis. So, effectively: Type I error means claiming something has happened when it hasn't, while

Type II error means that you claim nothing is happening when in fact something is.

You may want to communicate your grasp of the concepts with an example and how it might be relevant to the business at hand. Type I error, or a false positive, would be telling a man he was pregnant, while Type II error would be telling a pregnant woman she wasn't.

If you were running a fraud detection business, you might have a very high tolerance for false positives (a client will not fuss about an email on the potential of fraud), but a false negative (not detecting fraud when it is happening) could be disastrous.

Confidence Intervals

What are the population mean and the sample mean?

Solution from [Key Differences]

The sample mean is an average of a random sample derived of the population. Population mean is the average of an entire group.

The sample mean is calculated as under:

$$Sample\ Mean\ \bar{x} = \frac{1}{n} \sum_{i}^{n} = 1^{a_i}$$

where, n = Size of sample

∑ = Add up

ai = All the observations

population mean can be calculated as:

$$Population\ Mean\ \mu\ = \frac{1}{N}\sum_{i}^{N} = 1^{a_i}$$

where N = Size of the population

∑ = Add up

ai = All the observations

What is population standard deviation and sample standard deviation?

Solution from [Khan Academy]

Standard deviation measures the spread of a data distribution. It measures the typical distance between each data point and the mean.

The formula we use for standard deviation depends on whether the data is being considered a population of its own, or the data is a sample representing a larger population.

- If the data is being considered a population on its own, we divide by the number of data points, N
- If the data is a sample from a larger population, we divide by one fewer than the number of data points in the sample, n-1

Population standard deviation:

$$\sigma = N\sum(xi{-}\mu)2$$

Sample standard deviation:

$$sx = n-1\sum(xi-\bar{x})2$$

The steps in each formula are all the same except for one—we divide by one less than the number of data points when dealing with sample data.

For step-by-step solutions, read the rest of the article here.

Probability Distribution Type of Variables

What is a probability distribution?

Answer from Towards Data Science

Probability distributions provide the likelihood for all possible values of a given process. It describes how likely it is that a single observation of a random variable is equal to a particular value or range of values. In other words, for any given random process there is both a range of values that are possible and a likelihood that a single draw from the random process will take on one of those values.

Check out this in-depth article on probability distributions here.

What is a probability mass function?

Solution from StatisticsHowTo

A probability mass function (PMF)— also called a frequency function— gives you probabilities for discrete random variables. "Random variables" are variables from experiments like dice rolls, choosing a number out of a hat, or getting a high score on a test. The “discrete” part means that there’s a set number of outcomes. For example, you can only roll a 1,2,3,4,5, or 6 on a die.

A PMF equation looks like this:

P(X = x)

That just means “the probability that X takes on some value x.”

It’s not a very useful equation on its own; what’s more useful is an equation that tells you the probability of some individual event happening. For example:

P(X=1) = 0.2 * 0.2.

How you come up with these equations depends mostly on what type of event you have. For example, the binomial distribution PMF is:

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

And the Poisson distribution PMF is:

$$\Pr\{Y = k \mid \mu\} = \frac{e^{-\mu}\mu^k}{k!} \qquad \text{for } k = 0, 1, 2, \ldots.$$

Full solution and explanation here.

What is a probability density function?

Solution from Penn State; more reading from Wikipedia

The probability density function is used to specify the probability of the random variable falling within a particular range of values, as opposed to taking on any one value.

To find the probability that X falls in an interval (a, b) you need to find $P(a < X < b)$.

Continuous random variable X with support S is an integrable function f(x) satisfying the following:

(1) f(x) is positive everywhere in the support S, that is, $f(x) > 0$, for all x in S

(2) The area under the curve f(x) in the support S is 1, that is:

$$\int S f(x) dx = 1$$

(3) If f(x) is the p.d.f. of x, then the probability that x belongs to A, where A is some interval, is given by the integral of f(x) over that interval, that is:

$$P(X \in A) = \int A f(x) dx$$

Full example and solution here.

Autoencoders

What is an autoencoder? What does it "auto-encode"?

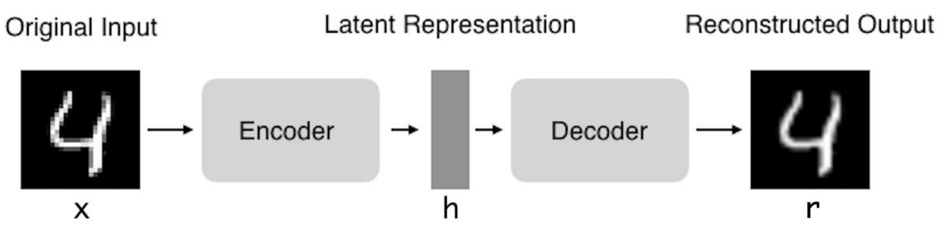
Solution from Towards Data Science

Autoencoders (AE) are neural networks that aim to copy their inputs to their outputs. They work by compressing the input into a latent-space representation, and then reconstructing the output from this representation.

Autoencoders are learned automatically from data examples. It means that it is easy to train specialized instances of the algorithm that will perform well on a specific type of input and that it does not require any new engineering, only the appropriate training data.

This kind of network is composed of two parts:

1. Encoder: This is the part of the network that compresses the input into a latent-space representation. It can be represented by an encoding function h=f(x).
2. Decoder: This part aims to reconstruct the input from the latent space representation. It can be represented by a decoding function r=g(h).



Architecture of an autoencoder

The autoencoder as a whole can thus be described by the function g(f(x)) = r where you want r as close as the original input x.

More reading on Autoencoders.

Programming Skills

Pick an algorithm and write the pseudocode for a parallel implementation.

More reading: Writing pseudo-code for parallel programming (Stack Overflow)

This kind of question demonstrates your ability to think in parallelism and how you could handle concurrency in programming implementations dealing with big data. Take a look at pseudo-code frameworks such as Peril-L and visualization tools such as Web Sequence Diagrams to help you demonstrate your ability to write code that reflects parallelism.

What are some differences between a linked list and an array?

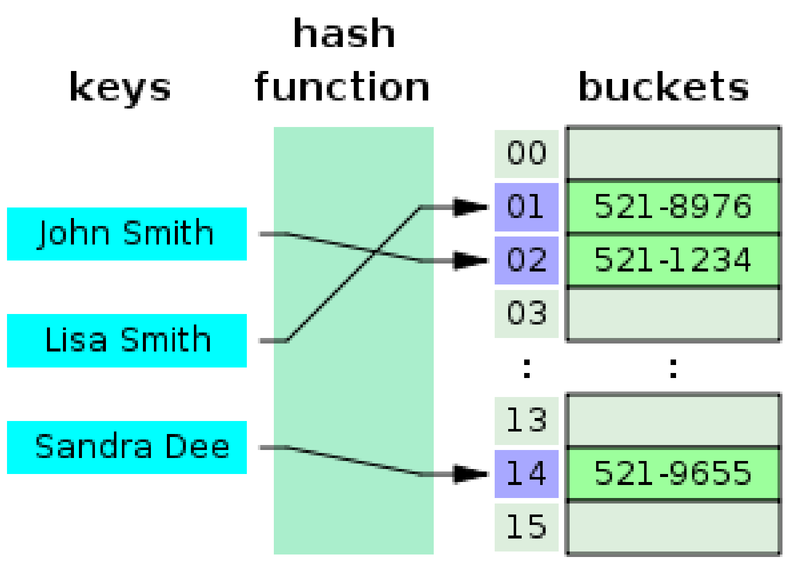
More reading: Array versus linked list (Stack Overflow)

An array is an ordered collection of objects. A linked list is a series of objects with pointers that direct how to process them sequentially. An array assumes that every element has the same size, unlike the linked list. A linked list can more easily grow organically: an array has to be pre-defined or redefined for organic growth. Shuffling a linked list involves changing which points direct where — meanwhile, shuffling an array is more complex and takes more memory.

