BEGINNER TO (EMPLOYED) DATA SCIENTIST

COMPLETE ROADMAP

SKILLS, PROJECTS, RESUMES, INTERVIEWS, & MORE

GRATITUDE DRIVEN

So, you want to become a Data Scientist. Maybe you're pivoting from a different field, or maybe you've just graduated and you're looking at this mountain of skills to learn and conflicting information about how to prioritize your time, and you're wondering, "Where do I even begin?"

Don't worry. In this roadmap, I'm going to walk you through the exact steps to go from zero experience to landing your first role as a Data Scientist in just one year – without a bootcamp or advanced degree.

You'll learn what skills to prioritize (and in what order), how to build real-world experience before you even have a job, how to make your resume and LinkedIn stand out, and how to excel in tricky technical interviews. We'll even talk about what life is like in your first role so you know what to expect and how to get your career started on the right foot.

Quick intro: I’m Marina. I work as an Applied Scientist (a hybrid between a Data Scientist and a Machine Learning Engineer) at Twitch (i.e. Amazon). I also mentor at an machine learning bootcamp and coach people one-on-one on how to break into the field.

I transitioned into data science from a non-technical background, so I know firsthand how challenging—and rewarding—this path can be. As a little motivation, **I earn five times what I did before transitioning into the field**, while at the same time having better work-life balance and a more engaging day-to-day now than I did back then.

With consistency and a solid plan, you can do it, too!

If you feel like you need more support, you can book a 1:1 call with me on topmate.io. And don't forget to check out my YouTube channel for more Data Science education and motivation.

Also, no pressure at all, but if you find this guide helpful and feel so inclined, you can click here to buy me a coffee.

Should You Still Become a Data Scientist?

Right away, let's address a question you might be worried about: Is it even worth it to become a Data Scientist now? With all the layoffs in tech and the increasing presence of AI, it's natural to question whether data science is still a smart career choice.

Here's why I believe that it is: First, while tech layoffs have absolutely been real, they don't tell the whole story. Even with recent cutbacks, **data science remains one of the most in-demand fields**. According to the U.S. Bureau of Labor Statistics, employment of Data Scientists is projected to grow 36% from 2023 to 2033, which is much faster than most other professions.

And what about AI? It's true that AI is reshaping industries, but it's not eliminating Data Scientist roles – at least not yet. More importantly, **the core skills you'll develop**—like programming, statistics, and critical thinking—**are valuable across many industries**, even if you do need to pivot in the future.

So, while it might be harder to break into data science now compared to a few years ago, the rewards are absolutely still there for those who are persistent and focused. If you're willing to put in the effort, this path can lead to a well-paying, intellectually rewarding career with plenty of opportunities for growth.

In This Roadmap

Timeline

You've probably seen videos or courses promising you can become a Data Scientist in just a few weeks or a few months.

While that *might* be realistic for someone with a super strong technical background and *a lot* of free time, for most beginners, it's not going to be feasible.

Data science isn't easy. It's not a job where you just learn a few simple things and then coast. Building the necessary skills takes time, but it's worth the investment.

This process is definitely a marathon, not a sprint.

That being said, if you can dedicate about 20 hours a week to learning, this 12-month timeline will guide you step-by-step from beginner to that first job.

So if you're starting in January, this timeline fits perfectly. If not, don't worry, just use it as a flexible template. Let's break it down.

January to June (Weeks 1-26)

Building the Basics

The first six months are all about establishing a strong foundation. During this phase, you'll learn the essential skills every Data Scientist needs, while also starting to build your network.

Step 1: Join the Data Science Community

First things first, we're going to start meeting people. Find local meetups, Discord servers, or LinkedIn communities where Data Scientists connect.

Building relationships early will help you down the road, whether it's finding mentors, collaborators, or even job opportunities. Plus, you'll start learning the language of the field and getting familiar with the kinds of things people are working on, which will be helpful when you're coming up with projects and preparing for interviews.

Step 2: Start Learning

This is your "reading and practicing" phase. We're going to start with beginner-friendly courses, books, and YouTube tutorials to dive into the fundamentals. Throughout this phase, we'll work on follow-along projects in courses, as well as small self-directed projects to apply what you're learning.

Step 3: Showcase Your New Skills

Around halfway through this first phase (say, around March or April), you should be confident enough with the basics to showcase your new skills and start getting a feel for the job market.

Update your LinkedIn and resume with the skills and projects you've worked on so far, and start applying to jobs casually—not necessarily with the goal of landing a job right away, but to practice and get feedback. Every application and (potentially interview) will teach you something.

July (Weeks 27-31)

Your First Big Project

After the first learning phase, it’s around July. Now that you’ve got a solid foundation, it’s time to tackle your first self-directed end-to-end project. Think of a project that demonstrates your skills comprehensively, such as:

- Scraping news websites and running sentiment analysis about a particular topic to track trends over time
- Predicting a genre of a song based on its lyrics
- Building a recommendation system for something that interests you, like board games or podcasts
- Creating a real-time Crypto price prediction model

Your goal at this point is to stretch yourself to apply your skills to new problems that you can add to your resume and speak confidently about in an interview.

August (Weeks 32-35)

Real-World Experience

Now that you have some experience working on projects independently, it's time to try to get some experience working with real clients. Start reaching out to family friends, local small businesses, nonprofits – anyone who might work with you – and offer your data skills for free.

For example, you might volunteer to analyze customer data for a local shop, or build a dashboard for a nonprofit. These real-world projects will boost your portfolio and build connections. We'll talk a lot more about how to do this in the How to Get Experience section.

We'll also want to ramp up job applications at this time. But, we're not going to just apply through job boards. Instead, we're going to proactively reach out to recruiters and hiring managers – more on this in the How to Get Interviews section.

Lastly, we'll use this time to start building a habit of practicing coding problems (LeetCode and SQL) so that we're ready when it's interview time.

September & October (Weeks 36-44)

The Grind

By September we should be working on a second or third major project. Ideally this should be for a real client (even for free), but if you don't hear back, continue to work on larger scale self-directed projects that simulate real-world use-cases as much as possible.

November (Weeks 45-48)

Interview Prep Mode

By around November, your skills and portfolio should be solid, so it's time to shift gears to interview prep. This is something we can do with or without a specific interview lined up.

In addition to the coding practice we've been doing, we'll also want to start thinking about preparing for case studies and presenting our work in the best possible light. I'll talk about interview prep in more detail later on in the roadmap.

December (Weeks 49-End of year)

The Finish Line

December is when all your hard work comes together. Stay consistent and keep pushing until you land that first role.

At this stage, you should still be applying to jobs daily, leveraging your network, and keeping your LinkedIn active with posts about your projects.

You're also going to want to continue working on your portfolio, practicing coding challenges, and preparing for interviews.

With persistence and the right strategy, you'll land your first data science role—and start the next year ready to hit the ground running.

Area of Focus

One of the first decisions you'll need to make on your journey is understanding the type of data science role you're aiming for. As a beginner, you might not know that Data Scientists can do very different things at different companies.

In my experience, there are three main areas of focus: Analytics, Data Science Generalist, and Machine Learning/production-focused roles.

Each has its own unique set of skills, day-to-day tasks, and career paths. Let's break down what each one involves, and some pros and cons.

1. Analytics

If you enjoy diving into data to help businesses make better decisions, an Analytics role might be the right fit. These roles focus on extracting and interpreting insights rather than building complex machine learning models. You'll often work closely with stakeholders to solve business problems through dashboards, reports, and A/B tests. These roles are generally more collaborative and less likely to have significant heads-down time.

Typical Tasks

- Analyzing trends in sales, customer behavior, or operational efficiency.
- Creating dashboards and visualizations to present data clearly.
- Running experiments to optimize business strategies.

Pros

- These roles often have a strong business impact and can be rewarding if you like problem-solving in that context.
- Typically have a shorter learning curve for beginners since they rely more on data wrangling and basic statistical skills.
- Great foundation if you're eventually interested in a broader data science role.

Cons

- Can be less technically challenging. If you're looking to dive into machine learning or advanced statistical modeling, this may not be totally satisfying for you.
- Compensation can be slightly lower than in more specialized data science or ML-focused roles.

2. Data Science Generalist

Like the name implies, data science generalist roles require a mix of skills, from data wrangling and statistics to building predictive models. If you enjoy variety and flexibility, this path allows you to work across the entire data science pipeline—from exploratory data analysis to deploying machine learning models.

