ELITEDATASCIENCE.COM'S

DATA SCIENCE CAREER GUIDE

Detailed Answers to FAQs From 29,265 Email Subscribers

Follow me on LinkedIn for more:
Steve Nouri
https://www.linkedin.com/in/stevenouri/

How to Learn DS & ML, Land a High-Paying Job, and Future-Proof Your Career

EliteDataScience.com

COPYRIGHT NOTICE

TABLE OF CONTENTS

– – – * – – –

CH. 1 - LAUNCHING YOUR CAREER

CH. 2 - ROLES AND REQUIREMENTS

CH. 3 - BEST ADVICE FOR ________?

CH. 4 - FUTURE-PROOFING YOUR CAREER

Welcome to EliteDataScience.com's Data Science Career Guide!

When we surveyed 29,265 subscribers on our email list, one of the most common questions was, *"How do I get started in data science and machine learning?"*

We've compiled this guide of FAQs to help you do just that… and much more. We hope that you'll use this guide to jumpstart your journey and cut the learning curve.

Let's start with how to build a rock-solid foundation of practical skills and knowledge. Then, later in this guide, we'll cover specific tips for people of various backgrounds.

To start:

1. **Read the rest of this guide in its entirety.** We surveyed 29,265 subscribers on our email list, and these are the most common questions we've received. Chances are that you have a few of these questions as well.

2. **Circle back to the answer for the question,** *"What's the most efficient way to learn DS / ML as a busy professional?"* In that answer, we outline what we've found to be the most efficient roadmap for learning these skills.

3. **Get your hands wet immediately.** We've prepared several tutorials for you to get started, and we recommend diving into them ASAP. You can find the full list of links and resources later, but here are a few important ones to look out for:

    a. Data Science Primer: The Core Steps of the ML Workflow

    b. Tutorial #1: Python for DS Ultimate Quickstart Guide

    c. Tutorial #2: Intro to Machine Learning with Python and Scikit-Learn

Throughout this guide, we'll also have some external links to additional resources or articles. We recommend reading through the complete guide first, and then checking them out afterwards.

You've made an outstanding career decision to start learning more about DS & ML (even if you decide it's not for you). So without further ado, let's keep going!

CH. 1 - LAUNCHING YOUR CAREER

– – – * – – –

1.1 What do I need to know in order to become a data scientist? / How do I land a job as a data scientist?

While there are a variety of positions that could fall under DS, we've categorized them into two types:

Business Data Scientists and ***Product Data Scientists***.

First, we'll address the core skills that every data scientist needs. Then, we'll address those categories separately. There are also hybrid roles that require the skills from both the business and the product side.

Finally, please note that we're not trying to provide an exhaustive list of everything you might run into. Instead, our goal is to list the core skills within each category that will give you the biggest bang for your buck.

There are only 24 hours in a day... and you still need to sleep, eat, work, go to school, and/or spend time with family and friends. So we're going to introduce the core skills that will ***get you a foot in the door.***

And yes, some employers will have more requirements. But if you lock down the following core skills, you WILL be able to land a high-paying job in this field, guaranteed.

All Data Scientists

1. **Data Analysis / Exploratory Analysis** - First, you need to be able to analyze data and extract key insights. You should do this before any modeling or building any product. That includes data visualization and calculating key summary statistics. Proper exploratory analysis guides you throughout the rest of your project.

2. **Data Preprocessing** - Includes extracting, cleaning, transforming, aggregating, and de-aggregating data. In other words, be comfortable developing raw data into a more useful format for analysis.

3. **Applied Machine Learning** - It doesn't matter if you'll directly be doing the modeling or not... machine learning is one of THE central technologies within this field. Applied ML includes data exploration & cleaning, feature engineering, algorithm selection, and model training.

Business Data Scientist

Business data scientists improve business profitability through data analysis, predictive modeling, and testing. For business data scientists, the emphasis is on the ***insight*** that you can derive from the data.

Examples include:

- Marketing - Building predictive models and bidding strategies for ad markets like Google Adwords or Facebook Ads

- Investing - Using stock price data, global macro-economic indicators, and machine learning to predict stock prices

- Strategy - Using clustering to find "similar" test and control stores for a chain-wide experiment

- Operations - Building models that predict customer churn, allowing the company to proactively reach out

Aspiring business data scientists should add the following core skills to their skillset:

4. **Domain Knowledge** - Data science is never done in a vacuum. You will always be applying your DS skills in a domain (e.g. Marketing or Finance) to drive real business value. You either need to have domain knowledge or the desire to acquire domain knowledge. In fact, it's not uncommon for DS interviews to include case interviews.

5. **Communication and Presentations** - As a business data scientist, arriving at the right data-driven answer is only half the battle. The other half is communicating your insights to key stakeholders to get buy-in. In fact, your job has many similarities with management consulting.

Product Data Scientist

Product data scientists build ML / AI tools and software. They train models, build prototypes, and integrate ML solutions into other parts of the software. For product data scientists, the emphasis is on the ***product*** that you build.