Describe a hash table.

More reading: [Hash table (Wikipedia)]

A hash table is a data structure that produces an associative array. A key is mapped to certain values through the use of a hash function. They are often used for tasks such as database indexing.

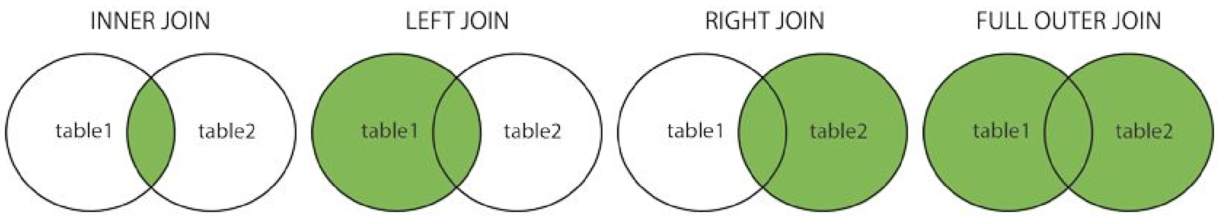


SQL

Although you don't have to be a SQL expert for most machine learning positions—it is more common for a data scientist role—definitely some SQL-related questions could come up.

Here's a good and quick refresher on [the different types of joins]:

- (INNER) JOIN: Return records that have matching values in both tables
- LEFT (OUTER) JOIN: Return all records from the left table, and the matched records from the right table
- RIGHT (OUTER) JOIN: Return all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN: Return all records when there is a match in either left or right table



Other SQL resources to do a refresher:

1. W3schools SQL
2. SQLZOO

How would you implement a recommendation system for our company's users?

More reading: How to Implement A Recommendation System? (Stack Overflow)

A lot of machine learning interview questions of this type will involve implementation of machine learning models to a company's problems. You'll have to research the company and its industry in-depth, especially the revenue drivers the company has, and the types of users the company takes on in the context of the industry it's in.

Algorithms and Learning Theory

What's the trade-off between bias and variance?

More reading: [Bias-Variance Tradeoff (Wikipedia)]

Bias is error due to erroneous or overly simplistic assumptions in the learning algorithm you're using. This can lead to the model underfitting your data, making it hard for it to have high predictive accuracy and for you to generalize your knowledge from the training set to the test set.

Variance is error due to too much complexity in the learning algorithm you're using. This leads to the algorithm being highly sensitive to high degrees of variation in your training data, which can lead your model to overfit the data. You'll be carrying too much noise from your training data for your model to be very useful for your test data.

The bias-variance decomposition essentially decomposes the learning error from any algorithm by adding the bias, variance, and a bit of irreducible error due to noise in the underlying data set. Essentially, if you make the model more complex and add more variables, you'll lose bias but gain some variance—in order to get the optimally reduced amount of error, you'll have to trade off bias and variance. You don't want high bias or high variance in your model.

What is the difference between supervised and unsupervised machine learning?

More reading: [What is the difference between supervised and unsupervised machine learning? (Quora)]()

Supervised learning requires labeled training data. For example, in order to do classification (a supervised learning task), you'll need to first label the data you'll use to train the model to classify data into your labeled groups. Unsupervised learning, in contrast, does not require labeling data explicitly.

How is KNN different from k-means clustering?

More reading: [How is the k-nearest neighbor algorithm different from k-means clustering? (Quora)]()

K-nearest neighbors is a supervised classification algorithm, while k-means clustering is an unsupervised clustering algorithm. While the mechanisms may seem similar at first, what this really means is that in order for k-nearest neighbors to work, you need labeled data you want to classify an unlabeled point into (thus the nearest neighbor part). K-means clustering requires only a set of unlabeled points and a threshold: the algorithm will take unlabeled points and gradually learn how to cluster them into groups by computing the mean of the distance between different points.

The critical difference here is that KNN needs labeled points and is thus supervised learning, while k-means doesn't and is thus unsupervised learning.

What's your favorite algorithm, and can you explain it to me in less than a minute?

This type of question tests your understanding of how to communicate complex and technical nuances with poise and your ability to summarize quickly and efficiently. Make sure you have an answer and can explain different algorithms so simply and effectively that a five-year-old could grasp the basics!

What's a Fourier transform?

More reading: Fourier transform (Wikipedia)

A Fourier transform is a generic method to decompose generic functions into a superposition of symmetric functions. Or as this more intuitive tutorial puts it, given a smoothie, it's how we find the recipe. The Fourier transform finds the set of cycle speeds, amplitudes, and phases to match any time signal. A Fourier transform converts a signal from time to frequency domain—it's a very common way to extract features from audio signals or other time series such as sensor data.

Data Sets and Big Data

What cross-validation technique would you use on a time series data set?

More reading: Using k-fold cross-validation for time-series model selection (CrossValidated)

Instead of using standard k-folds cross-validation, you have to pay attention to the fact that a time series is not randomly distributed data—it is inherently ordered by chronological order. If a pattern emerges in later time periods, your model may still pick up on it even if that effect doesn't hold in earlier years!

You'll want to do something like forward chaining, where you'll be able to model on past data then look at forward-facing data.

fold 1 : training [1], test [2]

fold 2 : training [1 2], test [3]

fold 3 : training [1 2 3], test [4]

fold 4 : training [1 2 3 4], test [5]

fold 5 : training [1 2 3 4 5], test [6]

How is a decision tree pruned?

More reading: [Pruning (decision trees)]

Pruning is what happens in decision trees when branches that have weak predictive power are removed in order to reduce the complexity of the model and increase the predictive accuracy of a decision tree model. Pruning can happen bottom-up and top-down, with approaches such as reduced error pruning and cost complexity pruning.

Reduced error pruning is perhaps the simplest version: replace each node. If it doesn't decrease predictive accuracy, keep it pruned. While simple, this heuristic actually comes pretty close to an approach that would optimize for maximum accuracy.

How do you handle missing or corrupted data in a data set?

More reading: [Handling missing data (O'Reilly)]

You could find missing/corrupted data in a data set and either drop those rows or columns, or decide to replace them with another value.

In Pandas, there are two very useful methods that will help you find columns of data with missing or corrupted data and drop those values: isnull() and dropna(). If you want to fill the invalid values with a placeholder value (for example, 0), you could use the fillna() method.

How would you handle an imbalanced data set?

More reading: [8 Tactics to Combat Imbalanced Classes in Your Machine Learning data set (Machine Learning Mastery)]

An imbalanced data set is when you have, for example, a classification test and 90% of the data is in one class. That leads to problems: an accuracy of 90% can be skewed if you have no predictive power on the other category of data! Here are a few tactics to get over the hump:

1- Collect more data to even the imbalances in the data set.

2- Resample the data set to correct for imbalances.

3- Try a different algorithm altogether on your data set.

What's important here is that you have a keen sense for what damage an unbalanced data set can cause, and how to balance that.

Do you have experience with Spark or big data tools for machine learning?

More reading: [50 Top Open Source Tools for Big Data (Datamation)]

You'll want to get familiar with the meaning of big data for different companies and the different tools they'll want. Spark is the big data tool most in demand now, able to handle immense data sets with speed. Be honest if you don't have experience with the tools demanded, but also take a look at job descriptions and see what tools pop up: you'll want to invest in familiarizing yourself with them.

Model and Feature Selection

What's the difference between a generative and discriminative model?

More reading: [What is the difference between a generative and discriminative algorithm? (Stack Overflow)]

A generative model will learn categories of data while a discriminative model will simply learn the distinction between different categories of data. Discriminative models will generally outperform generative models on classification tasks.

Which is more important to you: model accuracy or model performance?