Typical Tasks

- Cleaning and analyzing data to uncover patterns.
- Building and testing predictive models.
- Communicating findings through visualizations and reports.

Pros

- Offers a wide range of learning opportunities since you'll touch every aspect of the data science workflow.
- Opens up multiple career paths, allowing you to specialize later on in analytics, statistical modeling, or machine learning.
- Often requires less technical depth in math and ML but still gives you a solid, broad foundation.

Cons

- You'll need to be flexible and comfortable working on different types of projects.
- The breadth of skills required can feel overwhelming at times, especially if you're just starting.

3. Machine Learning-Focused

If the technical and engineering side of data science sounds fun to you, an ML-focused role might be the best fit. These roles often require deeper programming, math, and algorithmic knowledge, making them more challenging in some ways, but also (in my opinion) really fun.

These roles are more likely to allow you to have solo deep work time and are generally less collaborative than generalist or analytics roles.

Typical Tasks

- Developing machine learning models.
- Deploying models to production systems.
- Potentially working on advanced algorithms for computer vision, natural language processing, or applications like recommender systems.

Pros

- Highly specialized skill set that's in demand, especially in tech-driven companies, and especially right now.
- Often offers higher compensation due to the technical complexity and lack of talent to fill open roles.
- Opportunity to work at the cutting edge of technology.

Cons

- Steeper learning curve, especially for those without a technical background.
- Requires more engineering-type work, which may or may not appeal to you.

Which One Should You Choose?

For this roadmap, we'll assume you're aiming for a **Data Science Generalist** role. It's a great starting point because it provides flexibility and opens doors to both analytics and machine learning positions later.

That said, the advice in this roadmap applies across all three paths. The key difference lies in what you prioritize in your learning:

- For Analytics roles: Focus more on SQL, data visualization tools (like Tableau or Power BI), and business problem-solving.
- For ML-focused roles: Dedicate extra time to programming, advanced math, and machine learning algorithms. For more senior roles, you'll also need to understand ML System Design.

The important thing is to start somewhere. As you gain experience, you'll naturally discover which path excites you the most.

With your focus decided, let's move on to the core skills you'll need to succeed.

What to Learn

There are different paths to learning the skills to become a Data Scientist. You could go the traditional route with a degree, join a bootcamp, or teach yourself.

For this roadmap, **I'm assuming you're going the self-taught route**—it's often the fastest, the most flexible, and obviously the least expensive. But even if you're pursuing a degree or bootcamp, this guide will help you fill in the gaps. Because trust me, no matter what formal program you're in, there's always extra self-study involved.

That said, **having an advanced degree can help**. It won't guarantee you a job, but it won't hurt either—resume scanners love seeing Master's and PhDs, and without them you might struggle to be considered for some roles. Nowadays there are many low-cost part time online programs, so if traditional education appeals to you, it's something to consider.

Still, some Data Scientists break into the field without an advanced degree, so don't let this hold you back.

Should You Get Certificates?

There are many paid online certificate programs available for Data Science. **While many of them are high quality and the certificate certainly can't hurt your chances of getting hired, don't feel stressed if you can't afford them.** You can absolutely demonstrate your skills through self-directed, project-based learning as well, at zero cost.

Now, **a big question is where to start: With math or coding.**

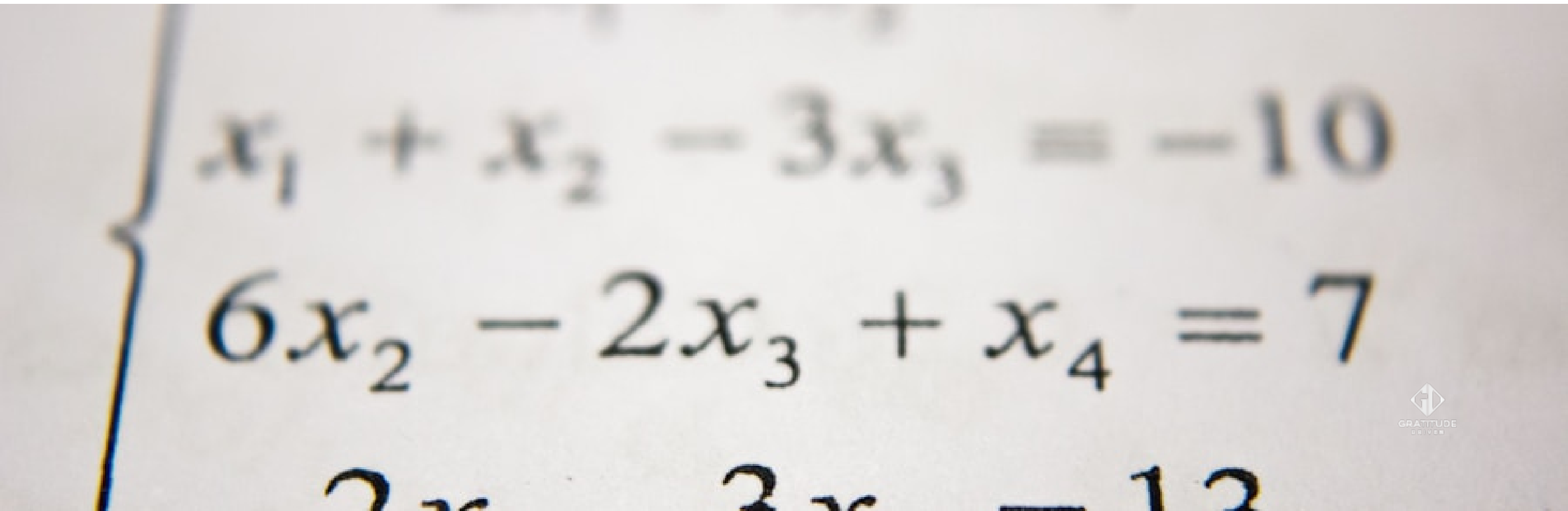
Here's my take—**start with both and use each to learn the other.**

I think this is a good approach because they're equally essential, and if you don't enjoy either, this might not be the field for you, and it's important to figure that out before you dedicate a ton of time to something you don't actually like doing.

But don't worry, you don't need to master calculus or become as good at coding as a Software Engineer right away. The idea is to start with the basics of both and build them up together as you go.

Here's how we're going to do it:

- Spend your first month learning the basics of both math and coding. For example, study Python for data analysis while brushing up on essential statistics.
- As you progress, pair them together. When you learn about a concept like regression in statistics, implement it in Python. This integrated approach will make the material more practical and easier to understand.



A Note on Effective Studying

Data science concepts can be challenging to understand. AI tools like ChatGPT can be super helpful for learning, debugging, and clarifying concepts—but don't let them do all the work.

The straining, slightly uncomfortable feeling of wrapping your brain around a concept is exactly how you grow. If something's hard, lean into it. Use ChatGPT to help, but always aim to understand the *why* behind the answers, and let yourself struggle a bit before you just ask ChatGPT to explain it to you right away. Because sometimes it won't be able to, and you need to have the skill and mental endurance to figure things out on your own.

With that mindset, let's jump into the specific skills you need to prioritize, starting with Python.

Python

Python is the main coding language used for data science, and for good reason—it's versatile, beginner-friendly, and it has a massive ecosystem of libraries tailored for data analysis and machine learning.

As a Data Scientist, you'll spend a lot of time coding, whether it's cleaning datasets, building models, or automating workflows. While some companies may use R, Python dominates the field. The good news is that once you've mastered Python, transitioning to R if you need to—or any other language—isn't that hard.

To get started with Python, we're first going to learn the very basics:

- Variables, data types, loops, conditionals, and functions.
- Understand how to work with data structures like lists, dictionaries, sets, and tuples.
- Know how to handle files (read/write operations) and exceptions.

Next, we'll need to understand package management systems like pip and conda to install and manage dependencies for your projects.

Then, we'll want to get familiar with the main libraries used by Data Scientists:

- Pandas: For data manipulation and basic plots.
- Numpy: For numerical computations.
- Scikit-learn: For building machine learning models.

The goal here isn't to memorize every function but to become comfortable with the tools and learn how to reference documentation effectively when you're stuck.

Learning Resources

Don't get too hung up deciding which Python course to take. Any reputable beginner course will do—just pick one and stick to it. Some great options include:

- Code Academy
- Python for Data Analysis e-book
- Udemy courses (wait for sales/use coupon codes for Udemy)

Bonus

The above is enough to get started. But if you want to stand out, I highly recommend the book Software Engineering for Data Scientists (affiliate link) to level-up your coding and software development skills.