Examples include:

- E-Commerce - Building and integrating a dynamic pricing model into an e-commerce platform
- Entertainment - Building a recommendation engine to recommend other movies a user might enjoy
- Banking - Building a fraud detection system after analyzing large numbers of credit card transactions
- SaaS - Building a chatbot platform that uses natural language processing (NLP) to provide smarter chatbots

Aspiring product data scientists should add the following core skills to their skillset:

4. **Software Development Basics** - You won't need to know as much about software development as a full-stack engineer. But product data scientists usually work closely with software engineers... so you'll need to be able to speak a shared language. Be familiar with concepts like agile development, version control, and software architectures at a high level.

5. **Data Pipelines** - As a product data scientist, managing databases and data pipelines could be a big part of your job. Become familiar with DB languages such as SQL. Also get to know other data formats (e.g. JSON files, web scraping, or unstructured data).

1.2 What are the most relevant tools to learn TODAY in terms of commercial value?

There are many tools with commercial value—too many to list. In fact, you can find high-paying jobs that use almost any modern DS tool... whether it's in Python, R, or a less common language like Julia or MATLAB.

So let's make this question more interesting. Let's consider two more factors aside from employability:

1. Ease of learning - how easy is it for a complete beginner learn?
2. Versatility - do the tools open doors for you a variety of domains?

Considering these two factors, the clear winner is the Python programming language. Python is the most popular language among data scientists, leading to a wider range of opportunities. It's also famously intuitive and easy to learn.

Thus, our recommendations for tools to learn will all fall under the ***Python stack:***

- Python - programming language
- Jupyter Notebook - lightweight IDE (great for analysis and prototyping)
- NumPy - library for numeric computations
- Pandas - library for data management
- Scikit-Learn - library for general-purpose ML
- Keras - library for neural networks and deep learning
- Matplotlib & Seaborn - libraries for data visualization

You can download all those libraries for free using the Anaconda distribution. We are not affiliated with the authors of that distribution, but we use it for all of our work as well.

*Note: Download the latest version for **Python 3.X**. Python 2.X is also viable, and is still used in some places. But all of the major libraries have already been updated to work with Python 3.X, which will become the standard going forward.*

1.3 What's the most efficient way to learn DS / ML as a busy professional?

As a busy professional, you won't have time to dig into all the math and theory right from the start… and you won't need to.

Academia favors this antiquated "bottom-up" approach... but it's not very practical for working professionals seeking a career transition. Not only is it long and tedious, but you'll also be more likely to lose motivation along the way.

The "Top-Down" Approach

Instead, we recommend a "top-down" approach: Your first priority will be to see an entire DS analysis or ML project from start to finish… warts and all.

You'll start with *tutorials* instead of *lectures.* A tutorial teaches you how to do something in as streamlined of a way as possible. As you'll notice, you won't understand how everything is working under the hood… *yet*.

However, if you follow the tutorial step-by-step, you should be able to see an entire DS task from start to finish. This is invaluable for your learning journey! Because when you start to see the big picture, you'll understand how all the moving pieces fit together.

Solidifying Your Skills

After you complete a tutorial, it's time to apply what you learned to new datasets. This will allow you to solidify your skills and begin expanding your knowledge.

For example, when you try the same modeling process on a new dataset, you might run into a new error. Upon googling the error, you might discover that it's because the dataset had a different format... or missing values... or mislabeled classes... and so on. Now you can dig into that topic further and expand your knowledge... ***within the context of what you've already learned.***

This technique of "learning in context" is one of the most powerful learning tools that we've seen. It's especially useful for busy professionals on a tight schedule.

Roadmap of Topics

Note: We'll cover some of these in more detail throughout the rest of this guide.

1. **Understand the DS & ML workflow at a high level**
   a. Read the Data Science Primer
   b. Read the guide to Modern Machine Learning Algorithms
2. **Learn Python programming basics**
   a. Complete the Python for Data Science Quickstart Guide
   b. Bookmark this Python for DS Cheat Sheet
3. **Learn the basics of the Pandas library**
   a. Complete the Python Data Wrangling Tutorial with Pandas
   b. Bookmark its official documentation page (you'll reference it often)
4. **See the modeling process from start to finish**
   a. Complete the Python Machine Learning Tutorial with Scikit-Learn
   b. Complete the Kaggle Titanic Dataset Training Competition
5. **Download more datasets you find interesting**
   a. Download from a hand-picked list here.
   b. Project ideas: Fun Machine Learning Projects for Beginners
6. **Practice the other core skills of applied ML using those datasets**
   a. Data visualization and exploratory analysis (Tutorial)
   b. Data cleaning (Examples)
   c. Feature engineering (Examples, More Examples)
7. **Build a portfolio of real-world projects. Then apply!**
   a. See the question, *"How do I build a portfolio of real-world projects?"*

1.4 How do I switch careers as quickly as possible?

Many people have the misconception that you need to ***learn, learn, learn... and learn more*** to land a job as a data scientist. That’s fine, but it’s not the most efficient way of switching careers as quickly as possible. Time is money, and every extra day you spend on extraneous tasks will directly cost you lost income.

Instead, we recommend you to ***learn enough... then show, show, show.***

What does this mean?

What does this mean? Well… first, it means that you shouldn’t try to learn everything about DS & ML. Instead, you should pick the closest goal posts and execute against that target.

Target the core skills that we discussed in the previous question, *“what do I need to know in order to become a data scientist?”*

The first three—**data analysis**, **data preprocessing**, and **applied machine learning**—are especially important for all data scientist roles.