More reading: [Accuracy paradox (Wikipedia)]

This question tests your grasp of the nuances of machine learning model performance! Machine learning interview questions often look toward the details. There are models with higher accuracy that can perform worse in predictive power— how does that make sense?

Well, it has everything to do with how model accuracy is only a subset of model performance, and at that, a sometimes misleading one. For example, if you wanted to detect fraud in a massive data set with a sample of millions, a more accurate model would most likely predict no fraud at all if only a vast minority of cases were fraud. However, this would be useless for a predictive model—a model designed to find fraud that asserted there was no fraud at all! Questions like this help you demonstrate that you understand model accuracy isn't the be-all and end-all of model performance.

What's the F1 score? How would you use it?

More reading: F1 score (Wikipedia)

The F1 score is a measure of a model's performance. It is a weighted average of the precision and recall of a model, with results tending to 1 being the best, and those tending to 0 being the worst. You would use it in classification tests where true negatives don't matter much.

When should you use classification over regression?

More reading: Regression vs Classification (Math StackExchange)

Classification produces discrete values and data set to strict categories, while regression gives you continuous results that allow you to better distinguish differences between individual points. You would use classification over regression if you wanted your results to reflect the belongingness of data points in your data set to certain explicit categories (e.g., if you wanted to know whether a name was male or female rather than just how correlated they were with male and female names).

Name an example where ensemble techniques might be useful.

More reading: [Ensemble learning (Wikipedia)]

Ensemble techniques use a combination of learning algorithms to optimize better predictive performance. They typically reduce overfitting in models and make the model more robust (unlikely to be influenced by small changes in the training data).

You could list some examples of ensemble methods, from bagging to boosting to a "bucket of models" method and demonstrate how they could increase predictive power.

How do you ensure you're not overfitting with a model?

More reading: [How can I avoid overfitting? (Quora)]

This is a simple restatement of a fundamental problem in machine learning:

the possibility of overfitting the training data and carrying the noise of that data through to the test set, thereby providing inaccurate generalizations.

There are three main methods to avoid overfitting:

1. Keep the model simple: reduce variance by taking into account fewer variables and parameters, thereby removing some of the noise in the training data.
2. Use cross-validation techniques such as k-folds cross-validation.
3. Use regularization techniques such as LASSO that penalize certain model parameters if they're likely to cause overfitting.

What evaluation approaches would you work to gauge the effectiveness of a machine learning model?

More reading: How to Evaluate Machine Learning Algorithms (Machine Learning Mastery)

You would first split the data set into training and test sets, or perhaps use cross-validation techniques to further segment the data set into composite sets of training and test sets within the data. You should then implement a choice selection of performance metrics—here is a fairly comprehensive list. You could use measures such as the F1 score, accuracy, and the confusion matrix. What's important here is to demonstrate that you understand the nuances of how a model is measured and how to choose the right performance measures for the right situations.

How would you evaluate a logistic regression model?

More reading: Evaluating a logistic regression (CrossValidated), Logistic Regression in Plain English

A subsection of the question above. You have to demonstrate an understanding of what the typical goals of a logistic regression are (classification, prediction, etc.) and bring up a few examples and use cases.

Deep Learning

What is deep learning, and how does it contrast with other machine learning algorithms?

More reading: Deep learning (Wikipedia)

Deep learning is a subset of machine learning that is concerned with neural networks: how to use backpropagation and certain principles from neuroscience to more accurately model large sets of unlabeled or semi-structured data. In that sense, deep learning represents an unsupervised learning algorithm that learns representations of data through the use of neural nets.

Why does deep learning matter? Why is it useful?

Solution from: Forbes

Deep learning networks avoid the drawback of machine learning because they excel at unsupervised learning. The key difference between supervised and unsupervised learning is that data are not labeled in unsupervised learning. For example, if you were building an image recognition network, even though the pictures of cats don't come with the label "cat," deep learning networks could still learn to identify the cats.

You can imagine that the ability to learn from unlabeled or unstructured data is an enormous benefit for applications in the real world. It allows you to create systems that can learn from a chaotic and spontaneous world, with many different inputs.

More reading on deep learning.

What kinds of architectures are there?

Solution from Deep Learning

Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks.

One of the most popular types of deep neural networks is known as convolutional neural networks (CNN or ConvNet). A CNN involves learned features with input data, and uses 2D convolutional layers, making this architecture well suited to processing 2D data, such as images.

CNNs eliminate the need for manual feature extraction, so you do not need to identify features used to classify images. The CNN works by extracting features directly from images. The relevant features are not pretrained; they are learned while the network trains on a collection of images. This automated feature extraction makes deep learning models highly accurate for computer vision tasks such as object classification.

Other networks include:

- Unsupervised Pretrained Networks (UPNs)
- Recurrent Neural Networks
- Recursive Neural Networks

More on deep learning architectures; more on neural networks.

How would you develop an image recognition network?

There is a great full solution for this here.

At a high level, it looks like:

- Collecting the data set
- Importing libraries and splitting the data set
- Building the CNN
- Full connection
- Data augmentation
- Training our network
- Testing

But it's best that you take a look at the solution.

How would you develop a time series network?

I found another long, incredibly comprehensive solution here for this on Machine Learning Mastery.

It goes through:

- How to develop CNN models for univariate time series forecasting.
- How to develop CNN models for multivariate time series forecasting.
- How to develop CNN models for multi-step time series forecasting.

The same author also wrote a great guide on building a time series model more generally.

Regularization

Explain the difference between L1 and L2 regularization.

More reading: [What is the difference between L1 and L2 regularization? (Quora)]

L2 regularization tends to spread error among all the terms, while L1 is more binary/sparse, with many variables either being assigned a 1 or a 0 in weighting. L1 corresponds to setting a Laplacean prior on the terms, while L2 corresponds to a Gaussian prior.

What is dropout?
From Machine Learning Mastery

Dropout is a regularization technique patented by Google for reducing overfitting in neural networks by preventing complex co-adaptations on training data.

A single model can be used to simulate having a large number of different network architectures by randomly dropping out nodes during training. It very computationally cheap and is a remarkably effective regularization method to reduce overfitting and improve generalization error in deep neural networks of all kinds.

Clustering

What is distortion function? Is it convex or not-convex?

Solution from Towards Data Science

The k-means clustering distortion function measures how well the data fits a cluster and is computed as a simple sum of squared distances:

$$I_j = \sum_{t=1}^{N_j} [d(x_{jt}, w_j)]^2$$

Single cluster distortion

For a given cluster j, we add the squared distances from all the cluster points x to the cluster center w. The total distortion is just the sum of all distortions for a given value of K:

$$S_K = \sum_{j=1}^{K} I_j$$

Total distortion for K

The scree plot shows how the total distortion changes as we increase K. At the limit, when K equals the number of samples in the data set, when every point in the data set corresponds to its own cluster, the total distortion is zero.

This function is not-convex.

More reading here; more on convex vs not-convex and proof here.

Describe the EM algorithm

From Data Science Central

The EM algorithm finds maximum-likelihood estimates for model parameters when you have incomplete data. The "E-Step" finds probabilities for the assignment of data points, based on a set of hypothesized probability density functions; The "M-Step" updates the original hypothesis with new data. The cycle repeats until the parameters stabilize.

More reading here.

Natural Language Processing (NLP)

What is WORD2VEC?

Solution from Skymind.ai

WORD2VEC is a two-layer neural net that processes text. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus. While Word2vec is not a deep neural network, it turns text into a numerical form that deep nets can understand. Deeplearning4j implements a distributed form of Word2vec for Java and Scala, which works on Spark with GPUs.

Word2vec's applications extend beyond parsing sentences in the wild. It can be applied just as well to genes, code, likes, playlists, social media graphs, and other verbal or symbolic series in which patterns may be discerned.