Timeline

Plan to spend about four weeks mastering the basics of Python that we went over.

Example Projects

To solidify your learning, here are some example projects to put things into practice:

Write a program to process words in a text file

- Read in some data.
- Count the frequency of each word and identify the top 10 most frequent words.
- Plot the top words.
- Save the results into a new file.

Create a basic calculator tool

- Write functions for addition, subtraction, multiplication, and division.
- Take user input and perform the selected operation.
- Add exception handling to manage invalid inputs or division by zero.
- Print the results.

Probability & Statistics

At the same time, we're going to start studying statistics. As a Data Scientist, you'll use statistics in nearly every aspect of your work—from analyzing datasets to evaluating machine learning models, and even designing experiments. The better your grasp of statistics, the better your ability to interpret results and not make huge mistakes that cost the company millions!

Just kidding. Kind of. We really do want to know what we're talking about when we make decisions based on data.

Just like with Python, start with the basics and build your knowledge over time. Here's a roadmap of some key topics:

1. Descriptive Statistics

- Understand measures of central tendency: mean, median, and mode.
- Learn measures of dispersion: like variance and standard deviation.
- Explore common distributions like the normal distribution, skewed distributions, and uniform distribution.

2. Data Characteristics

- Differentiate between continuous vs. discrete variables and nominal vs. ordinal data.
- Learn how to identify and handle outliers.

3. Correlation and Multicollinearity

- Understand the difference between correlation and covariance.
- Learn about multicollinearity and its implications for regression models.

4. Inferential Statistics

- Understand hypothesis testing, including p-values, z-tests, t-tests, and chi-squared and ANOVA.
- Dive into basic probability concepts, including Bayes' Theorem and randomization.
- Understand confidence intervals.
- Learn about sampling techniques, resampling methods, and common statistical biases.

5. Experimentation Basics

- How to determine sample size.
- How to decide how long to run the test.
- Learn about the importance of randomization in experiments.

Bonus

Explore causal inference techniques to evaluate cause-and-effect relationships when we can't run an experiment.

Learning Resources

- This Udemy course is a super gentle introduction (wait for sales/use coupon codes for Udemy)
- StatQuest (YouTube channel): Fantastic for breaking down complex topics in an easy-to-understand way.
- Khan Academy: Offers a structured and beginner-friendly approach to core statistics topics.
- Practical Statistics for Data Scientists book (affiliate link): A great hands-on resource that ties statistical concepts directly to data science applications in R and Python.

Timeline

Dedicate three-four weeks to studying statistics alongside Python (so, it'll take around eight weeks total to learn the basics of both). Spend the first week focusing on descriptive statistics and data characteristics, then move on to inferential statistics and experimentation in the following weeks.

Example Projects

- Download a Kaggle sales dataset and analyze the distribution of sales in each region and over time.
- Download a survey dataset from Kaggle and analyze stats for question answers, find outliers, and compare answers across groups (demographic or age etc.)

SQL

Next, we're introducing another coding language: SQL. No matter where you work, you'll almost certainly use SQL to interact with data, because this is the language we use to query data from a database. While some smaller organizations may rely only on CSV files, most companies use relational databases to store their data, and SQL is the primary tool for accessing and manipulating it.

Luckily, it's pretty easy! Especially if we just focus on the core concepts we need to get started in data science:

1. Core Querying Skills

- Learn how to write queries using SELECT, WHERE, ORDER BY, and LIMIT.
- Understand how to deal with NULL values effectively to avoid errors or unexpected results.
- Get comfortable with handling dates and timestamps.

2. Intermediate Features

- Master JOINS: Understand the differences between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN.
- Learn to use UNIONs and self-joins for more complex queries.
- Explore CASE statements to add conditional logic to your queries.
- Understand GROUP BY and aggregate functions (e.g., COUNT, AVG, SUM, MIN, MAX) for summarizing data.

3. Advanced Techniques

- Dive into Common Table Expressions (CTEs) to break down complex queries into manageable pieces.
- Learn how to write and optimize subqueries.
- Explore window functions like RANK, ROW_NUMBER, and LAG to perform advanced analytics within a dataset. This will almost definitely come up in interviews, so don't skip it.

4. Query Optimization

- Learn basic strategies to make your queries run more efficiently.

Learning Resources

- Any intro YouTube course or Udemy Course. I took this one on Udemy.
- CodeAcademy
- W3Schools
- SQLBolt
- LeetCode

Timeline
Dedicate two weeks to learning the basics of SQL.

Example Projects

For SQL, just work through practice problems in the interactive hands-on learning platforms linked above.

Tableau/Power BI/Data Visualization

As a Data Scientist, a lot of your job is about telling a story with data. Data visualization tools like Tableau, Power BI, and Python data visualization libraries allow you to transform raw data into insights that non-technical stakeholders can easily understand.

While you don't need deep expertise in these tools to get hired, being familiar with them and having them listed on your resume is useful.

For your first role, focus on the fundamentals of data visualization, both with dedicated tools and programming libraries:

1. Basics of Tableau and Power BI

- Connect to a dataset and explore the interface.
- Create basic charts like bar plots, line graphs, and scatter plots.
- Combine visuals into dashboards and add interactivity (e.g., filters or slicers).

2. Python Visualization Libraries

- Generate static and interactive visualizations for data exploration.
- Customize plots with titles, labels, and legends.

Learning Resources

- Intro to Tableau
- Intro to Power BI
- Intro to Python visualization libraries

Timeline

Spend one week learning the basics of Tableau, Power BI, and Python-based visualization libraries.

Example Projects

1. *Sales Dashboard*

- Use a sales dataset from Kaggle.
- Create bar charts to show sales performance by category.
- Add line charts to track sales trends over time.
- Use filters for interactive analysis by region or time period.
- Highlight top-performing regions and products.

2. Customer Segmentation Visualization

- Use a customer dataset with demographics, purchase history, or behavioral metrics.
- Visualize clusters of customers using scatterplots (e.g., spending vs. frequency).
- Create demographic breakdowns using pie charts and bar charts.
- Add slicers to explore different customer segments.
- Add a KPI card showing key metrics like average spend per customer.

3. Movie Reviews

- Use a movie dataset.
- Create histograms to show the distribution of ratings.
- Use scatter plots to compare ratings with budget.
- Highlight trends in average ratings over time using line plots.
- Annotate key points or outliers.

Git

As we learn, we need to have some way to save our code and projects, and to share it with others once we build some portfolio-worthy stuff. To do that, we'll use Git and GitHub.

Git is a tool for collaboration and keeping track of your work. Whether you're working on a solo project or as part of a team, Git helps you save, organize, and track changes in your codebase. It also makes it easy to experiment without fear of losing work, and to share your progress with others. GitHub is the UI for us to interact with Git.

What To Know

- Repositories (Repos): These are like folders for your project that track all changes to files.
- Version Control: Learn how Git tracks changes to your files and allows you to move back to previous versions if needed.
- Branches: Use branches to experiment with new features or ideas without affecting your main codebase.
- Merging: Combine branches back into the main branch once your changes are ready.
- Conflict Resolution: Handle situations where multiple people make changes to the same part of a file.
- How to use the GitHub platform to interact with Git.

Basic Commands to Master

`git init` - Create a new repository.

`git add` - Stage changes for a commit.

`git commit` - Save a snapshot of your project.

`git push` and `git pull` - Share changes with others or pull updates from a shared repo.

`git clone` - Copy an existing repository.

`git branch` and `git merge` - Create and merge branches.

Learning Resources

- Learn Git Branching
- Game to learn (Oh My Git)
- Data Camp
- CodeAcademy
- W3 Schools

Timeline

Spend ½ week learning the basics of Git.

Example Projects

Start by creating a personal project repository and practicing commands. As you work on projects, use Git consistently to develop good habits.

Check-in

At this point, you've been studying for about 2.5 months. If you started in January, we're now around mid-March, and you've already built a strong foundation in Python, Statistics, SQL, and Data Visualization. Here's the neat part: **You now have the skills to be a Data Analyst.**

This doesn't mean you've completed your journey to becoming a Data Scientist, but it does mean you're ready to start marketing yourself as someone who's job-ready now. So at this point we're going to take some time to start putting yourself out there in the job market.