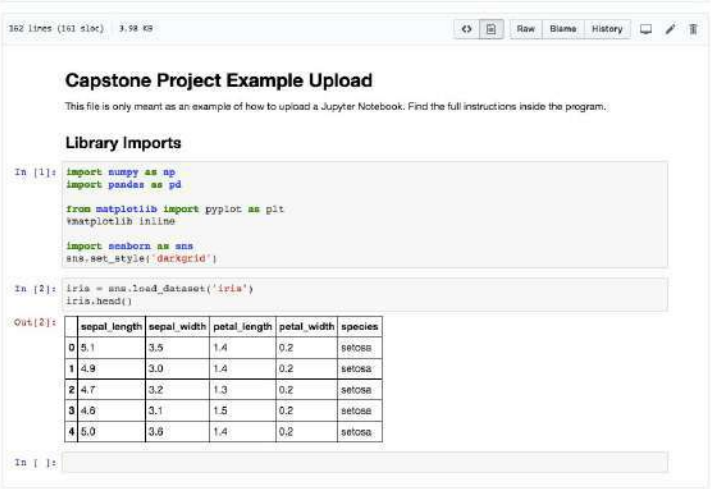
Once you’ve learned the basics of those skills, don't expand the scope of your studying (a common mistake we see). Instead, focus on ***showing*** those skills to employers! Build a portfolio of real-world projects that you can point to and *prove* your competency.

1.5 How do I build a portfolio of real-world projects?

1. **Learn the core skills for data science and applied machine learning.** See *question 1.3* for our recommended roadmap.

2. **Pick out a dataset to start with.** Choose a dataset that is in a domain you might wish to enter and allows you to show your skills (i.e. no toy problems). We've hand-picked some great datasets for you on this resource page.

3. **Start your project in Jupyter Notebook.** Jupyter Notebook is a lightweight IDE for Python DS. It's available for free as part of the Anaconda distribution. Jupyter notebooks can run code, display outputs, and keep notes all in one place. Plus, after you complete your project, can host it online seamlessly (more on this later).



Capstone Project Example Upload

This file is only meant as an example of how to upload a Jupyter Notebook. Find the full instructions inside the program.

Library Imports

```
In [1]: import numpy as np
        import pandas as pd

        from matplotlib import pyplot as plt
        %matplotlib inline

        import seaborn as sns
        sns.set_style('darkgrid')
```

```
In [2]: iris = sns.load_dataset('iris')
        iris.head()
```