More fantastic reading and in-depth on WORD2VEC here.

What is t-SNE? Why would you use PCA instead of t-SNE?

Solution from DataCamp

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional data sets. It is extensively applied in image processing, NLP, genomic data, and speech processing.

How it works:

- The algorithm starts by calculating the probability of similarity of points in high-dimensional space and calculating the probability of similarity of points in the corresponding low-dimensional space. The similarity of points is calculated as the conditional probability that a point A would choose point B as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian (normal distribution) centered at A.
- It then tries to minimize the difference between these conditional probabilities (or similarities) in higher-dimensional and lower-dimensional space for a perfect representation of data points in lower-dimensional space.
- To measure the minimization of the sum of difference of conditional probability, t-SNE minimizes the sum of Kullback-Leibler divergence of overall data points using a gradient descent method.

More reading here.

Springboard

Loss Optimization

Name some typical loss functions used for regression. Compare and contrast.

Solution from Heartbeat of Fritz.ai

Mean square error, quadratic loss, L2 loss

Mean square error (MSE) is the most commonly used regression loss function. MSE is the sum of squared distances between our target variable and predicted values.

Mean absolute error, L1 loss

Mean absolute error (MAE) is another loss function used for regression models. MAE is the sum of absolute differences between our target and predicted variables. It measures the average magnitude of errors in a set of predictions, without considering their directions.

Huber loss, smooth mean absolute error

Huber loss is less sensitive to outliers in data than the squared error loss. It's also differentiable at 0. It's basically absolute error, which becomes quadratic when error is small. How small that error has to be to make it quadratic depends on a hyperparameter, δ (delta), which can be tuned. Huber loss approaches MAE when $\delta \sim 0$ and MSE when $\delta \sim \infty$ (large numbers).

When to use which?

One big problem with using MAE for training of neural nets is its constantly large gradient, which can lead to missing minima at the end of training using gradient descent. For MSE, gradient decreases as the loss gets close to its minima, making it more precise.

Huber loss can be really helpful in such cases, as it curves around the minima which decreases the gradient. And it's more robust to outliers than MSE. Therefore, it combines good properties from both MSE and MAE. However, the problem with Huber loss is that we might need to train hyperparameter delta, which is an iterative process.

More reading and full solutions here.

What is the 0–1 loss function? Why can't the 0–1 loss function or classification error be used as a loss function for optimizing a deep neural network?

From Encyclopedia of Machine Learning

Zero-one loss is a common loss function used with classification learning. It assigns 0 to loss for a correct classification and 1 for an incorrect classification.

The reason it isn't a good fit as a loss function for optimization has to do with convexity. It is non-convex and also non-differentiable at 0. Therefore, even if you derive a derivative to make the function differentiable, the function remains non-convex and difficult to optimize. Convex functions are a better choice, such as the hinge loss in conjunction with the SVM model.

More reading from Quora here.

Monte Carlo Methods

What are Monte Carlo algorithms?

Solution from Towards Data Science

Springboard

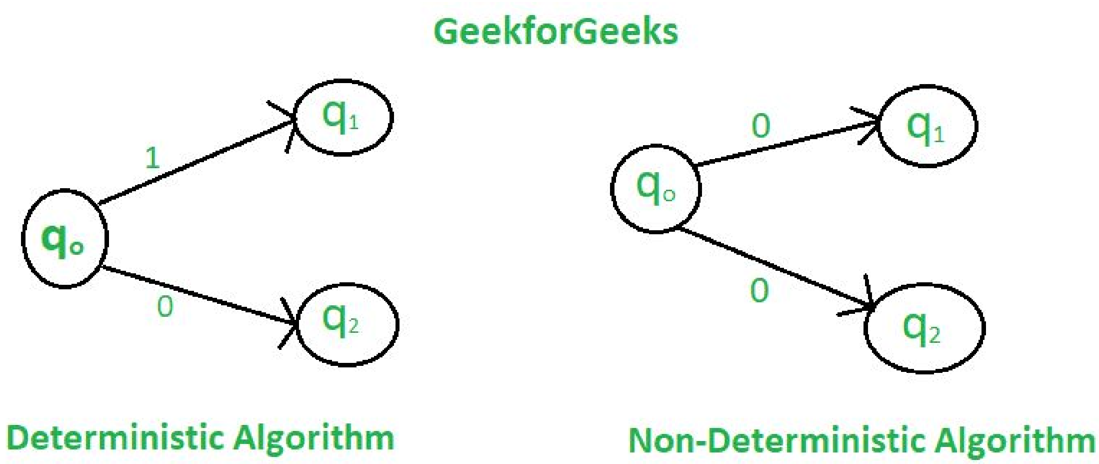
Monte Carlo (MC) methods are a subset of computational algorithms that use the process of repeated random sampling to make numerical estimations of unknown parameters. The method finds all possible outcomes of your decisions and assesses the impact of risk.

More on the Monte Carlo simulation [here](); how to estimate Pi using the Monte Carlo method [here]().

What are deterministic algorithms?

Solution from [GeekforGeeks]()

In deterministic algorithms, for a given particular input, the computer will always produce the same output going through the same states. Deterministic algorithms are the most common type of algorithm, and most practical, because they can run on computers efficiently. But in the case of non-deterministic algorithms, for the same input, the compiler may produce different outputs in different runs. In fact, non-deterministic algorithms can't solve the problem in polynomial time and can't determine what the next step is.



Representation

What is representation learning? Why is it useful?

From [Quora](), citing Yoshua Bengio's [original paper]()

Representation learning is learning representations of input data typically by transforming it, making it easier to perform a task like classification or prediction. There are various ways of learning different representations. For instance:

- in the case of probabilistic models, the goal is to learn a representation that captures the probability distribution of the underlying explanatory features for the observed input. Such a learned representation can then be used for prediction.
- in deep learning, the representations are formed by composition of multiple non-linear transformations of the input data with the goal of yielding abstract and useful representations for tasks like classification, prediction, etc.

It is useful because models are dependent on the presentations that they learn to output. In deep learning, it is useful because representation allows a complex block to transform an input into a rich representation, which only requires a simple layer to do task specific preparation (in using BERT for NLP tasks, for example). The model outputs representations of input that can be used for a number of NLP tasks. Further, understanding how different representations are good for specific tasks can help practitioners better know various deep learning model architectures.

What trade-offs does representation learning have to consider?

Solution from [Deep Learning Book]

Most representation learning problems face a trade-off between preserving as much information about the input as possible and attaining nice properties (such as independence).

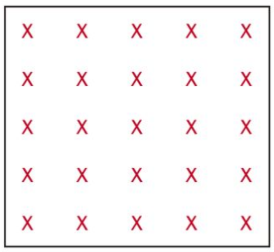
Dimensionality Reduction

Describe the curse of dimensionality with examples. Why do we need dimensionality reduction techniques?

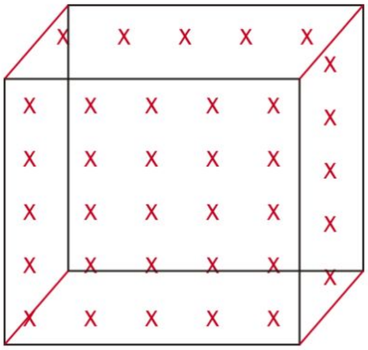
From [Towards Data Science]

As the dimensionality of the features space increases, the number of configurations can grow exponentially, and thus the number of configurations covered by an observation decreases. Basically, the more data there is, the more processing power we need, and we also need to have more training data in order to have a meaningful model.

A one-dimensional features space with five data points



A two-dimensional features space with 25 data points



A three-dimensional features space with 125 data points

So you can see that if there's a way to reduce dimensionality, you can economize. The technique of dimensionality reduction can help to compress data without losing too much signal.