1. Update Your Resume and LinkedIn

- Add all the skills you've learned to your skills sections.
- Add a section for your projects, briefly describing what you did, the tools you used, and the outcomes.
- Make sure your LinkedIn photo is professional and approachable.
- Update your resume summary and LinkedIn headline/description to use keywords from jobs you'd like to target. Market yourself as already being the thing you want to be. So, your headline shouldn't say, "Aspiring Data Analyst," just Data Analyst.

2. Update GitHub

Make sure your GitHub is super clean with all your work so far (we'll talk more about this in the Marketing Your Experience section).

3. Start Applying to Jobs

While it may seem a little early, we're going to start applying to jobs now.

The idea here isn't really to get a job now. If that happens, that's awesome, but the idea is more to start getting a feel for what job descriptions look like, the skills they're looking for, keywords to target, and how to tailor your resume and cover letter to these roles.

Applications at this point are more information-gathering than anything.

Now, back to studying.


LAUREN CH
Digital Marketing Sp
CONTACT
Phone
(212) 256-1414
Address
Chicago, IL 60622
Email
lauren.chen@gmail.com
LinkedIn
linkedin.com/in/lauren-chen/
EDUCATION
Bachelor of Arts,
Communications
New York University,
New York, NY
Honors: cum laude (GPA: 3.6/4.0)
May 2017
RELEVANT SKILLS
Digital Data Analytics
Digital Marketing
Adobe Photoshop
Adobe Illustrator
Adobe InDesign
AutoCAD
Rhinoceros
Microsoft Office
Slack
Salesforce
Digital Marketing Specialist with 4+ years of experience in online marketing,
branding, and business strategy across music, media, and entertainment
industries. Skilled in evaluating financial needs and implementing multi-pronge
digital strategies that increase revenue and drive brand growth.
PROFESSIONAL EXPERIENCE
Digital Marketing Specialist
2019 – Present
Triangle Music Group, New York, NY
Manage digital sales and streaming accounts to improve brand positioning and growth
Source and develop new strategic partnerships, social engagements, and advertising opportunities that generate new revenue streams
Collaborate with internal departments to execute national advertising campaigns, plan global digital distribution, and re-deploy a 1M+ consumer sales and marketing database
Led the concept and launch of multiple crowdfunding campaigns for priority artist releases, resulting in new revenue of $80K+
Digital Marketing Associate
2019 – Present
Momo Software, New York, NY
Worked with management to develop and apply digital marketing plans with a focus on driving acquisition and conversion
Devised and implemented robust digital acquisition plans, ensuring precision in financial reporting, budgets, and forecasts
Increased conversions by 15% from paid sources (PPC, Grant, Display, and VOD)
Enhanced conversion rates by 12% via A/B testing landing pages for a better performing conversion funnel
Marketing Intern
2016 – 2017
Kingston Digital, New York, NY
Helped research, write, and edit blog posts for Kingston's website
Determined relevant keywords and entities for pages using Semrush, Ahrefs, and Page Optimizer Pro
Gathered and analyzed data from social media PPC campaigns
GRATITUDE

Math for Machine Learning

Math is often the part of data science that intimidates people the most. The truth is, you don't need an advanced understanding of calculus or linear algebra to get started in the field. While a deeper grasp of these topics is useful for understanding machine learning algorithms and optimization techniques, it's not critical to be an effective practitioner in your early career. Many Data Scientists – myself included – developed a more in-depth understanding of math on the job as their work demanded it.

That said, having some basic math fundamentals under your belt will make it easier to understand key concepts like gradient descent, optimization, and machine learning in general.

Linear Algebra is important for understanding how data is represented and manipulated in machine learning models (think vectors and matrices). To get started, we need to understand vectors, matrices, matrix operations (addition, multiplication), and the concept of dot products.

Calculus plays a role in optimization, such as understanding how gradient descent works to minimize error in machine learning algorithms. At the beginning, focus on derivatives and partial derivatives—enough to understand how gradient descent works—and learn the concept of gradients and optimization.

We are not going to spend time understanding how to calculate this stuff by hand. Instead, we're just going to get an **intuition** for what's going on.

Learning Resources

Luckily, the perfect resource for this already exists: Just watch 3blue1brown's Essence of Linear Algebra and Essence of Calculus playlists.

If you want to dig deeper after that, I'd recommend:

- Deeplearning.AI Coursera Specialization (affiliate link)
- Imperial College London Coursera Specialization (affiliate link)
- Khan Academy: A classic resource for understanding the basics of calculus and linear algebra in a beginner-friendly format.
- Math for ML Book

Timeline

Plan to dedicate two weeks to learning these math fundamentals. Again, focus on understanding the intuition of what is happening vs. the calculations themselves. You can always come back to the math later in your career.

Machine Learning

Now we're ready to get started with the most fun part, in my opinion. Machine learning!

At its core, data science is about solving problems using data. Machine learning is one of the most powerful tools in your toolkit because it enables you to uncover patterns and make predictions from data.

Before diving into the specific algorithms, there's a lot we need to understand first:

- **Data Cleaning**: Learn how to fix messy data, handle missing values, and deal with inconsistent entries.
- **Feature Engineering**: Turn raw data into useful inputs for your model by creating features.
- **Bootstrapping**: Understand how to use sampling to estimate the properties of your data.
- **Feature Preparation**: Handle common issues like scaling, imputing missing values (NULLs), and normalizing your features.
- **Train-Test Split** and Avoiding Data Leakage: Keep your training and testing data separate to ensure your models generalize well.
- **Exploratory Data Analysis (EDA)**: Know how to summarize, visualize, and explore data to spot trends and outliers.
- **Supervised vs. Unsupervised Learning**: Learn the difference between working with labeled data vs. finding patterns in unlabeled data.
- **Classification vs. Regression**: Understand when to predict categories (classification) vs. continuous numbers (regression).
- **Metrics**: Master regression metrics (e.g., MAE, RMSE) and classification metrics (e.g., accuracy, precision, recall, F1, and AUC score).

Once you have some of this background, you can get started with some of the **fundamental algorithms**:

- Linear regression
- Logistic regression
- Decision Trees
- K-means Clustering

Learning Resources

- Andrew Ng's classic ML Specialization on Coursera (affiliate link)
- A Gentle Introduction to Machine Learning StatQuest Playlist
- DataCamp Machine Learning Fundamentals in Python
- Google's Machine Learning Crash Course

Timeline

Expect to spend about four weeks learning the fundamentals.

Example Projects

Work through all of the projects in the courses.

Then, we're ready to move on to some more advanced concepts to help us handle real-world challenges.

- **Overfitting**: Understand what it is and how to tackle it.
- **Regularization**: Learn techniques like ridge and lasso regression to keep models from overfitting.
- **Cross-Validation**: Validate your models properly by splitting data into multiple training/testing sets.
- **Principal Component Analysis (PCA)**: Use PCA to reduce dimensionality and focus on the most informative aspects of your data.
- **Hyperparameter Tuning**: Learn how tweaking model settings can boost performance.
- **Ensemble Models**: Get familiar with random forests and gradient boosting models (e.g., XGBoost).
- **Time Series Basics**: Learn how to analyze data over time, focusing on trends, seasonality, and simple forecasting methods.
- **Introduction to Deep Learning**: Understand the basics of neural networks, how they work, and when to use them.

Learning Resources

- Time Series models
- Gradient Boosting
- Deep Learning:
  - 3Blue1Brown Learning Intro
  - Andrew Ng's Deep Learning Specialization (affiliate link)
- Bonus for ML specialists: Designing Machine Learning Systems (affiliate link)

Timeline

Expect to spend about four more weeks getting familiar with these concepts. You won't be an expert by any means, but this is kind of a survey of what exists right now that you could use as tools in future projects.

Example Projects

To put your learning into practice, do all of the exercises in the courses. We'll turn to self-directed projects soon.

Cloud Platforms

Speaking of things you should be familiar with but definitely won't have time to master, we need to learn a little bit about cloud platforms.

Cloud platforms like AWS, Google Cloud Platform (GCP), and Microsoft Azure are essential tools for modern data science. They allow you to work with large-scale data, deploy machine learning models, and manage resources efficiently. While you won't need to become an expert in all of them, you do need to know what they do and how to use key features in your workflow.

AWS is the most popular, so unless you already know you're targeting a company that uses GCP or Azure, just focus on AWS. The concepts are similar with the others and you can always learn them later.

