Data Science Learning Roadmap for 2021

Harshit Tyagi

Although nothing really changes but the date, a new year fills everyone with the hope of starting things afresh. If you add in a bit of planning, some well-envisioned goals, and a learning roadmap, you'll have a great recipe for a year full of growth.

This post intends to strengthen your plan by providing you with a **learning framework, resources, and project ideas** to help you build a solid portfolio of work showcasing expertise in data science.

Just a note: I've prepared this roadmap based on my personal experience in data science. This is not the be-all and end-all learning plan. You can adapt this roadmap to better suit any specific domain or field of study that interests you. Also, this was created with Python in mind as I personally prefer it.

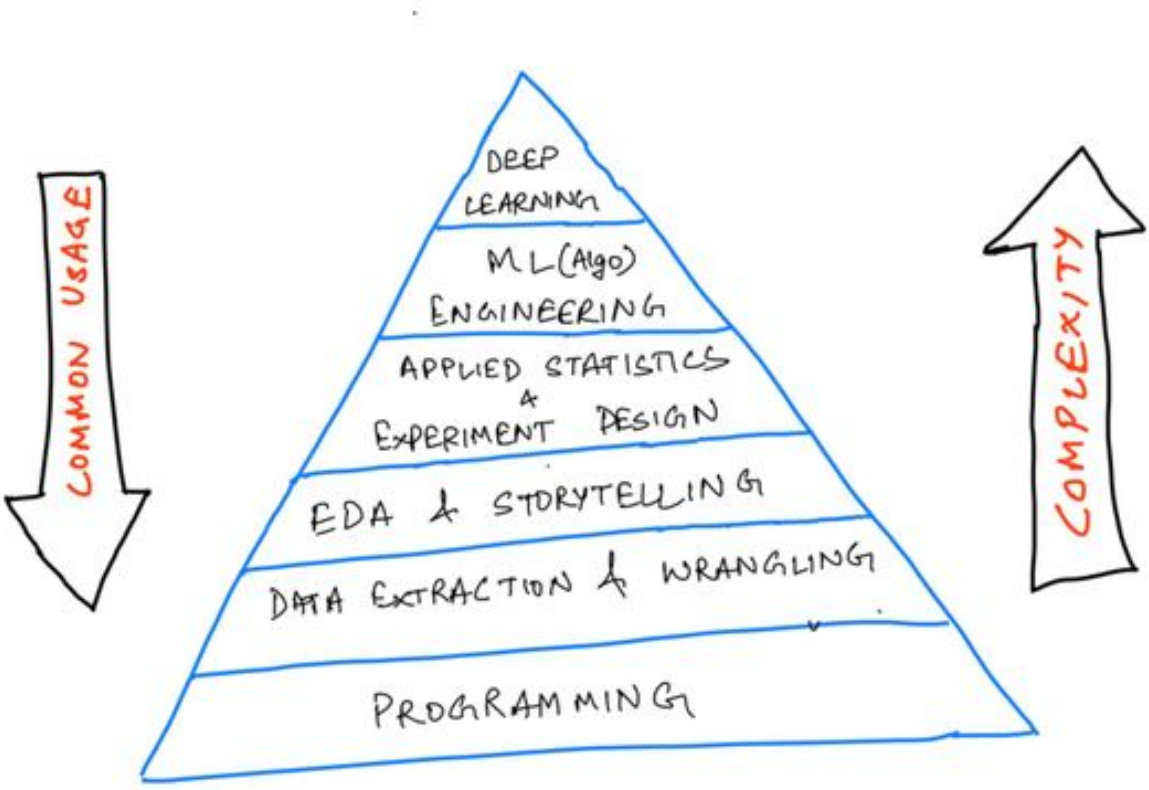
What is a learning roadmap?

A learning roadmap is an extension of a curriculum. It charts out a multi-level skills map with details about **what** skills you want to hone, **how** you will measure the outcome at

each level, and **techniques** to further master each skill.

My roadmap assigns weights to each level based on the complexity and commonality of its application in the real-world. I have also added an estimated time for a beginner to complete each level with exercises and projects.

Here is a pyramid that depicts the high-level skills in order of their complexity and application in the industry.