Out[2]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
In [ ]:
```

4. **Explore the data and make sure you understand the features.** The first step is to explore the data and make sure you understand it from an intuitive perspective. Only then can you pose interesting questions to answer.

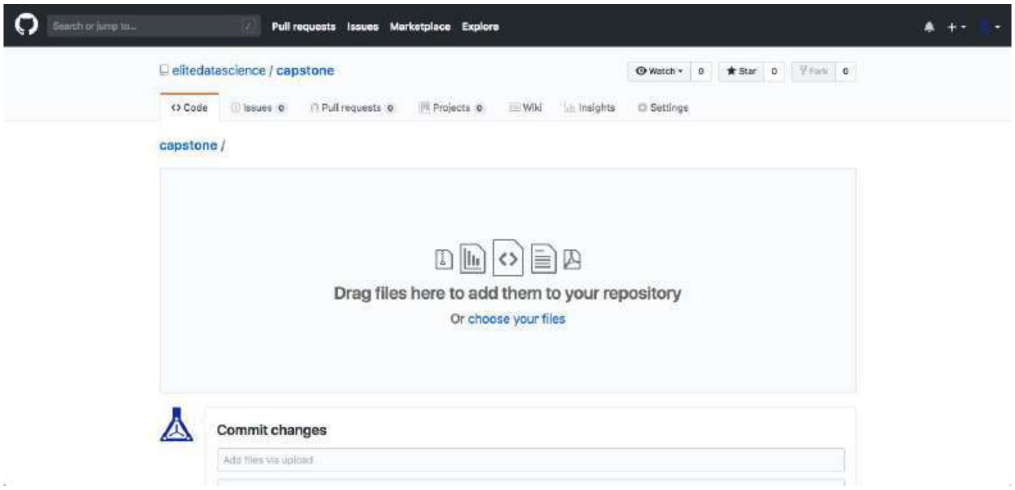
5. **Define an interesting objective to pursue.** You'll have a much better sense of how to do so after you've learned the core skills (step 1). For example: *Can I train a model that predicts X? Can I classify Y based on the features available in the dataset? Do natural clusters appear in the observations?*

6. **Clean the data, engineer features, and build your analytical base table (ABT).** The next step is to create an "analytical base table" from the original dataset. Pre-processing the data allows you to answer more interesting objectives.

7. **Complete your analysis / train your models.** Once you've created your analytical base table, you'll have already done most of the heavy lifting. All that's left is to finish the analysis/modeling part of your project.

8. **Write about your project directly inside your Jupyter notebook.** Write a detailed intro. Then, explain your data, describe your objective, and summarize your results / key-takeaways. You can also write about how you'd expand upon your project further.

9. **Upload your project to Github.** Github is a free file and versioning management system. You can upload your completed project (in Jupyter notebook) into your own "repository" (a.k.a. folder) on Github and host it for free. Then, you'll get a link that you can share. Potential employers will be able to view your project directly from their browsers!



Drag files here to add them to your repository

Or choose your files

10. **Repeat steps (2) to (9) for a handful of other datasets and problems.** Et voilà, your portfolio is ready to go! Finally, link to your portfolio from your resume, LinkedIn, and job board accounts.

CH. 2 - ROLES AND REQUIREMENTS

– – – ⋆ – – –

2.1 What is the difference between Data Science, Machine Learning, AI, Data Analysis, and Deep Learning?

Rather than try to define each of these terms from scratch, we'll start with the basic Wikipedia entries. Then, we'll provide our own commentary on top.

In summary: Data science encompasses data analysis and machine learning. Deep learning is a family of ML methods that deal specifically with neural networks. Artificial intelligence is the broader study of mimicking human cognitive functions using computers. Machine learning offers one path of research toward AI.

Data Analysis

> *Data analysis is a process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, informing conclusions, and supporting decision-making.* - Data Analysis, Wikipedia

This one is fairly self-explanatory. Data analysis has existed in some form or another since the ancient world. Ancient Roman armies would send *Speculatores* and *Exploratores* ahead to scout and track enemy movements (i.e. collect "data"). Then, military advisors would "analyze" that data and help the commanders make more informed decisions.

Today, it's the same idea—modernized. Software collects the data. Analysts extract insights from it. And business leaders get to make more informed decisions.

Data science introduces machine learning methods on top of traditional data analysis. ML helps process massive amounts of data, build more accurate models, and mimic human cognitive functions.

Data Science

> *Data science is a multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data.* - Data Science, Wikipedia

In practice, data science leverages data analysis, applied machine learning, and domain knowledge. It's commercially-oriented. So it's essential to develop your domain expertise as a data scientist, and not only the technical skills.

Machine Learning

> *Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead.* - Machine Learning, Wikipedia

The key word there is "explicit." For true machine learning, a computer must be able to recognize patterns that it's not explicitly programmed for identify. Machine learning algorithms process data and build models from the patterns they observe.

For more information, see the section titled "What makes machine learning so special?" in our *Bird's Eye View of Applied Machine Learning*.

Artificial Intelligence

> *Colloquially, the term "artificial intelligence" is used to describe machines/computers that mimic "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".* - Artificial Intelligence, Wikipedia

Think of AI as a destination and machine learning as one path to get there. AI research and development aims to mimic human cognitive functions, including decision making. Machine learning is the most promising attempt toward AI to date.

Imagine you wanted to program a self-driving car and train the computer to know what to do at a traffic light. Well, you *could* explicitly instruct the computer to always stop at a red light, slow down at yellow light, and go through at a green light. In fact, this is how the AI of most computer games work—with a set of specific instructions for various game states.

And yes, that would certainly be an attempt toward AI... but it wouldn't be very effective in the messy real world. There are so many "states" you might not have accounted for.

For example, what if someone is still crossing the road when the light turns green? What if the light becomes broken? What if it's flashing yellow? What if the light is not a traffic light but rather something else, such as police lights?

Machine learning, on the other hand, does not rely on explicit instructions for each state. Instead, you'll feed the computer as much relevant data as you can gather. Then, one of many possible "algorithms" will build a "model" from that data. That "model" will then be able to take a new input (captured by the camera) and provide an output (instructions for the car) with a certain level of confidence.

Deep Learning

> *Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on artificial neural networks.* - Deep Learning, Wikipedia

Deep learning refers to a family of ML methods that deal with neural networks. Neural networks usually need much more data to train than other ML methods. Deep learning offers exceptional performance in some, but not all domains. It shines in domains like computer vision, natural language processing, and audio processing.