What to Know

- Storage and Databases: Learn how cloud platforms handle data storage (e.g., Amazon S3) and how to access data securely.
- Compute Services: Understand how to use cloud resources to run large-scale computations or train models (e.g., AWS EC2).
- ML Services: Get familiar with cloud-based tools like AWS SageMaker for training, deploying and managing machine learning workflows.

Learning Resources

- Fully-Managed Notebook Instances with Amazon SageMaker - a Deep Dive
  SageMaker playlist
- AWS SageMaker For ML And DL Tutorial Playlist

Timeline

- Spend one week exploring the basics. Start with free-tier accounts on AWS, GCP, or Azure to get hands-on experience.
- Dive deeper during projects when you need to deploy models or work with cloud-based data storage and analytics.

Example Projects

Walk through example tutorials in your own environment to get started. Then, use cloud resources in your self-directed end-to-end projects (next learning phase!)

Business Logic

Having strong business intuition is also crucial for Data Scientists. It's not enough to build models—you need to work on projects and interpret results in ways that drive business impact. This is usually tested via case studies, which are designed to test your ability to think critically about business problems, identify key metrics, and propose actionable solutions. These skills are essential for success in interviews, and obviously on the job where business impact is ultimately the thing that will drive your career forward.

What to Know

1. Understand common causes for changes in metrics, such as seasonal trends, product changes, or external factors.
2. Break down metrics into components (e.g., customer acquisition vs. retention) to pinpoint areas of change.
3. Key Metrics for Different Business Types:
   - 3.1. E-commerce: Conversion rates, average order value (AOV), customer lifetime value (CLV).
   - 3.2. SaaS (Software as a Service): Monthly recurring revenue (MRR), churn rate, user engagement.
   - 3.3. Marketing: Return on investment (ROI), click-through rate (CTR), cost per acquisition (CPA).
   - 3.4. Operations: Efficiency metrics, downtime, capacity utilization.
4. How to Answer Case Studies. We'll talk about this in the interview section.

Learning Resources

- YouTube mock interviews
- InterviewQuery has practice questions where you can read others' answers.

Timeline

- Spend ½ week understanding the basics of business metrics and structured problem-solving. Basically, just get your mind thinking about these kinds of things in general. It will take a lot longer to get good at business intuition and answering case study questions.
- Dedicate additional time to practice and preparation before each interview. Regularly work on mock case studies to refine your approach and gain confidence.

LeetCode

The last thing on our roadmap is “LeetCode-style" coding.

If you’re not sure what that means, basically it’s a type of coding question where you’re given algorithmic challenges to solve. These challenges test your understanding of data structures (like arrays, linked lists, and trees), algorithms (such as sorting, searching, and dynamic programming), and computer science fundamentals.

I personally despise LeetCode for data science interviews, because in my opinion it’s completely irrelevant for the kind of coding we do on the job. Data science typically involves wrangling messy data, building predictive models, and implementing data pipelines — not designing the most optimal binary search algorithm. **But the fact is, a lot of companies still use it, so you need to know how to pass the interview.**

That being said, for most data science interviews it’ll just be questions on arrays and strings, so we don’t need to go into software engineer-level learning on data structures and algorithms.

What I’d suggest learning is:

- Basic familiarity with Big-O notation.
- Common edge cases and how to approach coding problems in general by asking clarifying questions.
- String and Array questions.
- Practice coding and speaking at the same time (not easy! You need to really practice this).

Also, quick note that data science interviews also often test SQL, so don’t forget to practice that as well.

Learning Resources

- Big-O Notation
- NeetCode
- Grokking the Coding Interview Patterns in Python
- You can also just walk through the LeetCode arrays questions and ask ChatGPT to help you understand the “tricks” and optimal solutions.

Timeline

Dedicate about ½ week to familiarizing yourself with LeetCode and solving introductory problems. Then, **start a habit of practicing one Python and one SQL question every day until you get a job.**

Summary of Skills to Learn

Primary Topic	Specific Skills	Time to Learn
Python	- Basics: Variables, data types, loops, conditionals, functions	4 weeks
	- Data structures: Lists, dictionaries, sets, tuples	
	- File handling, exceptions	
	- Libraries: Pandas, Numpy, Scikit-learn	
	- Package management: pip, conda	
Statistics	- Descriptive statistics: Mean, median, variance, distributions	4 weeks
	- Inferential statistics: Hypothesis testing, p-values, confidence intervals	
	- Experimentation: A/B testing, sample size, randomization	
	- Probability: Bayes' theorem, basic concepts	
	- Multicollinearity, correlation, causation	
SQL	- Core querying: SELECT, WHERE, ORDER BY, LIMIT, NULL handling	2 weeks
	- Joins: INNER, LEFT, RIGHT, FULL OUTER	
	- Aggregations: GROUP BY, COUNT, AVG, SUM	
	- CASE statements, CTEs, window functions, subqueries	

Summary of Skills to Learn

Primary Topic	Specific Skills	Time to Learn
Data Visualization	- Python: Matplotlib, Seaborn, Plotly	1 week
	- Dashboards: Tableau, Power BI	
Git	- Basic commands: init, add, commit, push, pull, clone	0.5 weeks
	- Branching and merging	
	- GitHub: Repository setup, documentation	
Math for ML	- Vectors, matrices, dot products, matrix operations	2 weeks
	- Derivatives, partial derivatives, gradients and gradient descent	
Machine Learning	- Fundamentals: Regression, classification, supervised/unsupervised learning	8 weeks
	- EDA, data cleaning, feature engineering, train-test split	
	- Metrics: RMSE, F1, accuracy, recall, precision	
	- Fundamental aglorithms: Regression, decision trees, k-means	
	- Regularization, cross-validation, hyperparameter tuning	
	- Ensemble models, time series, deep learning	

Summary of Skills to Learn

Primary Topic	Specific Skills	Time to Learn
Cloud Platforms	- Storage: e.g. S3	1 week
	- Compute: e.g. EC2	
	- ML Services: e.g. SageMaker	
Business Logic	- Metric analysis: KPIs, A/B testing	0.5 weeks
	- Industry-specific metrics: SaaS, e-commerce, marketing	
LeetCode Practice	- Arrays, strings, Big-O, edge cases	0.5 weeks and ongoing
	- SQL challenges	

Extra Credit

There are some more things that I think will be really valuable to you in your first job, but aren't essential. If you have time and curiosity, I would also recommend learning about:

- How to test your code.
- Object-Oriented Programming.
- Code Reviews: Understand how to give and receive constructive feedback on code.
- Docker & Packaging: Learn the basics of containerization and how to make your projects portable.
- Airflow: For building data pipelines.

If you read **these two books,** you'd be off to a very solid start on the above (and more!):

- Designing Machine Learning Systems (affiliate link)
- Software Engineering for Data Scientists (affiliate link)

Things NOT to Learn

We've covered A LOT. But there is so much more! Here are things I would skip for now:

- **Database Creation and Optimization** (e.g. indexing, sort keys, etc.). This is absolutely good to know, but not critical for entry-level Data Scientists).
- **Deep Learning for Niche Applications**: Working with GANs, transformers, or advanced architectures is typically more relevant for specialized roles in computer vision or NLP.
- **Reinforcement Learning**: This is highly specialized and rarely required for most data science generalist roles.
- **Hadoop & Apache Spark**: While useful for large-scale data processing, most Data Scientist roles don't require this, and there are typically software engineers to help (or simpler services to get the job done).
- **Kafka**: Real-time data streaming tools are more relevant for engineering-focused roles.
- **Kubernetes**: While container orchestration is essential for large-scale deployment, most entry-level Data Scientist roles don't involve heavy deployment responsibilities.
- **NoSQL Databases**: Databases like MongoDB, Cassandra, and DynamoDB are good to know but less critical than mastering SQL for relational databases.
- **Graph Databases**: Neo4j and similar tools are great for analyzing network data but rarely required in typical Data Scientist roles.
- **Edge Computing and IoT Analytics.**
- **Nitty gritty on how LLMs work**. It's a good idea to get some experience with prompting and working with APIs, but you're unlikely to be asked to code a Transformer from scratch in an entry-level data science generalist role.

GRATITUDE DRIVEN

How to Get Experience and Build a Portfolio

By now, you've gained foundational skills in Python, statistics, SQL, machine learning, and more, and have completed a few small follow-along projects. But here's the next challenge: Hiring managers want to see real-world experience. How are you going to get experience, if no one will hire you without experience?

What we're going to do is create our own opportunities. Here's how to do it step-by-step.