Data science tasks in the order of complexity

This will mark the base of our framework. We'll now have to deep dive into each of these

strata to complete our framework with more specific, measurable details.

Specificity comes from examining the critical topics in each layer and the resources needed to master those topics.

We'd be able to measure the knowledge gained by applying the learned topics to a number of real-world projects. I've added a few project ideas, portals, and platforms that you can use to measure your proficiency.

Important NOTE: Take it one day at a time, one video/blog/chapter a day. It is a wide spectrum to cover. Don't overwhelm yourself!

Let's deep dive into each of these strata, starting from the bottom.

1. How to Learn About Programming or Software Engineering

(Estimated time: 2-3 months)

First, make sure you have sound programming skills. Every data science job description will ask for programming expertise in at least one languages.

Specific programming topics to know

include:

Common data structures (data types, lists, dictionaries, sets, tuples), writing functions, logic, control flow, searching and sorting algorithms, object-oriented programming, and working with external libraries.

SQL scripting: Querying databases using joins, aggregations, and subqueries

Comfort using the Terminal, version control in Git, and using GitHub

Resources to learn Python:

learnpython.org [free]— a free resource for beginners. It covers all the basic programming topics from scratch. You get an interactive shell to practice those topics side-by-side.

Kaggle [free]— a free and interactive guide to learning python. It is a short tutorial covering all the important topics for data science.

Python certifications on freeCodeCamp [free] – freeCodeCamp offers several certifications based on Python, such as scientific computing, data analysis, and machine learning.

Python Course by freecodecamp on YouTube [free]—This is a 5-hour course that you can follow to practice the basic concepts.

Intermediate python [free]— Another free course by Patrick featured on freecodecamp.org.

Coursera Python for Everybody Specialization [fee]—this is a specialization encompassing beginner-level concepts, python data structures, data collection from the web, and using databases with python.

Resources for learning Git and GitHub

Guide for Git and GitHub [free]: complete these tutorials and labs to develop a firm grip over version control. It will help you further in contributing to open-source projects.

Here's a Git and GitHub crash course on the freeCodeCamp YouTube channel

Resources for learning SQL

Here's a course on SQL and Databases on the freeCodeCamp YouTube channel

Intro to SQL and Advanced SQL on Kaggle.

Treehouse offers a good introductory SQL course here.

Measure your expertise by solving a lot of problems and building at least 2 projects:

Solve a lot of problems here: HackerRank (beginner-friendly) and LeetCode (solve easy or medium-level questions)

Data Extraction from a website/API endpoints—try to write Python scripts from extracting data from webpages that allow scraping like *soundcloud.com.* Store the extracted data into a CSV file or a SQL database.

Games like rock-paper-scissor, spin a yarn, hangman, dice rolling simulator, tic-tac-toe, and so on.

Simple web apps like a YouTube video downloader, website blocker, music player, plagiarism checker, and so on.

Deploy these projects on GitHub pages or simply host the code on GitHub so that you learn to use Git.

2. How to Learn About Data Collection and Wrangling (Cleaning)

(Estimated time: 2 months)

(Estimated time: 2 months)

A significant part of data science work is centered around finding apt data that can help you solve your problem. You can collect data from different legitimate sources—scraping (if the website allows), APIs, Databases, and publicly available repositories.

Once you have data in hand, an analyst will often find themself cleaning dataframes, working with multi-dimensional arrays, using descriptive/scientific computations, and manipulating dataframes to aggregate data.

Data are rarely clean and formatted for use in the “real world”. Pandas and NumPy are the two libraries that are at your disposal to go from dirty data to ready-to-analyze data.

As you start feeling comfortable writing Python programs, feel free to start taking lessons on using libraries like **pandas** and **numpy**.

Resources to learn about data collection and cleaning:

freeCodeCamp course on learning Numpy, Pandas, matplotlib, and seaborn [free].

Practical tutorial on data manipulation with NumPy and Pandas in Python from HackerEarth.

Kaggle pandas tutorial [free]—A short and concise hands-on tutorial that will walk you through commonly used data manipulation skills.

Data Cleaning course by Kaggle.

Coursera course on Introduction to Data Science in Python —This is the first course in

the Applied Data Science with Python Specialization.