Despite the allure of neural networks, they're not as widely applicable or beginner-friendly as other ML methods. Aspiring data scientists should start by learning more "general-purpose" ML methods such as Logistic Regression, Random Forests, and Boosted Trees.

2.2 How much math should I learn for DS / ML?

The short answer is: probably much less than you think.

The long answer is that it depends on your goal.

If your goal is simply to land a high-paying job in data science, then you can do so with very little math foundation... IF you learn how to apply the right tools, at the right places, in the right way.

If you can prove your skills, then at least one great company out there will give you a chance. Currently, the demand for DS skills vastly outpaces the supply... so companies will NOT turn you away if you can prove your abilities.

Follow the top-down approach we outlined in *"What's the most efficient way to learn DS / ML as a busy professional?"* Then, focus on building a portfolio of real-world projects.

After you master the DS / ML workflow, you can then dive into the theory to supplement your practical skills.

If you wish to perform original research in ML and work on things like self-driving cars, then you'll need more math. Yet even so, our recommendation would still be to pick the nearest goalpost and *start* with that. Follow the top-down approach to get a foot in the door *first*. This will give you a professional environment to dive further into the math and theory.

2.3 Do you need an advanced degree / CS degree / math degree to become a successful data scientist?

No. Perhaps 10-15 years ago, but not today and definitely not in the future. The technical and mathematical requirements for DS have largely been overblown.

The biggest opportunities with DS & ML in the future will NOT lie in their ***implementation***, but rather in their ***application***.

Today, data scientists will almost NEVER code an algorithm from scratch or derive any sort of math formula. Instead, pre-existing implementations (like Python's Scikit-Learn library) have become the industry standard.

The technical skills will not be difficult to learn. Instead, the value that you can add as a data scientist will come from your creativity and domain expertise.

Pre-existing libraries such as Python's Scikit-Learn offer you:

- Optimized implementations of the most popular and useful algorithms
- Easy-to-learn APIs (i.e. interfaces) for interacting with them
- Large communities of users that will help you overcome roadblocks
- Constant updates to stay on par with the latest technologies and best practices
- A variety of data pre-processing modules

Pre-existing libraries allow you to focus on business growth instead of code implementation. That also makes them the *smart business decision*.

In the business world, companies care about ***results***. A data scientist who leverages existing tools will outperform one who tries to do everything from scratch.

2.4 What makes a good data scientist?

We’ve already covered many of the specific skills throughout this FAQ. So let’s go over a few important mindset differences between bad and good data scientists.

Bad Data Scientists	***Good Data Scientists***
Fixate on technical details	Always drive toward business value
Only obsess over the math	Develop elite communication skills
Are married to specific methods	Know when & when NOT to use ML
Lose track of time while optimizing	Understand tradeoffs / deadlines
Only improve their technical skills	Seek to acquire domain knowledge
Use data to prove their biases	Use data to correct their biases
Over-emphasize algorithms	Focus on feature engineering
Consider themselves masters	Consider themselves lifelong students

2.5 Am I too old / too young to become a data scientist?

Anyone can become a data scientist, regardless of age, educational background, or prior work experience. That's not an exaggeration.

We've seen...

- people with zero relevant experience
- retirees re-entering the workforce
- college dropouts
- super busy working professionals
- and people from many other walks of life

...land high-paying data science jobs.

They do so by:

(1) developing real skills and then...

(2) building a portfolio of projects that help them prove their real skills beyond a shadow of a doubt.

We've covered how to do both of these steps earlier in this chapter.

Fact: DS skills are in heavy demand right now, with not enough supply. Fact: as long as you can prove that you have these skills, *someone* will give you a shot.

CH. 3 - BEST ADVICE FOR ________?

– – – * – – –

3.1 Best advice for people with business backgrounds seeking to enter the field?

People with business backgrounds tend to overestimate the difficulty of learning technical skills. On the flipside, they tend to underestimate their own unique advantages. Here's what you can do:

1.) Pick the nearest goal post and get a foot in the door first.

A lot of people try to jump straight into the deep end. This neither necessary nor recommended for aspiring data scientists seeking entry-level positions. For example, if you don't have a technical background, don't start by aiming to research neural nets at Google.

Even if that's where you'd like to end up, it's not the best target to start with. Begin with the core skills of ***data analysis*** and ***applied machine learning***. You'll get more mileage from these fundamental skills. They'll give you "marketability" to get hired. Then, you can always learn the rest along the way.

2.) Start with a top-down approach, and don't get lost in the weeds.

First of all, Know that you can develop the technical skills fairly quickly by using a "top-down" approach to skip the unnecessary parts of the theory, instead of a "bottom-up" approach... and when you do, there will be HEAVY demand for someone of your profile. For more info, see our answer to the question, *"What's the most efficient way to learn DS / ML as a busy professional?"*

3.) Emphasize your domain expertise.

Remember that data science is never done in a vacuum, and technical skills are only one piece of the puzzle. The bottom line is that employers want to know if you can use DS to help them make more money.