More on dimensionality reduction [here](#).

Interest and Understanding of ML

What are the last machine learning papers you read?

More reading: [What are some of the best research papers/books for machine learning?](#)

Keeping up with the latest scientific literature on machine learning is a must if you want to demonstrate interest in a machine learning position. This overview of [deep learning in Nature] by the scions of deep learning themselves (from Hinton to Bengio to LeCun) is a good example of the kind of paper you might want to cite.

Do you have research experience in machine learning?

Related to the last point, most organizations hiring for machine learning positions will look for your formal experience in the field. Research papers, co-authored or supervised by leaders in the field, can make the difference between you being hired and not. Make sure you have a summary of your research experience and papers ready—and an explanation for your background and lack of formal research experience if you don't.

What are your favorite use cases of machine learning models?

More reading: What are the typical use cases for different machine learning algorithms? (Quora)

The Quora thread above contains some examples, such as decision trees that categorize people into different tiers of intelligence based on IQ scores. Make sure that you have a few examples in mind and describe what resonated with you. It's important that you demonstrate an interest in how machine learning is implemented.

How do you think Google is training data for self-driving cars?

More reading: Waymo Tech

Machine learning interview questions like this one really test your knowledge of different machine learning methods, and your inventiveness if you don't know the answer. Google is using recaptcha to source labeled data on storefronts and traffic signs. They are also building on training data collected by Sebastian Thrun at Google X, some of which was obtained by his grad students driving buggies on desert dunes!

How would you simulate the approach AlphaGo took to beat Lee Sidol at Go?

More reading: Mastering the game of Go with deep neural networks and tree search (Nature)

AlphaGo beating Lee Sidol, the best human player at Go, in a best-of-five series was a seminal event in the history of machine learning and deep learning. The paper above describes how this was accomplished with "Monte-Carlo tree search with deep neural networks that have been trained by supervised learning, from human expert games, and by reinforcement learning from games of self-play."

Case Studies/Scenarios

These are more specific scenarios that a company could give you that apply directly to their business.

What do you think of our current data process?

More reading: The Data Science Process Email Course (Springboard)

This kind of question requires you to listen carefully and impart feedback in a manner that is constructive and insightful. Your interviewer is trying to gauge if you'd be a valuable member of their team and whether you grasp the nuances of why certain things are set the way they are in the company's data process based on company- or industry-specific conditions. They're trying to see if you can be an intellectual peer. Act accordingly.

How can we use your machine learning skills to generate revenue?

More reading: Startup Metrics for Startups (500 Startups)

This is a tricky question. The ideal answer would demonstrate knowledge of what drives the business and how your skills could relate. For example, if you were interviewing for music-streaming company Spotify, you could remark that your skills at developing a better recommendation model would increase user retention, which would then increase revenue in the long run.

The startup metrics linked above will help you understand exactly what performance indicators are important for startups and tech companies as they think about revenue and growth.

Product Management Skills

Here are key questions and prompts from a great prep document from Rafi Lurie. I haven't gone into specific solutions here as they are much more open-ended, but check the guide for additional breakdowns on how to think about these problems.

Product design: Design a washing machine for blind people.

- As a next step, design a laundromat full of those washing machines and describe the experience from the moment you walk in until the moment you leave.

Product vision: What product do you feel has a lot of potential but hasn't achieved it yet?

- Why?

- What would you build to help this product become successful?

Conceptualizing, designing, and building a new feature is only half the battle. How do you launch a new feature?

- How does it get incorporated into the existing product?
- Does that stay stable over time or does this feature change throughout a user's lifetime?
- How would you spread awareness about your product?

Related: a comprehensive outline of a product manager interview process here; a good breakdown of how you should spend your preparation time.

In Their Own Words: What Hiring Managers Are Looking For

We interviewed hiring managers from startups, software companies, and big corporations in order to give you a broad understanding of the different types of interviews.

Further, these managers each have different roles and hire for different types of candidates. They should give you a decent sense of what each of them are looking for and what you can do to prepare and stand out.

Susie Pan - Royal Bank of Canada, Product Lead

What do you look for?

One of the first things we do is look at what a candidate has done on GitHub. We want to see how active they are as a contributor and what personal projects they have worked on and/or open-source projects that they have contributed to.

For junior people, we would also look at how they performed in university projects, hackathons, and research. But for senior people, it's far more about work experience. We would want to see that they have done work with real data, not just standard pre-processed data sets, and what they've done with it.

What's an outline of your interview process?

For machine learning engineers, it's:

- Resume screen
- Phone screen
- Technical interview (mix of take-home challenge and on-site coding)
- Deep, technical on-site interview (with our engineering team) with a product on-site interview (with our product team)

What's the best advice that you can give job seekers?

My best advice is to actively work on projects and have them available on your GitHub profile.

Springboard

Also, work on real problems with real data because in the real world, most problems don't have perfect data. The majority of your time is going to be spent cleaning up data.

You need to be able to show that you can go through the entire workflow from data engineering to modeling to productionizing your code.

What are the key questions that you ask? Anything particularly unique?

For technical questions, we would ask questions like:

- When have you worked with a data set and what did you do with it?
- We will ask, in different scenarios, why do you select one model versus another one? What are the pros and cons?
- Can you explain how you got your results? How do you go through model validation?
- What is the implication of what you found?

For non-technical questions, we want to see your thought process:

- How do you deal with ambiguity?
- How do you figure out what to work on, not just how to work on it?
- Do you understand the business implications of what you are working toward and its impact?

What do you test for?

Ultimately, we are testing for machine learning literacy, that you have the ability to use your knowledge in practical situations to see the whole process through, and that you have a genuine outside-of-work interest in ML and AI.

Integrate.ai - Rachel Jacobson, VP of People; Brennan Biddle, AI Recruiter

What do you look for?

At Integrate.ai, we are incredibly values oriented. We look for people who share our DNA and people who are truly interested and excited to work with us. It's also important for us to know that a candidate is comfortable with the ambiguity and fast pace of a startup. Our key values are: love people, build trust, focus on impact, take action, and be present.

We aren't looking for a cookie-cutter profile. Hiring for these roles used to mean "find someone in San Francisco that has a Ph.D." But it's a bit different now. These days, everyone and their neighbor calls themselves a machine learning scientist or a data scientist. So we have to go deeper. We look for diversity across a number of different angles. Yes, our researchers still tend to have Ph.D.s, but they don't have to be in data or stats: one of our engineers has a Ph.D. in astrophysics and did research on blackholes.

We are also open to hiring from around the world and bringing people to Canada.

Another note is that we don't typically employ consultants or contractors. For us, it's really important to build an employee base and culture.

What's an outline of your interview process?

Step 1: 30-minute phone call with a machine learning recruiter

- Discuss projects you have worked on and the impact you have had on those projects.
- What did you actually do to make it happen?

- What will it take for you to be happy working here?

Step 2: 45-minute technical phone interview with a member of the machine learning team

- Shared screen interview.
- Writing code live with one of the team members (typically in Python).

Step 3: 3-4 hour in-person interview

- Behavior and values
    - This could even be with an entry-level employee, as long as they are well versed in the values of the company.
- Technical - three 45-minute technical interviews, often with three different people
    - Each will focus on something different. Each interviewer will know something specific about the candidate.
    - We'll go through systems integrations, data structures, and algorithms (currently; we change these constantly).
    - Candidates don't need to hit a home run in all three, but they do have to do fairly well in all three (i.e., As in two with a C in the other).

Note that:

- We don't allow the skills and values interviewers to interact.
- Sometimes this allows us to get to candidates that we wouldn't otherwise get to.
- The hiring panel will meet either at the end of that day or early the next day.