Building a Portfolio

1. Align Your Projects with Your Career Goals

At this point you should be pretty familiar with the kinds of jobs that are out there, since you've been casually applying for a few months. Align your project plan with the expectations of the industries you're targeting. For example:

- E-commerce: Recommendation systems, customer segmentation, or sales forecasting.
- Finance: Fraud detection, credit risk modeling, or portfolio optimization.
- Healthcare: Predictive analytics for patient outcomes or medical image analysis.
- Tech: Sentiment analysis of social media, churn prediction, or A/B test analysis.

Also, consider the kind of role you're aiming for:

- Analytics roles: Highlight SQL, dashboards, and data visualization.
- Machine learning roles: Focus on building and deploying predictive models end-to-end.
- Generalist roles: Showcase versatility with a mix of data wrangling, analysis, and modeling.

2. Choose Projects That Showcase a Full Workflow

To stand out, your portfolio projects should **demonstrate an end-to-end data science workflow**. That means showing your ability to:

- Define the problem: Start with a clear, actionable question or goal.
- Collect and clean data: Source raw data through web scraping, APIs, or public datasets. This is super important! We don't want to have a project with an unrealistically-clean dataset. It should be something a little more messy and representative of the real-world.
- Perform Exploratory Data Analysis (EDA): Use visualizations and summary statistics to uncover patterns and outliers.
- Engineer features: Create meaningful inputs for your models through scaling, encoding, combining data, and handling missing values.
- Build and evaluate models: Train, test, and validate models with clear metrics. Highlight how your model solves the problem effectively.
- Communicate insights: Translate findings into actionable recommendations, using visualizations or dashboards. Consider using Tableau Public, Power BI, or Streamlit to create interactive visuals and shareable dashboards.
- Optionally, deploy the model!

3. Pick a Project Idea

Now that you know all the pieces that should go into a project, it's time to pick a specific project to work on. Ideally, **pick something that is actually interesting to you**. You're going to be thinking about it for weeks, and you're going to do a better job if it's fun and showcases your personality.

If you're not sure where to start, here are some project ideas:

Public Data Analysis

- Social Media: Use Twitter, Reddit, or YouTube APIs to analyze trends, sentiment, or user behavior.
- Economic Data: Analyze public datasets like World Bank or Census data to identify trends in GDP, unemployment, or urbanization.
- Weather Data: Use NOAA or OpenWeatherMap APIs to analyze patterns and build forecasting models.

Recommendation Systems

- Collaborative Filtering: Build a movie, book, or product recommendation system using user-item interactions.
- Content-Based: Recommend items based on attributes like product features or user preferences.

Sentiment Analysis

- Text Analysis: Analyze customer reviews, social media posts, or product feedback to assess sentiment or extract insights.
- Topic Modeling: Use NLP techniques to identify themes in large text datasets.

Predictive Models

- Churn Prediction: Predict customer churn for a subscription service or product.
- Fraud Detection: Identify fraudulent transactions in financial data.
- Time-Series Analysis: Build forecasting models for stock prices, sales data, or environmental variables.

4. Present Your Project

Once the project is complete, ensure it's presented in a professional and engaging way.

It's up to you whether you'd rather have a separate professional website for your portfolio, or just make your GitHub look really nice. Also, every time you finish a project, make sure to add it to your resume and LinkedIn. In the next section on marketing your skills I'll go way more into detail about this.

Now you have demonstrated you can actually do some stuff! **It's time to use those skills to get real-world experience.**

Real-World Project

My favorite “hack” for getting real-world experience is to offer your services for free to real clients.

I realize this is controversial. Yes, in an ideal world we'd all be paid for our time, but the fact is that as a beginner your options are limited. It's worth it to invest your time for a big payoff once you transition into the field.

These projects are valuable because they allow you to work with actual data, address real business problems, and showcase your ability to apply data science techniques in practical scenarios. Here’s how you can take the initiative to gain this experience:

Start Where You Are

If you’re currently employed, look for ways to apply data science within your organization, even if it’s outside your official role. For example:

- If you work in retail, analyze sales trends to optimize inventory management.
- In a small office setting, automate repetitive tasks or analyze performance metrics.
- If you’re in customer service, explore feedback data to identify trends or improve satisfaction scores.

These internal projects not only add to your portfolio but also demonstrate initiative, problem-solving skills, and an ability to add value in a business context. Often, success in these projects can lead to formal recognition, such as transitioning into a more analytical role within the same company. **Just make sure you’re following all the rules about handling this data.**

Leverage Your Network

If you aren't currently employed or your current job doesn't provide opportunities for data science work, there are still plenty of options.

Start with people you already know—friends, family, or acquaintances—who may own businesses or manage teams. For instance:

- Help a friend's family-owned restaurant analyze customer reviews to improve service.
- Build a pricing optimization model for someone running an e-commerce shop.
- Develop a tool to track and predict seasonal trends for a local business.

Pitch to Small Businesses or Nonprofits

If you don't know anyone, reach out to local businesses and nonprofits.

Small businesses and nonprofits often lack the resources for advanced analytics but can benefit significantly from your skills. Reach out to these organizations with a specific, actionable proposal. For instance:

- Offer to create a dashboard to track donation trends for a nonprofit.
- Analyze sales and customer data for a local store to help optimize marketing efforts.
- Develop a basic predictive model for a startup to improve customer retention.

When pitching, focus on the business impact of your work. Most small organizations will not fully understand data science concepts, so frame your proposal in terms of how it will save time, reduce costs, or improve efficiency. Even if these projects start as unpaid, they often lead to recommendations, paid opportunities, or at the very least, excellent additions to your portfolio and resume.

Volunteer with Existing Organizations

Organizations like DataKind are excellent ways to gain real-world experience while contributing to meaningful causes. These platforms connect Data Scientists with nonprofits in need of expertise. Projects range in scope from figuring out survey design to developing computer vision models.

Not only do these experiences enhance your technical skills, but they also allow you to learn from more experienced Data Scientists and grow your network.

Keep Building Independently

While real-world projects with clients or businesses are ideal, don't let the lack of immediate opportunities stall your progress. If responses from potential clients are slow, keep developing portfolio projects on your own.

The goal is to maintain momentum, refine your skills, and add depth to your portfolio while waiting for opportunities to come through.

Marketing Your Experience

If you've been following the year-long timeline, but now it’s around mid-August. You’ve learned the basics and even put it to use on some self-directed and/or real-world projects. Now it’s time to focus on updating your application materials (again) to reflect your new skills and position yourself as a strong candidate in the data science field.

Portfolio

First, your portfolio. I mentioned before that you can choose to have a website or just a nicely-formatted GitHub. Personally, I'd just go with Github. Here are some awesome examples:

- https://github.com/katiehuangx
- https://github.com/TeneikaAskew
- https://github.com/Erlemar

These all have a clear and engaging introduction, and easily direct the reader to projects.

If you've set up a professional website, make sure it includes:

- A brief but engaging bio
- Links to your GitHub and LinkedIn
- Clear sections highlighting your projects with descriptions of the problem, your approach, and results
- Contact information

Regardless of which approach you take, each project should have its own clean, organized GitHub repository for each project, including:

- A README with the project overview, tools used, and key findings.
- Well-commented code and modular scripts.
- Instructions for reproducing the project (if applicable).

One thing to highlight in your write-ups is *why you made the decisions you did during the project*. For example, why did you choose a specific model or pre-processing approach? This will help prepare you for interviews and show that you understood the concepts well enough to make good decisions.

It's also useful to **write about your project on Medium or LinkedIn to explain your process and results**. You'll want to focus on storytelling: what problem you solved, how you solved it, and why it matters. Remember, your communication skills are also important for data science, so creating an engaging narrative also highlights an important skill for future employers.

LinkedIn

Next, we're going to make sure your LinkedIn looks good. We've already done some of this in the mid-learning check-in, but here's a checklist to make sure everything is polished and optimized:

- Custom URL: Create a clean and professional LinkedIn URL (e.g., linkedin.com/in/yourname).
- Professional Photo: Use a high-quality photo with a friendly and approachable expression.
- Headline: Present yourself as already being in the field, such as "Data Scientist | Machine Learning Enthusiast," rather than "Aspiring" or "Student."
- Skills: Add a comprehensive list of data science and related skills. Make sure to update this with things you've learned since last time, and keywords you've noticed in relevant job postings.
- Projects: List key projects with concise descriptions and links to your portfolio or GitHub.
- Profile Summary: Use keywords and action-oriented language to describe your expertise, skills, and career aspirations.
- Links: Include links to your GitHub, portfolio website, and relevant articles or blog posts.
- Desired Roles: Add titles and industries you want to be contacted for, such as "Data Scientist," "Machine Learning Engineer," or "Data Analyst."
- Experience Titles: If you've done unpaid or freelance work, list those projects as legitimate roles. For example, label yourself as a "Data Scientist" even if it was an unpaid gig.
- Stay active: Share relevant projects, articles, or insights regularly to keep your profile active and increase visibility.