Data collection project Ideas:

Collect data from a website/API (open for public consumption) of your choice, and transform the data to store it from different sources into an aggregated file or table (DB). Example APIs include TMDB, quandl, Twitter API, and so on.

Pick any publicly available dataset and define a set of questions that you'd want to pursue after looking at the dataset and the domain. Wrangle the data to find out answers to those questions using Pandas and NumPy.

3. How to Learn About Exploratory Data Analysis, Business Acumen, and Storytelling

(Estimated time: 2–3 months)

The next stratum to master is data analysis and storytelling. Drawing insights from the data and then communicating the same to management in simple terms and visualizations is the core responsibility of a Data Analyst.

The storytelling part requires you to be proficient with data visualization along with excellent communication skills.

Specific exploratory data analysis and storytelling topics to learn include:

storytelling topics to learn include:

Exploratory data analysis—defining questions, handling missing values, outliers, formatting, filtering, univariate and multivariate analysis.

Data visualization—plotting data using libraries like matplotlib, seaborn, and plotly. Know how to choose the right chart to communicate the findings from the data.

Developing dashboards—a good percent of analysts only use Excel or a specialized tool like Power BI and Tableau to build dashboards that summarise/aggregate data to help management make decisions.

Business acumen: Work on asking the right questions to answer, ones that actually target the business metrics. Practice writing clear and concise reports, blogs, and presentations.

Resources to learn more about data analysis:

Learn data analysis with Python in this free course on the freeCodeCamp YouTube channel.

Data Analysis with Python—by IBM on Coursera. The course covers wrangling, exploratory analysis, and simple model development using python.

Data Visualization—by Kaggle. Another interactive course that lets you practice all the commonly used plots.

Build product sense and business acumen with these books: **Measure what matters,** Decode and conquer, Cracking the PM interview**.**

Data analysis project ideas

Exploratory analysis on movies dataset to find the formula to create profitable movies (use it as inspiration), use datasets from healthcare, finance, WHO, past

census, Ecommerce, and so on.

Build dashboards (jupyter notebooks, excel, tableau) using the resources provided above.

4. How to Learn About Data Engineering

(Estimated time: 4–5 months)

Data engineering underpins the R&D teams by making clean data accessible to research engineers and scientists at big data-driven firms. It is a field in itself and you may decide to skip this part if you want to focus on just the statistical algorithm side of the problems.

Responsibilities of a data engineer comprise building an efficient data architecture, streamlining data processing, and maintaining large-scale data systems.

Engineers use Shell (CLI), SQL, and Python/Scala to create ETL pipelines, automate file system tasks, and optimize the database operations to make them high-performance.

Another crucial skill is implementing these data architectures which demand proficiency in cloud service providers like AWS, Google Cloud Platform, Microsoft Azure, and others.

Resources to learn Data Engineering:

Data Engineering Nanodegree by Udacity—as far as a compiled list of resources is concerned, I have not come across a better-structured course on data engineering that

covers all the major concepts from scratch.

Data Engineering, Big Data, and Machine Learning on GCP Specialization—You can complete this specialization offered by Google on Coursera that walks you through all the major APIs and services offered by GCP to build a complete data solution.

Data Engineering project ideas/certifications to prepare for:

AWS Certified Machine Learning (300 USD)—A proctored exam offered by AWS, adds some weight to your profile (doesn't guarantee anything, though), requires a decent understanding of AWS services and ML.

Professional Data Engineer—Certification offered by GCP. This is also a proctored exam and assesses your abilities to design data processing systems, deploying machine learning models in a production environment, and ensure solutions quality and automation.

5. How to Learn About Applied Statistics and Mathematics

(Estimated time: 4–5 months)

Statistical methods are a central part of data science. Almost all data science interviews predominantly focus on descriptive and inferential statistics.

People often start coding machine learning algorithms without a clear understanding of underlying statistical and mathematical methods that explain the working of those algorithms. This, of course, isn't the best way to go about it.

Topics you should focus on in Applied Statistics and math:

Descriptive Statistics—to be able to summarise the data is powerful, but not always. Learn about estimates of location (mean, median, mode, weighted statistics, trimmed statistics), and variability to describe the data.

Inferential statistics—designing hypothesis tests, A/B tests, defining business metrics, analyzing the collected data and experiment results using confidence interval, p-value, and alpha values.

