Data Science Roadmap For Beginners

This book is a structured roadmap designed to help you learn data science in a clear and organized way. Whether you're just starting out or looking to deepen your knowledge, this guide will take you through the essential concepts, tools, and techniques needed to build a strong foundation.

Learn. Apply. Grow.

Curated collection by
Sai Ram Penjarla

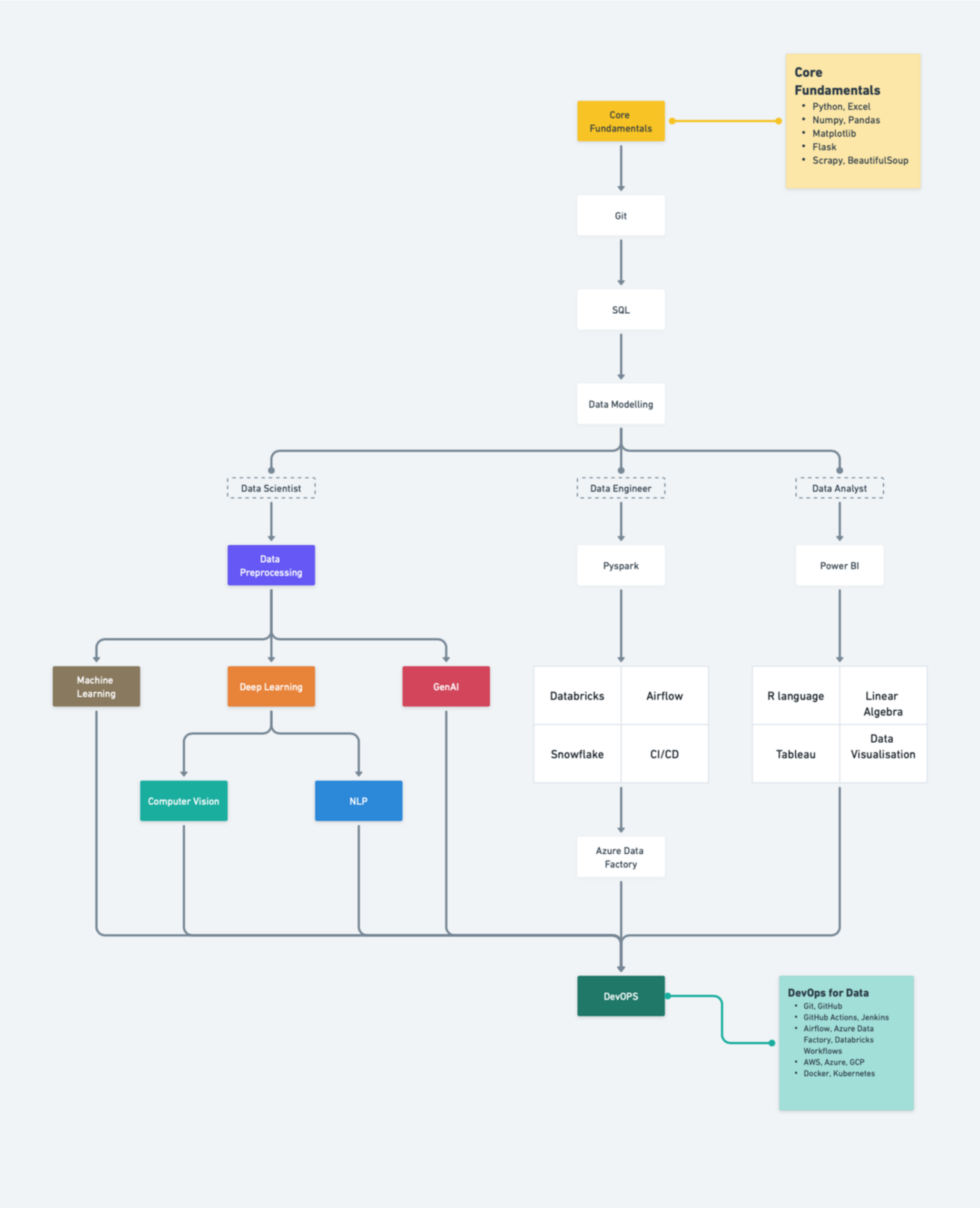
Chapter 1

Where Should I Start?

When you're just starting your data science journey, the first question you'll likely face is **"Where should I start?"** Here's how to approach it:

Follow the Roadmap for a Structured Learning Journey

Learning data science can feel overwhelming with the vast number of topics and resources available. To simplify your journey, this roadmap provides a clear, step-by-step guide to mastering essential concepts. Whether you're a beginner or an experienced professional, following this structured path will help you build a strong foundation and progress efficiently. Stick to the roadmap, tackle each topic systematically, and ensure you're gaining practical experience along the way. By following this structured approach, you'll avoid unnecessary confusion and maximize your learning potential.