So emphasize your strengths. Show employers that you can spot opportunities. Show them that you can connect DS/ML with tangible business value. You can do so in two ways.

First, you should tailor your portfolio projects to highlight your domain expertise. More on this in the next tip.

Second, during your interviews, you should always shift the conversation to business value. Arrive prepared with ideas of how DS/ML can help the employer's business.

The first step is to learn the core skills of applied ML, which we've covered earlier. After you do so, you'll understand the capabilities and limitations of ML as a technology. Combine this understanding with your previous experience... and BOOM... you're now a candidate that employers will drool over.

4.) Build a portfolio of real-world projects that showcase your domain expertise.

Again, the first step is to learn the basics using tutorials. Then, hone your skills on real-world datasets with commercial use cases. You'll accumulate a portfolio of real-world projects that you can use to get a foot in the door. This is especially important for people coming from business backgrounds. It will prove your technical competency and show your willingness to learn.

5.) Don't limit your search to positions with "data scientist" in the job title.

This is especially true if your current position does not ask you to handle data or do any form of analysis. Seek adjacent positions that will eventually allow you to transition into data scientist.

Great examples would be Data Analyst, Marketing Analyst, or Business Intelligence roles. Each of these positions will expose you to some of the skills needed for DS, allowing you can make up the rest on your own.

3.2 Best advice for students seeking to enter the field?

The main hurdle students need to overcome is the lack of business experience. Many employers will see you as a risky hire. Here's how to overcome this obstacle:

1.) Focus on developing real skills that can drive business value.

Most employers will not care about your DS 101’s “final project” that has you classifying kittens and dogs. Instead, seek real-world datasets with commercial use cases. Hone your skills on those. These datasets are messier, more ambiguous, and contain red herrings to filter out.

2.) Build a portfolio of real-world projects, not toy problems from school.

This is an extension of tip #1. As you tackle those real-world datasets, you can build a portfolio of projects at the same time. You can do so by including write-ups with detailed introductions and descriptions.

Complete them in Jupyter Notebooks and host the final notebook online. There are a variety of free ways to do so (such as Github or Google Drive). You can then link to your portfolio on your resume, LinkedIn, and job board profiles.

This is one of the best ways to stand out from the sea of applicants who can only make empty claims.

3.) Seek internships while still in school.

The best way to land an internship is the same as landing a job. Prove that you have real skills that can help a company make more money. Learn the skills, build a portfolio, and apply to as many relevant positions as you can manage (it’s a numbers game).

After you apply, prepare for the interview process. Review key concepts and practice explaining projects in a clear and concise way.

4.) Don’t limit your search to positions with “data scientist” in the job title.

Also seek adjacent positions that will eventually allow you to transition into data scientist. Great examples would be Data Analyst, Software Developer, Marketing Analyst, Business Consultant, etc.

Each of these positions will give you invaluable work experience. At the same time, they'll expose you to a part of the skills necessary for DS, allowing you can make up the rest on your own.

5.) Don't be discouraged—just apply.

Many positions will claim they need X years of work experience. Think of that as a "target" instead of a hard "cutoff."

At the amusement park, for some rides you "must be this tall to ride." But the job market is different. At many places, the work experience "requirement" is more of a preference. It's "we prefer you to be this tall to ride." In other words, don't be discouraged.

As a student, time is on your side. You have more control over your time, so use that to an advantage. Go all out in the numbers game. Full court press. Just apply to as many relevant positions as possible... and let the opportunities filter themselves.

3.3 Best advice for people with software engineering backgrounds seeking to enter the field?

Software engineers already have strong technical skills. So focus on developing your analytical skills and domain knowledge. These will help you stand out from other candidates with strong technical skills.

1.) Data science is not only machine learning; analytical skills are crucial.

Software engineers often gravitate toward the machine learning side of data science. It’s closer to their comfort zone. But to become a well-rounded data scientist, analysis and domain expertise are vital.

In your preparation, be sure to practice analysis:

Find a good dataset and read its description. Then, brainstorm a list of compelling questions that the dataset might answer.

For example, let's say you find a dataset on school dropout rates. You might ask questions such as:

- *Which types of students are at highest risk of dropping out?*
- *What is the average grade in which students drop out?*
- *Are there any school programs correlated with lower dropout rates?*

Once you have a list of your questions, practice answering them! Try displaying key statistics from the dataset... or plotting visualizations... or taking slices of the data... or taking sums, averages, and so on.

Even if you discover that you can't answer the question, simply *trying* to will sharpen your analytical skills.

2.) Skip most of the math for now.

We’ve seen many software engineers who want to transition into the field get bogged down by the math. In reality, you probably need to know much less than you think you do.

Go with the “top-down” approach we outlined earlier. Don’t feel pressured to lock down all the math right from the start, as you can learn it as you go.

3.) Domain knowledge can help you stand out big time.