- Even though it's an exhaustive experience, we're able to make an offer within 24-48 hours.
- Shining differences for us:
    a. We don't hire machine learning scientists; we hire humans who do machine learning.
    b. There is a large amount of humanity in what we do.
    c. We want to get to know you as an actual person.
    d. We want Integrate to represent not just your best work years but the best years of your life.

What's the best advice you can give?

The best advice we can give is to be vulnerable and show your humanity in our interview process. We care about that just as much as your machine learning capabilities.

What do you test for?

Half of our interview process is focused on behavioral competencies and values. The other half is typically whiteboarding solutions to ambiguous problems, often using Python. We test for coding abilities, typically in Python. We also really care about software skills in data and being able to make sense out of and communicate insights about ambiguous data.

Geetu Ambwani - Data Science Lead, Flat Iron Health

What do you look for?

We are a health technology company focused on oncology. We use data to advance cancer research. We have a product-focused data science team, so we are hiring analysts, data scientists, and software engineers. They are focused on data-driven product discovery and making data-driven decisions.

What's an outline of your interview process?

1. The first step is usually a cold analytical exercise, where we give you a data set and a few questions for you to answer.

2. Then we have a phone/Zoom screen, testing for analytical skills, as well as to get to know you and the work that you have done. We'll ask you about one or two projects that you have done in your portfolio. And then the majority of the interview is really focused on an open-ended exercise, which is trying to get at your data modeling and data analytical skills.

Besides your technical skills, we also index on your ability to take something that is ambiguously framed and then to dive into a business or product context. We are interested to understand how much information you can give when thinking about the user.

3. Once you've finished the screen, you come on-site, usually speaking to four or five different groups.

One is focussed on your product thinking skill set. We would present 1-2 interesting open-ended product ideas that we would ask you to talk through and discuss how you would prioritize different decisions and trade-offs.

Next up is a coding interview, which is about whiteboarding a coding problem.

The third is a cross-functional collaboration exercise. Because of the nature of our work, we are in very cross-functional teams, everyone from medical professionals, oncologists, and data analysts on same team, side by side, as well as folks typical to a tech company, like a product manager. Everyone has a different set of skills and it's important for us to see how you could work in and collaborate across different functions. In this type of interview, it typically focuses on behavioral questions to see how you can talk across your space to someone from a different background.

The fourth exercise is something that is a bit experimental. Depending on your skill set or your path, we either give you a data science or a data modeling interview. Modeling is focused on—surprise—data modeling, as well as SQL and analytical skills, and how good you are at it. Data science is more statistical machine learning methodology.

What's the best advice that you can give?

Get your basics right. A lot of people are so focused on the buzzwords in machine learning and deep learning. Depending on where you're going, at least with us, we don't build large-scale systems or focus on the problems that Facebook or Google does. But for us, it's about understanding and framing an open-ended business problem into an analyst's domain. Take a business problem and understand how to use data to answer the question.

Then, being able to have machine learning fundamentals, like regularization.

What are the key questions that you ask? Anything particularly unique?

We care about if you are able to turn questions back to the customer and keep the end user in mind. We aren't obsessed with technical nuances. What does an MVP look like? What would it look like if you double the resources? How do you measure your success? And do you take feedback well.

What do you test for?

We care a lot about cross-functional people who also understand product. We also have a "reverse interview" process where candidates are allowed to schedule meetings with different members and functions of our team so that they can really know what's going on. It's important for us.

In Their Own Words: How Successful Candidates Made It

Patrick Lung - Product Manager, Microsoft

What surprised you and what did you find difficult?

I wasn't particularly surprised by any of the questions in the interview. I found during the interview process that it was very behavioral and a lot of it was about my own experiences and perspective; for example: "What do you think about machine learning and product development?"

To be clear, they also did ask a solid number of product questions. But even though I was interviewing for a machine learning role, there weren't a lot of questions that were specific to machine learning. Much of the interview was framed with "Tell me about a time that you..." and you would adjust your answer accordingly.

For example: "Tell me about a time you disagreed with an engineer." In my case, an example of a time that I disagreed with an engineer is when a machine learning algorithm we were working on together provided bad suggestions. Initially, we overreacted and realized that we did not create it in the way it's typically done. So we backtracked, made amends, and had an improved working relationship going forward.

What advice would you give to ace the interview?

Recognize if an application/problem is a good ml candidate for interview questions

For the interview it's important to recognize what applications and problems are good candidates for using machine learning, and which are not. There are many scenarios where you don't need to use machine learning. Further, there are times where employing machine learning could actually be counterproductive to the problem that you're trying to solve.

The essential question to ask yourself when you encounter an interview problem is "could I meet 80% of my needs with a rules-based model?" There are three key benefits to using rules-based models:

1) Rules-based models are cheaper. They take less time to build.

2) Rules-based models are clearer to troubleshoot, because you code and create the logic.

3) Some user behaviour is better off being deterministic. For example, user does X, so Y happens. It's more predictable.

You are soon changing jobs to work at Kite. Why did you decide to work there?

I wanted to work somewhere where machine learning is a principle aspect of the problem. Not powering a side feature. This company depends on machine learning for their whole product.

The second reason is that the team is very strong at machine learning. You want to learn the best practices and learn from the best.

Springboard

Srdjan Santic - Principal Data Scientist, Logikka

What do you do?

Right now I run Logikka, my own AI and ML consulting company. But my most recent job was at a startup that focuses on document classification and detecting sensitive information in documents. Much of it is about finding out the historical lineage of documents so that you can trace them back.

What advice would you give to ace the interview?

My advice is to be well prepared for the coding part of your interview. Your methodology matters significantly and most of the point of the interview is to show your work.

Of course, you also need to be up on your statistics and machine learning skills. And more than knowing simply the techniques and solutions, you need to know these on an intuitive level.

Knowing the math is obviously helpful as well.

Be aware that you might be interviewed by business stakeholders, not just technical ones. But regardless of who is interviewing you, it is important to remember that you are there to solve a business problem.

Also remember to be polite and kind to everyone, from the receptionist to the last person you talk to.

How did your interview process go?

- Phone screen: this was for company fit and asking about my recent experience.
- Then we had a take home challenge, which was a data set.
- Technical interview with a senior data scientist.
    - This is when we went through machine learning questions. It was where the tough questions became clearer.
    - The important part was relating my answer to my previous experience.

My general thoughts on the interview process are that:

1. In-person is a lot more personal. It's easier for you to gauge what the other person is like. And ultimately, despite the technical questions, they want to get to know you as a person. In my opinion, you can evaluate the company better in person. But if you are remote: make sure to ask about the projects that the company has recently done.
2. Ask questions like these:
    a. What would I work on if I were to come aboard?
    b. How are ML projects managed? Are they planned out?
    c. Are there agile practices in place?
    d. What hours do people typically work?
    e. How does the team keep up with current developments in the field? For example, some companies I have worked at have had a reading circle every Friday.
    f. When working on a project as a data scientist, whom would I report to?

Why did you choose your current job?

I wanted to move! The company was based in Ireland and I wanted to be there. Also, the company was a startup and I really wanted to work in a small startup, building something from scratch. Further, the team looked really great, and I had a chance to talk to everyone during a session where we all got to meet each other.

The technical interview was very challenging, but the interviewer was very pleasant, even in situations when we disagreed on something. So I knew that I would be learning from someone who was incredibly knowledgeable.

Val Andrei Fajardo - Director of Machine Learning Science, Integrate.ai

What surprised you and what did you find difficult?

Sometimes the job description may not exactly match what you end up interviewing for. I had come in for a more generalist position, but at one point there was an NLP-specific problem on the whiteboard. If I had known there had been an NLP problem for this role, I would have prepared for that. That can happen with a lot of companies and their job postings.

Something that surprised me which was great was that there were a lot of values pieces at Integrate. Not a lot of companies do that. To make sure there was not a conflict between my values and the company was really important.

What advice would you give to ace the interview?