Use These Curated Links to Learn Each Topic Effectively

Having the right resources is crucial for effective learning, which is why we've provided a carefully curated list of study materials for each topic. These links include tutorials, courses, videos, and documentation from trusted sources, ensuring you get high-quality information. Instead of spending hours searching for the best materials, you can focus on learning directly from these expert-recommended resources. Make sure to explore these links, practice regularly, and apply your knowledge through projects to solidify your understanding

✓ Core Fundamentals

Python

- Programming with Mosh YouTube video
- Amulya's Academy YouTube playlist
- Codecademy website
- HackerRank competitive programming practice

SQL

- Programming with Mosh
- Khan Academy SQL Course
- SQLZoo Interactive Tutorial

Excel

- Excel Exposure
- Microsoft Excel Training
- Chandoo.org Excel Tutorials

NumPy

- NumPy Tutorial by Corey Schafer
- Official NumPy Documentation
- DataCamp’s Intro to NumPy

Pandas

- Pandas Tutorial by Corey Schafer
- Official Pandas Documentation
- Data School’s Pandas Video Series

Matplotlib

- Matplotlib Tutorial by Corey Schafer
- Official Matplotlib Documentation
- Real Python’s Guide to Matplotlib

Flask

- Flask Tutorial by Corey Schafer
- Official Flask Documentation
- Miguel Grinberg’s Flask Mega-Tutorial

BeautifulSoup

- BeautifulSoup Tutorial by Corey Schafer
- Official BeautifulSoup Documentation
- Real Python’s Web Scraping Guide

Git

- Programming with Mosh
- Official Git Documentation
- Atlassian Git Tutorials

✓ Data Modelling

- Edureka YouTube

✓ Data Preprocessing

- Krish Naik
- Feature Engineering and Data Preprocessing (Kaggle Learn)

✓ Data Science Basics

Machine Learning Basics

Regression

- Regression Tutorial by StatQuest
- Linear Regression in Python by freeCodeCamp
- Introduction to Regression Analysis (Khan Academy)

Ensemble Methods

- Ensemble Learning Explained by StatQuest
- Bagging and Boosting Explained by Kris Naik
- Ensemble Methods Overview by Data School

Bagging, Boosting

- Bagging and Boosting Tutorial by StatQuest
- Understanding Random Forests (related to bagging) by StatQuest
- Boosting Algorithms Explained by Kris Naik

SVM

- Support Vector Machines Explained by StatQuest
- SVM Tutorial by freeCodeCamp
- SVMs in Python by Data School

K-Means

- K-Means Clustering by StatQuest
- K-Means Clustering Explained by freeCodeCamp
- Clustering with K-Means by Data School

XGBoost

- XGBoost Tutorial by Krish Naik
- XGBoost in Python by Data School
- XGBoost Explanation by StatQuest

Deep Learning Basics

TensorFlow

- TensorFlow Tutorial by freeCodeCamp
- TensorFlow 2.0 Complete Course by freeCodeCamp
- TensorFlow Crash Course by Amulya's Academy

ANN (Artificial Neural Networks)

- Artificial Neural Networks Explained by 3Blue1Brown
- Neural Networks Demystified by Welch Labs
- ANN Fundamentals by Kris Naik

CNN (Convolutional Neural Networks)

- Convolutional Neural Networks Explained by StatQuest
- CNN Tutorial by freeCodeCamp
- CNNs for Beginners by Simplilearn

Transfer Learning

- Transfer Learning Explained by Data School
- Transfer Learning with Hugging Face by Hugging Face
- Transfer Learning Tutorial by Kris Naik

Huggingface

- Hugging Face Course by Hugging Face
- Hugging Face Transformers Tutorial by Data School
- Hugging Face Overview by Kris Naik

Computer Vision Basics

OpenCV

- Murtaza's Workshop
- OpenCV Crash Course by Tech With Tim
- OpenCV Python Tutorial by Programming with Mosh

MediaPipe

- MediaPipe Tutorial by Murtaza's Workshop
- MediaPipe in Python by Tech With Tim
- MediaPipe Hands by Google Developers

OpenPose

- OpenPose Tutorial by CMU Graphics
- OpenPose Installation & Guide on GitHub
- OpenPose Overview by Kris Naik

NLP Basics

NER (Named Entity Recognition)

- NER Explained by Data School
- NER Tutorial by Kris Naik
- NER in Python by Tech With Tim

Sentiment Analysis

- Sentiment Analysis with Python by freeCodeCamp
- Sentiment Analysis Tutorial by Data School
- Building a Sentiment Analysis Model by Kris Naik

Gen AI

Open AI

- OpenAI Api Crash Course by CodeBasics

LangChain

- LangChain Crash Course by freeCodeCamp

RAG

- RAG Fundamentals by freeCodeCamp

Prompt Engineering

- Prompt Engineering by Ben AI

Vector DB

- Chroma - Vector Database BugBytes

✓ Intermediate

coming soon…

✓ Advanced

coming soon…

Chapter 2

Portfolio

A well-structured portfolio is crucial for showcasing your data science expertise. While a resume highlights your qualifications, your portfolio demonstrates practical experience. Recruiters seek candidates who can apply their skills to real-world problems, and projects are the best way to prove this.