Many software engineers are already very strong in their technical skills... so one of the best ways to stand out is to show your willingness to learn about the domain.

For example...

- For adTech, learn about ad auctions and marketing metrics.
- For finance / trading, learn about economics and data sources.
- For marketing, learn about the major social media platforms.
- For SaaS, read books like *Behind the Cloud* (the story of Salesforce).

You get the point. You can only connect DS with business value if you understand the business you're in.

4.) Be prepared for the mindset difference between software development and data science.

In general, software engineering is about making a plan and then executing on it. You’ll map out the architecture, spec out the features you’ll need, and then come up with a to-do list to execute against.

DS is very different in that it’s often a process of *exploration* and *discovery*.

Yes, you’ll navigate with a framework (e.g. clean data → engineer features → choose algorithms → train models). But you’ll often need to change your plan on the fly as you uncover more insights from the data.

5.) Practice your communication skills.

While some software engineers are great communicators, it’s usually not a big part of the job. So it’s crucial that you practice explaining complex topics in clear and concise ways. Our recommendation: grab a friend who knows nothing about what you do... and then try to explain your job to them in plain English. (It works!)

3.4 Best advice for someone with no relevant work experience seeking to enter this field?

Much of the advice we gave earlier still applies here.

Think of it this way: your lack of relevant work experience means that employers will see you as a risky hire. So how can you mitigate that risk for them?

Well, **step one is to develop the real skills capable of driving business value.** We're not trying to "hoodwink" anyone here. You can cut the learning curve by following the top-down approach we outlined earlier.

Then, once you've gotten the basics down, **step two is to prove that you have those skills.** You don't have the relevant work experience to back you up... so what do you do? You build a portfolio of real-world projects.

We've repeated this point several times by now, but it's really that simple. It's all about risk-mitigation for the employer. There's no better tool for doing so than having something tangible that you've built and can show.

3.5 Best advice for someone seeking to transition from data analyst to data scientist?

For data analysts, you'll need to show proficiency in two main areas: programming basics and applied ML. Here's how:

1.) Tighten up your programming skills.

How much? It depends on the tools you already work with. If you use R or Stata on a daily basis, then you'll have a nice head start.

If you mostly use Excel, then you’ll want to add basic programming skills to your repertoire.

You don’t need to get too fancy with it. Pick up the basics of a language like Python (our choice) or R. Then, try to recreate some of the analyses you’re already doing.

For example, you can try to replicate a pivot table using the Pandas library’s groupby function. (You'll discover that it’s often much faster and easier with Pandas when you get the hang of it.)

2.) Develop concrete knowledge in applied machine learning.

Remember, the key difference between data analysis and data science is the addition of machine learning.

Data scientists will need to understand machine learning, regardless of the role. Learning machine learning will also give you some great programming practice. We recommend the Scikit-Learn library.

3.) Propose machine learning projects or “pilots” at your workplace.

One of the best ways to transition into a data scientist is to start working more like a data scientist. You can do so by proposing machine learning projects at your workplace.

We’ve found that upper management are very receptive to the idea when you frame it as an “experiment” or a “pilot” to expand your team’s capabilities.

4.) Expand the range of datasets you’re familiar working with.

You might already be work with data during your day job, it might be limited by what your company has access to.

As you improve your applied ML skills, you should also expand the range of datasets you can work with. We’ve handpicked some for you here: Datasets for Data Science and Machine Learning.

5.) Build a portfolio of real-world (side) projects.

Ok, we’ve mentioned this several times already, so we’re not going to beat a dead horse. Just know that a portfolio of projects is one of the best things you could create to help get your foot in the door.

As you expand your comfort zone with new datasets (tip #4), think about how you can create full-length projects out of them. Then, host them online on a site like Github. Complete your project inside Jupyter Notebook. It integrates nicely with Github and also allows you to export your notebook as a web-page.

CH. 4 - FUTURE-PROOFING YOUR CAREER

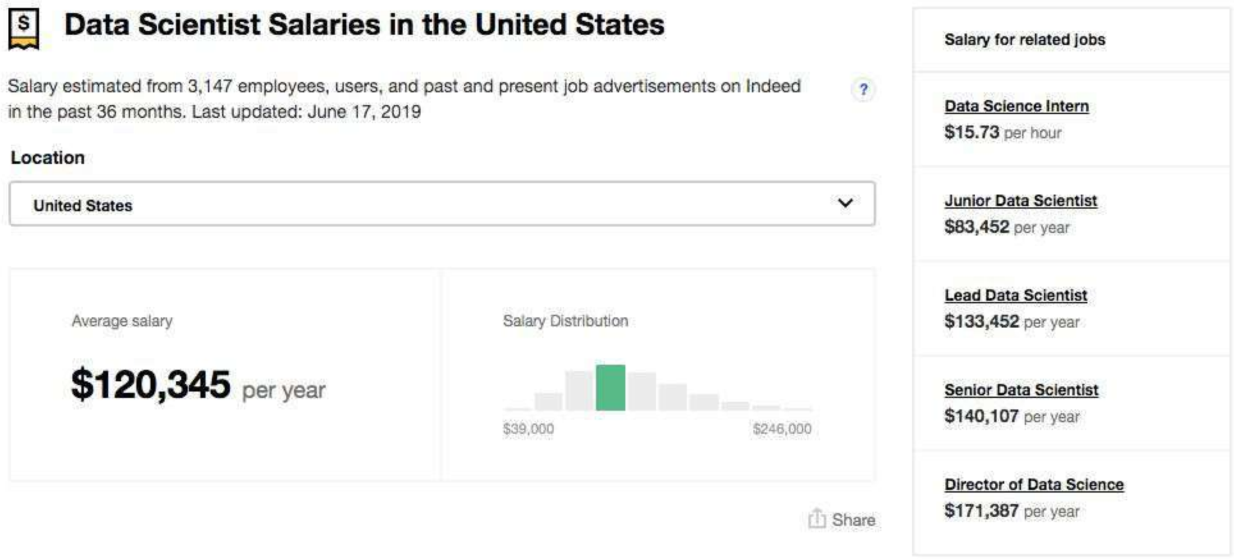
– – – * – – –

4.1 What does the career path of a data scientist look like?

In general, there are two types of data science career paths, each with its own appeal.

Path #1 - Leveling Up

The first is the ***level up path.*** Data science is a skill that you can continuously level up, and your career will grow alongside. For example, here's a sample path from *Data Science Intern* to *Director of Data Science*:



Data Scientist Salaries in the United States

Salary estimated from 3,147 employees, users, and past and present job advertisements on Indeed in the past 36 months. Last updated: June 17, 2019

Location: United States

Average salary

$120,345 per year

Salary Distribution

Salary for related jobs

Data Science Intern
$15.73 per hour

Junior Data Scientist
$83,452 per year

Lead Data Scientist
$133,452 per year

Senior Data Scientist
$140,107 per year

Director of Data Science
$171,387 per year

Source: Indeed.com

As you can see, this is the more "straightforward" career path within data science. As you get better, you'll earn higher salaries, lead bigger projects, and get more senior titles. Large players in the economy—from tech giants to Fortune

500's—are all hiring data scientists. When you're ready for the next level, there *will* be an opportunity awaiting you.

Path #2 - Choose Your Own Adventure