There is a lot of white boarding. So prepare for lots of open-ended questions on building solutions to a complicated problem. You have a chance to use the mathematics and foundation that you have to build a solution. It may not have even been the same one that they interviewed. But it will still achieve the same outcome.

If you can somehow get more specific prep from the company, without getting the actual questions, that can help you prepare.

In my case, if I were to do it again, I would use the script that, "Hey, I'm typically a generalist so it's very helpful for me to know what specific problems and ML areas I can focus on for the interview." I would also ask about the kinds of numbers and problems they are looking at as a business. Getting some insight there will help you understand how to best prepare.

One of the pitfalls in an interview like this is looking too narrowly at something—missing out on the big picture, the main values, the main elements that person is looking for. In my case, I was coming from more of a theoretical background. I faced a bias going into machine learning, people thinking that I was an academic, not as much of a coder. That's what you might face if you have a strong theoretical or stats background.

Also, have evidence that you know how to code, taking a mathematical problem and turning that into production-ready code. The more examples you have of products and projects that you've worked on, the better chance you will have.

The best advice I have for you on the behavioral side is only be yourself. Because if you have multiple behavioral interviews, you can't fake it multiple times. Stay humble and demonstrate your curiosity. Be vulnerable and authentic—show your humanity.

Why did you choose your current job?

I chose this job because of the interview process. The people I got to meet during it were amazing. They made be feel like a human throughout.

I also wanted to learn how machine learning could be used across a wide array of businesses. At Integrate, we work across a wide variety of industries. So… what better place to do that?

Jasmine Kyung - Senior Operations Engineer, Raytheon

What surprised you and what did you find difficult about the interview?

What I expected was a project that only focused on machine learning algorithms, but the project instead detailed even very basic code and didn't care too much about the algorithm in itself. The pre-analysis and my approach to solving the data science issue in itself was what they cared about. They weren't expecting something super complex or with 100% accuracy. It caught me off guard.

Something else that surprised me throughout the interview was that they were more impressed by the way I ran a pre-analysis after cleaning the data. They were more impressed by that rather than just applying the machine learning algorithm. They cared whether I could actually find the right inputs beforehand.

What advice would you give to ace the interview?

Get information and derive insights before jumping into the algorithm. After data cleaning, you can play with the data to see what insights you can get (e.g., a correlation chart). It's a simple analysis that you can do without any kind of algorithm. You can also share that with customers to get them excited fast.

Go in confidently. Even if you're new to data science or machine learning, your talent will be in high demand. Know that you are valuable. It's also not just, "Do you know tons about algorithms?" They're looking more at your train of thought and how you approach certain questions.

Why did you choose your current job?

It seemed really exciting! I was going to be the first data scientist, the one who brings data-driven solutions to the company. It was a very nimble and flexible organization with many opportunities to learn.

Also, the job seemed like a good one. It would get me ahead in the field.

Takeaways

The key takeaways for a machine learning interview are:

1. Don't think questions about basic material won't be covered. Read up on technical fundamentals before you go through the interview process.

2. Be prepared to do well in the behavioral interview. Companies care about your communication skills and your ability to get along with future coworkers as much as they care about your coding skills and ML knowledge.

3. Have stories ready to share. Have a portfolio, especially on GitHub. Be prepared to storytell about who you are and why your passions and skills are

uniquely valuable for the company at hand. Having relevant projects and being very clear about what you contributed to those works will mark you as a candidate worthy of passing to the next round.

4. Be patient. An interview process can take a long time. You'll want to be prepared to wait.

Now that we've gone through the actual machine learning interview process, let's look at what happens after you've finished interviewing.

7 Things to Do After the Interview

After you've finished your machine learning interview, you might think your work is finished. That's not necessarily the case.

Here is a list of things you can do after the interview to ensure, as best as possible, that you maximize your chances of making the best impression on your potential employers.

1. Send a (good) follow-up thank you note

How you follow up on an interview can make the difference between internal advocates fighting to get you in and apathy. It is now customary to send a thank you note. With each office worker receiving an average of 121 emails a day, however, you won't want to just stick with a boilerplate "thank you for the opportunity" email. Make sure you're remembered. A nice email is the bare minimum. Candidates who take the extra step of sending handwritten notes or a list of thoughts after the interview will stand out from the rest of the 120 emails.

2. Share thoughts on something brought up during the interview

One easy way to differentiate yourself is to go beyond simply saying thanks. Remember what happened in the interview and make a conscious effort to tease out some of the pain points the employer is trying to solve. If sample problems within the interview are oriented toward a technical direction, or a question suggests a disconnect between different teams, you'll want to make a note of it and send in-depth thoughts.

After all, an interview isn't just a test;it's a discussion. If you listen carefully to the questions presented and ask the right questions yourself, you will know exactly what problems the company is facing. Why not send thoughts on the solutions you'd pursue?

3. Send relevant work/homework to the employer

It can be difficult envisioning how your skills could apply to the office, especially for somebody who has just met you. The sharpest hiring organizations will often give you a sample problem to solve that is sourced from real issues they are facing. This gives you the chance to demonstrate how your efforts could impact the business in a positive manner. Organizations that don't do that will hesitate to hire the right candidate because they haven't sufficiently demonstrated how they'd drive impact for the company in question. However, you can be proactive and use what you learned in the interview to follow up. You don't have to stop at sending them thoughts that show you listened carefully;you can give them actual, tangible solutions.

The author of this post on Forbes was told that she didn't have enough of a portfolio to get a job as a freelance copywriter. Having listened carefully throughout the interview, the candidate knew that a major project (the redesign of a website) was just over the horizon. Instead of accepting defeat, she sent 10 proposed headlines for the website banner, free of charge. This burst of initiative got her a job doing the rest of the writing for the website.

You need to have a portfolio that shows the impact you can make, but sometimes that isn't enough. If you're astute and you ask the right questions, you can find a business problem or ML opportunity for the company. There's always something—that's why they're hiring in the first place! There's a project out there that everybody would love to see done or a thorny problem that no one can figure out. Send them a plan for what you'd do or play with some of the data they've divulged, and then give some solid insights into how you work. Initiative will go a long way to getting you an offer.

4. Keep in touch, the right way

One of the most awkward parts of the post-interview process is waiting for a response. You don't want to come off as desperate by following up too many times, but companies take their time if you don't engage with them proactively. It is possible to affect the post interview decision from outside of the company, but you should keep in mind the appropriate channel to reach somebody. Make sure to ask before the interview ends how best to reach your interviewer. Everybody has a preferred mode of communication;if they specify short emails or a call, follow that rule and dispel some of the post-interview awkwardness.

As a rule of thumb, don't check in more than once every week, or better yet, in 10 days. Sometimes companies just take their time.

5. Leverage connections

Ideally, you'll have come in with strong references both from external and internal sources. If you had been building your network and providing value to them, you should have strong advocates who can support your candidacy. Check in with people who referred you internally every once in awhile, and if needed, get them to mention how excited you would be to work at the company and how lucky the company would be to hire you. Hiring is often network-driven, and the best signal that you can send to a potential employer is a vibrant network of people who are willing to go to bat for you.

6. Accept rejection with professionalism

Look, there's a good chance you will get rejected. Sometimes you’re just not right for the role, or they might have found somebody who is a slightly better fit. It’s important at this point to maintain your composure, thank the employer for their time, and move on.

People in the industry talk amongst themselves, and being unprofessional at this point will only be bad karma and might get you ignored at other companies. Being professional ensures the health of your network. More importantly, a no isn’t always a no. Sometimes companies really do keep your profile on file and will reach out in the future.

Perhaps Winston Churchill put it best when he said, “Success is the ability to go from one failure to another with no loss of enthusiasm.” J.K. Rowling has shared her rejection letters from publishers. Brian Chesky, the founder of Airbnb, published seven rejection letters from potential investors. In order to achieve greatness, you will have to endure rejection. Everybody successful already has.