Some good LinkedIn examples:

- https://www.linkedin.com/in/katiehuangx/
- https://www.linkedin.com/in/teneikaaskew/

Resume

When updating your resume, keep it concise and tailored:

- Place relevant technical skills at the top.
- Highlight your projects, especially real-world ones, with specific details about your contributions, tools used, and measurable outcomes.
- If you worked on a freelance or volunteer project, list your title as "Data Scientist" or "Data Analyst" (just like we did on LinkedIn)
- Study job descriptions for roles you want and ensure your resume includes relevant keywords and phrases.
- Add links to your GitHub, portfolio, and LinkedIn.
- Present unrelated experience in terms of transferable skills—e.g., problem-solving, teamwork, or leadership.

A strong beginner data science resume should have:

- One page only
- A clear summary or heading at the top that aligns with your career goals (e.g., "Data Scientist | Machine Learning Engineer")
- Real-world projects and volunteer work
- Skills, tools, and technologies used in projects listed in your experience
- Links to your portfolio, GitHub, and LinkedIn

This advice is somewhat specific to the U.S., but the core ideas remain the same even if you live in a country that expects longer CVs.

Let's look at an example resume.

YOUR NAME

+1 (123) 456-7890 | Location | email@gmail.com | linkedin | website/blog

Innovative and analytical data scientist with a strong foundation in education and data-driven decision-making. Proficient in Python, SQL, and data visualization, with experience building machine learning models and conducting exploratory data analysis. Leveraged expertise in survey analysis and student performance metrics to drive actionable insights within an educational setting. Completed the Google Analytics Certificate and executed end-to-end data science projects, including predictive modeling and sentiment analysis.

SKILLS

Languages: Python (Pandas, NumPy, Matplotlib, Scikit-learn), SQL
Tools & Frameworks: AWS (S3, SageMaker), Git, Tableau, Excel
Applications: Machine learning, statistics, A/B testing, data visualization

EDUCATION & CERTIFICATIONS

Google Data Analytics Professional Certificate – Coursera, 2024
Master's in Education – University, 2020
Bachelor's in History – University, 2015

PROJECTS

STUDENT PERFORMANCE PREDICTOR (End-to-End Machine Learning Project)

- Objective: Predicted student test performance using anonymized historical data from a simulated school dataset.
- Tools Used: Python, Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
- Steps Taken:
  - Cleaned and preprocessed raw data, handling missing values and standardizing test score metrics.
  - Explored trends with visualizations, identifying key predictors such as study time and attendance.
  - Built and fine-tuned a random forest model to predict student performance with 85% accuracy.
  - Presented findings in an interactive Tableau dashboard for educators.

PARENT SURVEY SENTIMENT ANALYSIS (Exploratory Data Analysis and Natural Language Processing)

- Objective: Analyzed parent survey data to assess satisfaction with remote learning during the COVID-19 pandemic.
- Tools Used: Python, NLTK, Google Sheets
- Steps Taken:
  - Imported survey responses, cleaning and organizing textual feedback for analysis.
  - Conducted sentiment analysis using Python's NLTK to identify trends in positive, neutral, and negative sentiments.
  - Created visual reports (bar charts, word clouds) showcasing key themes like communication effectiveness and technology issues.
  - Delivered insights to improve future remote learning plans.

PROFESSIONAL EXPERIENCE

ELEMENTARY SCHOOL | City, State
Elementary School Teacher - October 2020 - Present

- Designed and implemented engaging lesson plans for history and social studies, fostering student growth.
- Collected and analyzed survey feedback from students and parents to improve teaching effectiveness, increasing satisfaction by 15%.
- Created Excel-based trackers to monitor student performance, identifying trends to tailor instruction.
- Collaborated with colleagues to integrate technology into classrooms, enhancing student engagement.

Here you can see:

- It's one page and formatted in an ATS-friendly way.
- Links are easy to find.
- The summary has lots of keywords, and describes the candidate as already being in the field.
- Skills are prominently listed at the top.
- Since there is a relevant certificate (or degree), Education is near the top as well.
- Then projects follow, with clear examples of tools used and metrics.
- Finally, professional experience is at the end, where we tried as much as possible to make it relevant.

Cover Letters

Now, cover letters: While I don't personally rely on cover letters too heavily at this point in my career, I think in the early days having a well-crafted cover letter that shows enthusiasm for this particular role at this particular company (with details), and transferable skills can go a long way.

For example, assuming we're the teacher with the resume above, and we want to apply for a Junior Data Scientist at a bank:

[Your Name]
[Your Address] (optional)
[City, State ZIP Code] (optional)
[Your Email] | [Your Phone Number]
[Date]

Dear [Hiring Manager's Name or "Hiring Team"],

I was inspired by [Bank Name]'s recent blog post on designing customer surveys to enhance financial product offerings. The focus on leveraging thoughtful survey design to drive actionable insights aligns closely with my experience analyzing parent and student surveys to improve engagement and outcomes in education. I'm excited to apply my skills in data analysis and visualization to help [Bank Name] continue its innovative approach to understanding customer needs.

After several rewarding years in teaching, I am eager to transition to data science to leverage my analytical strengths on a broader scale. I have developed expertise in Python, SQL, and predictive modeling, and have conducted projects such as a sentiment analysis of survey data and building a machine learning model to predict student educational outcomes with 85% accuracy. These experiences honed my ability to analyze complex datasets, derive meaningful insights, and communicate results effectively—skills that translate directly to improving customer segmentation, retention, and satisfaction in the financial sector.

I would welcome the opportunity to bring my analytical expertise and passion for impactful survey analytics to [Bank Name]. Thank you for considering my application, and I look forward to discussing how I can contribute to your team.

Sincerely,
[Your Full Name]

How to Get Interviews

Now that we have our application materials ready and a strong portfolio, **it's time to start applying in earnest**.

We often hear that applications are just a numbers game. And while that is true to an extent, we need to be a lot more strategic than just sending our resume out to every posting we see (and we absolutely don't want to be doing this with GenAI tools).

Instead, prioritize jobs where:

1. You have a strong chance, or
2. You would really love the role.

Spend time customizing your resume – or better yet, developing a networking strategy, which we'll talk about next – for those roles.

If you still have time after really focusing on the high probability or high payoff positions, then sure, go ahead and spam your resume a bit.

What we want to do is avoid the cycle of mindlessly sending out applications, getting rejected, feeling like there's no point putting in the effort, so we're lazy with the next batch of applications, and so on. This is a really poor approach strategically, and is bad for your mental health.

Networking

Even more promising than applications is networking and putting yourself out there one-on-one. In my data science mentoring, the people I'm seeing be successful are the ones with very little social fear. They are super proactive about reaching out to recruiters, hiring managers, and in-person networking.

I know this sounds painful – trust me, I hate doing it too – but the fact is that this is absolutely the highest leverage tool we have in our toolbelt.

If you have little or no industry experience, automated resume screening systems might filter out your application. Reaching out directly to a recruiter can bypass this initial screen and give you a chance to sell yourself as a candidate.

Recruiters receive a lot of messages, so keep yours short, respectful, and to the point.

Start by briefly introducing yourself, mentioning your relevant skills, projects, or certifications. Explain why you're interested in the company specifically, rather than sending a generic message.

For example:

"Hi [Recruiter's Name],

I'm [Your Name], a Data Scientist with skills in Python, SQL, and machine learning. I recently completed a project where I built a recommendation system to improve customer engagement for an e-commerce dataset. I'm particularly interested in [Company Name] because of your focus on [specific product, value, or challenge]. I'd love to connect to learn more about what you look for in data science candidates. Thank you for your time!"

Recruiters and hiring managers are looking for people who can solve real problems. **Instead of just stating you're interested in the role, we can even take it a step further by showing how your skills and projects could add value.**

Study the company's products, recent news, and any challenges they might be facing. LinkedIn, company websites, press releases, or even industry reports can give you ideas. In your message, suggest a few ways your data science or ML skills could address specific needs. For example, if the company focuses on e-commerce, mention how your experience with recommendation systems or customer segmentation projects could enhance their user experience.