Why Projects are Essential

- ✓ **Demonstrate Problem-Solving:** Projects illustrate your approach to a problem, including data cleaning and analysis, application of machine learning techniques, and interpretation of results.
- ✓ **Compensate for Lack of Experience:** For newcomers to data science, hands-on projects can distinguish you from other candidates.
- ✓ **Provide Interview Talking Points:** Projects provide concrete examples to discuss during interviews, allowing you to support your answers with real-world implementations.

Elements of a Strong Portfolio

- ✓ **Project Diversity:** Include projects spanning various domains like finance, healthcare, retail, and NLP to demonstrate adaptability.
- ✓ **End-to-End Implementation:** Focus on projects covering data collection, preprocessing, feature engineering, model building, and deployment.
- ✓ **Clear Documentation:** Use Jupyter notebooks, README files, and Markdown to provide structured explanations of your projects.
- ✓ **High-Quality Code:** Maintain clean, well-commented code following best practices, and organize your repositories effectively on GitHub.
- ✓ **Deployment (Optional but Recommended):** Deploy your projects using tools like Flask, FastAPI, Streamlit, or cloud platforms (AWS, Azure, Heroku) to showcase practical application.

Finding Project Ideas

- ✓ **Real-World Datasets:** Utilize datasets from Kaggle, Google Dataset Search, or government open data portals.
- ✓ **Industry Case Studies:** Recreate industry case studies involving business problem-solving.
- ✓ **Open-Source Contributions:** Explore open-source repositories and contribute to existing models.
- ✓ **Curated Project Lists:** Refer to curated lists (like the one at this link) for inspiration.

Further Portfolio Enhancement

- ✓ **Blogs and Case Studies:** Explain your projects in detail on platforms like Medium, Hashnode, or LinkedIn.
- ✓ **Open-Source Contributions:** Collaborating on GitHub enhances credibility and networking opportunities.
- ✓ **Personal Website:** Create a professional website using Wix, WordPress, Notion, or GitHub Pages.
- ✓ **Kaggle Competitions:** Participating in Kaggle competitions, even without winning, improves problem-solving and coding skills.

Chapter 3

Soft Skills

Technical expertise is fundamental, but soft skills differentiate good candidates from great ones. Companies value professionals who can communicate effectively, collaborate within teams, and adapt to challenges.

Key Soft Skills for Data Science and Tech Roles

- ✓ **Communication:** Explaining complex technical concepts to non-technical stakeholders is crucial.
- ✓ **Problem-Solving & Critical Thinking:** Employers seek candidates who can logically break down problems and find efficient solutions.
- ✓ **Time Management & Productivity:** Managing multiple projects and deadlines is common in the industry.
- ✓ **Adaptability & Continuous Learning:** The tech world is constantly evolving, requiring ongoing learning.
- ✓ **Collaboration & Teamwork:** Many projects involve working with cross-functional teams, including business analysts and engineers.
- ✓ **Presentation Skills:** Presenting data insights concisely and effectively is a significant advantage.

Improving Your Soft Skills

- ✓ **Public Speaking Practice:** Present your projects to peers or participate in online meetups.
- ✓ **Regular Writing:** Improve articulation by writing on LinkedIn, Medium, or your blog.
- ✓ **Engage in Technical Discussions:** Participate in forums like Stack Overflow, Kaggle, and LinkedIn groups.
- ✓ **Teamwork Experience:** Join open-source projects or hackathons to enhance collaboration skills.

- ✓ **Formal Training:** Consider courses on communication, leadership, and negotiation from platforms like Coursera, Udemy, or edX.

Chapter 4

Interviews

A structured approach to interview preparation is vital for success. Interviews typically assess technical skills, problem-solving abilities, and behavioral aspects.

Step 1: Master the Fundamentals

- ✓ **Programming:** Demonstrate proficiency in Python or R, including loops, functions, OOP, and libraries like Pandas and NumPy.
- ✓ **SQL:** SQL is essential for data roles. Practice writing complex queries, joins, window functions, and optimizing query performance.
- ✓ **Data Structures & Algorithms:** Basic knowledge of lists, dictionaries, stacks, queues, and trees is important, though less extensive than in software engineering interviews.
- ✓ **Statistics & Probability:** Understand concepts like hypothesis testing, distributions, p-values, and Bayesian probability.
- ✓ **Machine Learning:** Be familiar with key algorithms like linear regression, decision trees, random forests, gradient boosting, and neural networks.
- ✓ **Big Data & Cloud (Optional):** Basic familiarity with platforms like AWS, GCP, Hadoop, and Spark is a plus.