The second path is the ***choose your own adventure*** approach. The modern economy is data-driven. Data science is not a buzzword—the ability to extract actionable insights from data really does help companies make more profit.

So you don't even need to continue "leveling up" as a data scientist if you don't want to.

For example, you could instead transition into…

- **Marketing** (many CMO's now come from data backgrounds)...
- **Finance & investing** (banks and hedge funds are big employers in this space)...
- **Product management** (dozens of ML startups get funded every week)...
- Or even **freelancing** to earn a great income from the comfort of your home (data scientists command some of the highest hourly rates on sites like Upwork.com)...

The point is that when you develop the ***skills***, you're not tied to the *position* unless you prefer to be. Your data background will be in very high demand, opening the door to opportunities that you otherwise would not have access to.

Therefore, we always recommend learning DS & ML through hands-on practice with real-world datasets. We'll get into the specifics of how to do so later on in this guide.

4.2 Should I use libraries and pre-existing solutions, or should I code algorithms from scratch?

For 9 out of 10 data scientists, we'd recommend focusing on pre-existing solutions. For more context, see the question, *“Will DS/ML be automated in the future? How can I future-proof my skills and my career?”*

For learning purposes, you can choose to code a few of your favorite algorithms from scratch. But we wouldn’t recommend sinking too much time optimizing your code or worrying about the nitty gritty.

Coding from Scratch

Advantages	***Disadvantages***
Can learn how the algo works under the hood	Higher math & programming requirements
Customizable implementations	Takes a long time
Potentially faster implementations (but unlikely)	Difficult to beat pre-existing libraries

Libraries and Pre-existing Solutions

Advantages	***Disadvantages***
Easier to learn	Cannot customize implementations
Much more commercial demand	Cannot see each step of the algo
Pre-optimized implementations allow you to focus on the application and building better models	Limited in functionality by what’s already there in the pre-existing library

4.3 How can I stay abreast with the latest tools and best practices given the rapid pace of this industry?

We have some pretty unconventional advice for this one. The usual advice is to follow industry publications, blogs, and conferences. Of course, that advice WILL work. It's applicable to almost every field, including medicine, engineering, sales, and so on.

But as you've probably gathered by now, our approach is to try to get as much hands-on experience as possible. So we recommend the following:

1. **Land a job as a data scientist or in an adjacent role (e.g. data analyst).** The best way to stay up-to-date is to get paid for doing so. A news anchor's job helps them stay informed of current events... and your job as a data scientist will help you keep up with the latest best practices. We've already covered our recommended path to becoming a data scientist earlier, in Chapter 3.

2. **Compete in competitions, such as those on Kaggle.** The point is not to win or lose, but rather to be able to see solutions from others and expand your network by participating in the forums. Each competition winner will provide a write-up about their methodology, which Kaggle publishes.

3. **Work on at least one side project every month / quarter.** Continue working on side projects. Projects, especially those outside of work, allow you to expand your skillet. Plus, they keep you sharp and informed about the latest tools. Make sure you use these projects to give yourself exposure to new types of datasets as well.

4.4 Will DS/ML be automated in the future? How can I future-proof my skills and my career?

The best way to future-proof your career is to focus on the ***application*** instead of the ***implementation***.

Implementation requires you to pursue the “best technology.” But here's the problem. There are already very potent pre-existing libraries (e.g. Scikit-Learn). Cloud-based solutions (e.g. AWS ML), are also being actively developed.

In the future, ML and DS might even be more automated, with platforms that handle much of the DS/ML workflow. In other words, if you focus on implementation, you’ll be in race that you simply can't win.

But you know what can't be easily automated? The *application* of these technologies to drive real-world business value.

Application requires unique skills beyond the technical ones:

1. ***Opportunity Assessment*** - the ability to identify real-world use-cases for DS/ML.

2. ***Creativity*** - the ability to connect the dots between problems and solutions.

3. ***Domain expertise*** - the ability to anticipate what’s important and relevant in your domain.

4. ***Nuanced decision making*** - the ability to balance real-world trade-offs when choosing a solution.

5. ***Empathy*** - the ability to understand how your solutions will affect **real people**... and how to create win-win scenarios (i.e. expanding the pie instead of stealing the pie).

To practice, leverage existing implementations (e.g. Scikit-Learn and Keras) as much as possible. Practice on a variety of real-world datasets.

4.5 How can I use DS or ML to make money from home? / Are there remote opportunities?

We've received this question many times, especially from subscribers in developing countries. You don't need to have many local DS employers to succeed in DS. DS is a big part of the globalized economy. You have many virtual/remote opportunities available.

According to RemoteOK.io, the median salary for remote data scientists at the time of this writing is $88,750 USD. That is very healthy income in any part of the world, and it can be a life-changing salary in some.

We have hired from some of the following platforms, and we've heard great things about the others.

Freelancing

- UpWork
- Toptal

Remote Roles in Data Science

- AngelList
- FlexJobs
- RemoteOK.io
- ZipRecruiter
- SimplyHired

– – – ⋆ – – –

And that wraps up the EDS Career Guide! Visit EliteDataScience.com for more...

- ***Guides***
- ***Concept Explainers***
- ***Code Tutorials***
- ***Career Guidance***
- ***Tools & Resources***