Even if you don't have direct industry experience, reference any relevant projects, volunteer work, or skills that demonstrate your ability to solve similar problems. For instance, mention a project where you analyzed customer data or a volunteer role where you optimized a process – anything that shows how you'd contribute effectively.

Whether you're applying to job postings or reaching out to potential contacts, one super important thing is to remember that **you're not going to get a job by focusing on how great the job would be for you, but how you're the best person to solve the company's problems.** Make sure to clarify what differentiates you from other applicants, and how these specific skills are best positioned to solve the specific problems outlined in the job description. **You need to understand the role, and communicate clearly how you plan to bring value to the company – not the other way around.**

For example:

Instead of saying: *"This role is exciting to me because I'm passionate about machine learning."*

Say: *"I see this role focuses on customer segmentation, and my recent project using clustering algorithms demonstrated how to drive actionable insights from user data."*

Be consistent about reaching out to potential connections and being thorough with your applications for high-potential jobs, and at one point you'll start being called for interviews.

Interview Prep

Once you get an interview on the calendar, it's time to start preparing in earnest.

The key thing that I do differently with interview prep is, essentially, over-prepare. I put in a ton of work in advance to anticipate what might be covered, and dig deep into studying those topics so that it looks like I came up with clever, well thought-out solutions on the fly.

This is particularly important if you lack significant industry experience. Preparation will help you bridge that gap and present yourself as someone who's ready to learn and contribute from day one. You cannot afford to go into an interview unprepared.

Interviews can vary widely depending on the company and the role, so focus on the common components:

Recruiter Screen

Be prepared to give a brief introduction, summarize your background, and explain why you're interested in the role.

At this stage, it's important to **make a strong case for why your skills align with the job requirements**. Use keywords from the job description, reference relevant details from their blog or website, and ensure your elevator pitch is well-rehearsed.

Review your past projects thoroughly, understanding every detail, including the reasoning behind your decisions. This preparation ensures you're ready if the recruiter asks you to walk through a specific project. Whenever possible, tailor your examples and explanations to align with the requirements for the job.

Finally, **ask thoughtful questions** to gain clarity about the role and understand the next steps in the interview process. This not only shows your interest but also helps you prepare effectively for future stages.

Coding Questions

By now, you've been practicing coding, so you should be prepared. Make sure to practice thinking out loud and coding at the same time.

Remember, **it's not just about getting the correct answer — interviewers are evaluating how you think, approach problems, ask clarifying questions, and collaborate**. It's worth taking the time to practice coding interviews with another person who can provide feedback on these aspects. This will help you refine your communication skills and improve your ability to explain your thought process effectively.

Behavioral Questions

You can get hints about the kinds of behavioral questions you might be asked by **reviewing the company's values or leadership principles**. These often guide the questions interviewers ask to evaluate cultural fit and alignment with their priorities. Prepare at least one example from your past experience that demonstrates how you embody each of these values.

When answering, **use the STARR method (Situation, Task, Action, Result, Reflection)** to structure your responses effectively. This ensures your answers are clear, concise, and highlight your impact. Focus on examples that showcase teamwork, problem-solving, and adaptability while tying them back to the company's core values.

Case Studies

Case studies test your ability to solve real-world problems using data science and machine learning. **Prepare by researching the business you're interviewing with, understanding their industry, and anticipating relevant problems, metrics, and data types**. Dive into the company's products, services, and goals to tailor your approach and demonstrate domain-specific knowledge.

Business Context

Be ready to discuss **how your work as a Data Scientist can impact the company's bottom line, decision-making processes, or strategic goals**. This shows that you not only have technical expertise but also understand the business implications of your work. Tailor your responses to align with the company's mission and priorities.

Prepare Questions for Each Phase

For every interview stage, **have thoughtful questions prepared**. These could be about the role, the team, the company's challenges, or their expectations. Asking questions not only shows your genuine interest but also helps you gain insights to better position yourself as the ideal candidate.

Preparation is Key, But You Don't Need to Know Everything

The key to success in all of these is to think in advance about the business to such an extent that you're able to essentially guess what the case studies will be, so that you can narrow down the areas to study.

I have a detailed video that goes over how to prepare for every phase of the interview process, from the recruiter call to behavioral questions to case studies. I would definitely recommend checking that out!

All of the advice in that video is the same for beginners, except that **you will have more times when you do not have relevant experience to draw on**. In those cases, don't panic—pivot the conversation to how you would approach the problem. For example: *"While I haven't worked on a problem like this before, here's how I would break it down..."*

This demonstrates critical thinking and a willingness to tackle new challenges, both of which are valuable traits for any candidate.

It's also ok to ask clarifying questions of the interviewer and demonstrate your curiosity and growth mindset. Interviewers don't expect you to know everything, but they do want to see how you think and approach complex problems.

Remember, every interview you attend—even if it doesn't result in an offer—is a valuable learning experience. The more you practice, the more comfortable and confident you'll become.

How to Do Well When You Start Your First Job

Congratulations—you've landed your first data science role! Starting a new job can be intimidating, especially in a technical field like this one.

Here are some suggestions on how to do well when you start your first job:

1. Understand the Business Context

Your first priority should be understanding the company's goals and how your work fits into the bigger picture. This means:

- Learn the metrics that matter: Find out what KPIs (Key Performance Indicators) your team or company cares about and why.
- Understand the challenges: Read up on industry trends, internal documentation, or past projects to get a sense of the business's pain points.
- Talk to stakeholders: Ask stakeholders or team members what problems they're trying to solve and how your work can help. Building this understanding early on will guide your decisions and make your contributions more impactful.

2. Ask Questions and Seek Feedback

It's okay not to know everything—no one expects you to at this point. What matters is that you're proactive about learning. Here's how to approach this:

- Ask questions: Whether it's about the company's processes, tools, or expectations, don't hesitate to seek clarification. This shows initiative and a desire to get things right.
- Get feedback often: Ask for feedback on your work from peers, mentors, or managers. The sooner you know what you're doing well (and what you need to improve), the faster you'll grow.

3. Leverage Your Team

You're not in this alone. Your team is your greatest resource, so don't hesitate to:

- Pair program with colleagues: This can help you learn coding practices, tools, and workflows specific to your company.
- Learn from code reviews: Take reviews seriously and see them as opportunities to improve your skills.
- Find a mentor: Identify someone on your team who is experienced and approachable, and learn from their guidance.

Building strong relationships with your colleagues will not only help you learn faster but also create a supportive environment where you can thrive.

4. Focus on Delivering Value

In your first few months, aim to contribute to projects that provide visible value, even if they're small. For example:

- Automate a repetitive task.
- Create a simple dashboard or visualization to help stakeholders make decisions.
- Optimize an existing process or improve a basic model.

Delivering results, even incremental ones, builds trust and establishes your reputation as someone who gets things done.

5. Stay Curious and Keep Learning

The learning doesn't stop once you're hired (or ever, truth be told). Continue to build your skills by:

- Exploring new tools and techniques: You'll likely encounter tools or methods you haven't used before—take the time to master them, even outside of work.
- Learning from mistakes: If something doesn't work, treat it as a learning opportunity.

Final Thoughts

Becoming a Data Scientist in 2025 might feel like a monumental challenge, but with focus, persistence, and a clear roadmap, it's absolutely within your reach. Remember, this journey is a marathon, not a sprint. It's not about mastering every tool or algorithm under the sun—it's about building a strong foundation, solving real problems, and demonstrating your ability to drive business value.

If you're feeling overwhelmed, don't worry—that's normal. Take it one step at a time, stick to the timeline we've laid out, and keep pushing forward. The most important thing is to stay consistent, stay curious, and keep building—whether it's your skills, your portfolio, or your network.

If you're worried about systems to put all this into practice, I have a video on how I study consistently with a full-time job (learning never ends in this field) – I'd recommend checking that out next so you have the best possible chance of success in the coming year.

Thanks for sticking with me to the end of this roadmap, and good luck! You've got this.

Connect With Me

- 🎥 My YouTube channel: Marina Wyss - Gratitude Driven
- 📫 Sign up for my newsletter! I share weekly ideas and insights across personal growth, professional development, and the world of AI and data science.
- 💬 If you'd like to chat with me 1:1, you can book a call here: https://topmate.io/marina_wyss
- ❤ Or, find me on Instagram: link