7. Keep up hope

The interview process can be filled with great anxiety. Your future can be mapped out by deciding what company you work for. An interview can mean the beginning of a career change. It can mean moving cities. It is a period in our lives where other people have disproportionate control over our destinies. Nevertheless, as seen in the previous steps, you control a lot more than you think. It's important to keep your head up and do what you can.

The most important thing you can do during the interview process is keep up hope. Interviews are lengthy. Companies take time to get back to you. There are many internal checks and processes before a candidate is accepted. You may go through multiple rounds of interviews with the same company and not seem any closer to a final offer. You have to set expectations. You should never be disheartened during your journey.

How to Handle Offers

Your goal is to get as many offers as possible that you can evaluate and potentially negotiate. While the process itself is difficult, and may take longer than you expect, once you start getting offers, you'll have earned them. We can't emphasize enough how important it is to manage your expectations and keep your hopes up.

For many candidates in the AI and machine learning space, it can take months to even half a year to find the right role, particularly if they are coming from academia. Make sure you weigh what is presented to you and choose the future you deserve once you've put in all the hard work earning it.

What to Assess

If you complete the interview process successfully, you may have multiple offers. Congratulations! Accepting an offer is a commitment of a significant amount of your time to the company in question. Always keep that in mind.

Springboard

There are several factors you can use to ascertain whether or not an offer is the right one for you.

Company Culture

This might be one of the most important factors in determining when an offer is one you should accept. Make sure you ask about the company culture. Look for signs that the company employs individuals who genuinely enjoy spending time with one another; run away from generic descriptors or companies that struggle to define their culture or even wave the question away.

Great companies invest tons of time and effort into making sure they hire awesome people who love what they do. That'll come through in your questioning. You should also check external and objective sources, such as company reviews on Glassdoor. Approach current employees as well as former ones (whom you can find on LinkedIn) to get their side of the story. You'll often find candid tales that can give you a good preview of what working at your new job would be like.

Team

Company culture is an extension of the team that inhabits it, but you should be excited about coming to the office every day and working with everybody else. Make sure that you're working with a team that you can learn from. You are the average of the five people you spend the most time with—and you're going to be spending a lot of time with your colleagues.

Location

Make sure you're comfortable with the company's location, especially if you're moving a significant distance to take the role. Moving is a difficult process, so it’s important that you feel at ease with where you live. Factors like the weather and the transit system matter to a certain degree, especially if you’re planning to live under those conditions for several years.

Negotiating Your Salary

An astonishing 61% of people didn't negotiate their salary in their last job offer, despite the fact that those who do typically see their salary raised by 13.3%, according to CNBC and Glassdoor.

When you first get your offer, you’re at a unique leverage point that you might not see again for several years. This is the time to test what you’re worth. Reach out with a counteroffer—a company won’t fire you or cancel a contract offer because you were asserting your worth. Initial offers are sent with a buffer for slight negotiation. Take advantage.

During a salary negotiation:

1. Come with a well-researched number representing what you think you’re worth. Look to industry averages (listed below) and get a sense from people working in the field what you should expect. Never come into a negotiation without knowing what you want out of it.
2. Stay positive and don’t push too hard for what you think you “deserve.” Instead, use this as a positive experience to assert your worth and the value you can create.
3. Know what your minimum is and ask for more. Negotiate a little bit higher than the amount you think you’ll actually get. Anybody experienced at negotiation will come back to you with a counter, and you’d best be prepared for it.
4. Most importantly, don’t fear rejection! So long as you keep the process moving forward civilly and professionally, a company will appreciate

you being frank and positive at what is often the most difficult part of the recruitment process for them. Before you accept the offer, make sure you know how committed you are to the company, team, and money.

5. Use your other offers as leverage. It's always best to go into a salary negotiation with at least one other competitive offer. Even if you know the job you want, it's important to be able to negotiate to get what you want in terms of compensation.

Industry Benchmarks for Salaries

Negotiation is always easier if you have some information about average salaries to ground you. The more you know, the stronger you'll be at the negotiation table.

Salary data often changes, but here are some facts and figures that can start your research.

Indeed.com cites an average salary of $141,000 for machine learning engineers, an average salary of $120,000 for data scientists, and approximately $140,000 for product managers at the biggest tech companies. This varies from region to region, with the highest salaries tending to cluster in the tech-heavy Bay Area. California has the highest range and median of all regions when it comes to data science, according to O'Reilly Media. Globally, the United States has the highest median and range of data science salaries, while the United Kingdom, New Zealand, Australia, and Canada aren't too far behind. Asia and Africa tend to have the lowest medians.

The highest-paying industries are technology and social networking companies, while the lowest-paying ones tend to be education and nonprofit sectors. Salary also varies based on skills and tools used. O'Reilly has a definitive survey of hundreds of respondents in the industry. An open study, the results indicate a variety of factors that lead to different average salaries, including location, industry, and job title.

Check out Glassdoor for more valuable salary data. And this is a good resource for broader data on major tech companies.

What to Do Once You Accept

If you've accepted an offer, congratulations! Take a minute and exhale! You've accomplished the goal of this long process and broken into the job you've sought, a job that promises excellent compensation and the ability to drive significant social impact.

Be aware that good companies will work to make you as comfortable as possible. You should reach out to future teammates and figure out what they do and how you can help them solve their business problems. Take the time to socialize and meet as many people as you can.

More importantly, if you have time between when you accepted the offer and when you start, relax and enjoy! Make sure you catch up with as many people as you can in your life, take the chance to rest, and be completely refreshed for your first day at your new company.

Conclusion

The AI and machine learning interview process is one of the hardest recruitment processes to crack, and it's one of the most competitive. Your fellow interviewees will likely be experienced engineers, Ph.D.s, researchers, or product managers. And some of them will have extensive experience in the field.

While the space is attracting many talented people, remember that it has a slew of different related industries, teams, and roles. If you think outside of the box and apply a few battle-tested tactics, you'll be able to get an interview and take it all the way to an offer you love.

Split the process into its composite steps, and remember what it takes to succeed. Don't hunt for jobs like everybody else by limiting your search to the standard job boards and sending out typical cover letters. Reach out to people within organizations you admire for informational interviews. Do something different from the hundreds of other candidates. Stand out as a great technical thinker and, above all else, a proficient communicator (not to mention… a great person).

Use this and other guides to thoroughly understand the technical and nontechnical parts of the interview. Once you've mastered the thinking behind the questions and what hiring managers are looking for, you'll have a good sense of how to excel throughout the process.

Final Checklist

Here's a final cheat sheet for you as you prepare for your interview. Remember the following:

1) Understand the roles that your skills fit
2) Map out the industries and types of companies you want to work for
3) Prepare your LinkedIn, CV, and email templates
4) Research each company and role you want to aim for thoroughly
5) Reach out proactively to individuals for informational interviews
6) Build strong networks and referrals
7) Tackle the interview

8) Follow up respectfully
9) Negotiate and accept

Special Thanks

Thanks to the Springboard team and authors of "The Ultimate Guide to Data Science Interviews," which provided great structure and content for this guide:

T.J. DeGroat

Roger Huang

Sri Kanajan

Special thanks to two of my best friends and machine learning geniuses for reading this and providing edits, suggestions, and laughs:

Patrick Lung

Logan Graham

About the Author

Jaxson Khan is CEO at Khan & Associates, a global advisory firm that helps innovative companies and organizations with strategy, communications, and growth. He previously served as head of marketing at Nudge.ai, the Canadian AI Company of the Year in 2018. Jaxson is a host of the Ask AI podcast, a mentor with Techstars, an instructor at Product Faculty, an advisor to Century Initiative, and a member of the World Economic Forum