Linear Algebra, Single and multi-variate calculus to understand loss functions, gradient, and optimizers in machine learning.

Resources to learn about Statistics and math:

Learn college-level statistics in this free 8-hour course on the freeCodeCamp YouTube channel

[Book] Practical statistics for data science **(highly recommend)—**A thorough guide on all the important statistical methods along with clean and concise applications/examples.

[Book] Naked Statistics—a non-technical but detailed guide to understanding the impact of statistics on our routine events, sports, recommendation systems, and many more instances.

Statistical thinking in Python—a foundation course to help you start thinking statistically. There is a second part to this course as well.

Intro to Descriptive Statistics— offered by Udacity. Consists of video lectures explaining widely used measures of location and variability(standard deviation, variance, median absolute deviation).

Inferential Statistics, Udacity—the course consists of video lectures that educate you on drawing conclusions from data that might not be immediately obvious. It focuses on developing hypotheses and use common tests such as t-tests, ANOVA, and regression.

And here's a guide to statistics for data science to help you get started down the right path.

Statistics project ideas:

Solve the exercises provided in the courses above and then try to go through a number of public datasets where you can apply these statistical concepts. Ask questions like “Is there sufficient evidence to conclude that the mean age of mothers giving birth in Boston is over 25 years of age at the 0.05 level of significance”?

Try to design and run small experiments with your peers/groups/classes by asking them to interact with an app or answer a question. Run statistical methods on the collected data once you have a good amount of data after a period of time. This might be very hard to pull off but should be very interesting.

Analyze stock prices, cryptocurrencies, and design hypothesis around the average return or any other metric. Determine if you can reject the null hypothesis or fail to do so using critical values.

6. How to Learn About Machine Learning and AI

(Estimated time: 4–5 months)

After grilling yourself and going through all the major aforementioned concepts, you

should now be ready to get started with the fancy ML algorithms.

There are three major types of learning:

There are three major types of learning:

Supervised Learning—includes regression and classification problems. Study simple linear regression, multiple regression, polynomial regression, naive Bayes, logistic regression, KNNs, tree models, ensemble models. Learn about evaluation metrics.

Unsupervised Learning—Clustering and dimensionality reduction are the two widely used applications of unsupervised learning. Dive deep into PCA, K-means clustering, hierarchical clustering, and gaussian mixtures.

Reinforcement learning (can skip*)—helps you build self-rewarding systems. Learn to optimize rewards, using the TF-Agents library, creating Deep Q-networks, and so on.

The majority of the ML projects need you to master a number of tasks that I've explained in this blog.

Resources to learn about Machine Learning:

Here's a free full course on Machine learning in Python with ScikitLearn on the freeCodeCamp YouTube channel.

[book] Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition—one of my all-time favorite books on machine learning. Doesn't only cover the theoretical mathematical derivations, but also showcases the implementation of algorithms through examples. You should solve the exercises given at the end of each chapter.

Machine Learning Course by Andrew Ng—the go-to course for anyone trying to learn machine learning. Hands down!

Introduction to Machine Learning—Interactive course by Kaggle.

Intro to Game AI and Reinforcement Learning—another interactive course on Kaggle on reinforcement learning.

Deep Learning Specialization by deeplearning.ai

For those of you who are interested in further diving into deep learning, you can start off by completing this specialization offered by deeplearning.ai and the Hands-ON book. This is not as important from a data science perspective unless you are planning to solve a computer vision or NLP problem.

Deep learning deserves a dedicated roadmap of its own. I'll create that with all the fundamental concepts soon.

Track your learning progress

I've also created a learning tracker for you on Notion. You can customize it to your needs and use it to track your progress, have easy access to all the resources and your projects.

Find the learning tracker here.

Also, here's the video version of this blog:

Data Science with Harshit

This is just a high-level overview of the wide spectrum of data science. You might want to deep dive into each of these topics and create a low-level concept-based plan for each of the categories.

If this tutorial was helpful, you should check out my data science and machine learning courses on Wiplane Academy. They are comprehensive yet compact and helps you build a solid foundation of work to showcase.

Harshit Tyagi

Web and Data Science Consultant | Instructional Design