Step 2: Practice Case Studies & Real-World Problems

Companies often use case studies to assess business problem-solving skills. Prepare by:

- ✓ Solving real-world case studies from McKinsey, BCG, and Kaggle.
- ✓ Practicing feature engineering and model optimization.
- ✓ Learning to explain solutions concisely, covering problem definition, approach, results, and impact.

Step 3: Mock Interviews & Behavioral Questions

Mock interviews refine answers and build confidence. Common behavioral questions include:

- ✓ "Tell me about yourself."

- ✓ "Describe a time you worked in a team and faced a challenge."
- ✓ "How do you handle tight deadlines?"
- ✓ "Why do you want to join our company?"

Use the STAR method (Situation, Task, Action, Result) for structured and concise answers.

Step 4: System Design Basics (For AI/ML Roles)

Basic system design knowledge is helpful for AI/ML roles. Understand concepts like:

- ✓ API design and microservices.
- ✓ Database scaling, indexing, and caching.
- ✓ Model deployment using Flask, FastAPI, Docker, or Kubernetes.

Step 5: Job Applications & Networking

Strategic applications increase your chances of interviews:

- ✓ Customize resumes for each role, highlighting relevant projects and skills.
- ✓ Track applications in a spreadsheet.
- ✓ Network actively on LinkedIn and at meetups.
- ✓ Seek referrals, as they significantly increase interview opportunities.

Step 6: Continuous Learning

- ✓ Follow AI and data science blogs, research papers, and YouTube channels.
- ✓ Subscribe to newsletters like KDnuggets, Towards Data Science, and DeepLearning.AI.

- ✓ Work on new projects and explore emerging trends in AI, MLOps, and cloud computing.

Chapter 5

Frequently Asked Questions

Should I learn Python or R?

Both Python and R are excellent languages for data science. However, Python is more widely used in industry due to its vast ecosystem of libraries (NumPy, Pandas, Scikit-learn) essential for data manipulation, analysis, and machine learning. R excels in statistical analysis and visualization, particularly in academic and research settings. For beginners, Python is generally recommended for its simplicity and versatility.

Do I need a strong background in programming?

A basic understanding of programming is necessary, but you don't need to be an expert. Python is beginner-friendly, with numerous online resources and tutorials. Your programming skills will improve as you delve deeper into data science. You can begin by focusing on learning data science tools without initially mastering advanced coding concepts.

How much mathematics (linear algebra, calculus, statistics) is required?

A solid understanding of linear algebra (vectors, matrices) and calculus (derivatives, integrals) is essential for machine learning and deep learning. For basic data science tasks, a good grasp of statistics and probability is more important. Focus on statistics and probability initially, and delve into linear algebra and calculus as you progress to more complex topics like machine learning algorithms.

Is deep learning necessary, or should I focus on traditional machine learning first?

Start with traditional machine learning. Techniques like linear regression, decision trees, and clustering are foundational for understanding how machine learning models work. Deep learning involves more complex architectures (neural networks) and requires more computational resources. Once comfortable with traditional machine learning, you can explore deep learning.

Which Python libraries should I focus on?

The essential Python libraries for data science are:

- ✓ NumPy: Numerical operations and arrays
- ✓ Pandas: Data manipulation and dataframes
- ✓ Matplotlib and Seaborn: Data visualization
- ✓ Scikit-learn: Traditional machine learning algorithms
- ✓ TensorFlow and PyTorch: Deep learning and neural networks

Is Hadoop or Spark required for data science jobs?

Hadoop and Spark are used in big data environments. Learning Spark can be beneficial for working with massive datasets, especially in industries like e-commerce or finance. However, they are not typically required for beginner-level data science roles.

Should I learn cloud platforms like AWS, GCP, or Azure?

Learning cloud platforms (AWS, GCP, Azure) is helpful but not mandatory for beginners. Many companies use cloud platforms for scalability and storage, so basic cloud services and storage knowledge is a plus. Focus on data science fundamentals first.

How can I find real-world projects to work on?

Find real-world projects through:

My Website

- ✓ GitHub repositories
- ✓ Data science communities/forums (Kaggle, Data Science Central)
- ✓ Hackathons and competitions

Can I get a data science job as a fresher?

Yes, entry-level data science roles are available for freshers. A strong portfolio and practical experience are essential. Internships, Kaggle competitions, and personal projects can build your credentials.

Do companies hire freshers without experience in data science?

Some companies hire freshers for entry-level roles, especially those with relevant skills, certifications, or a solid portfolio. However, competition is tough, so practical experience and projects are key.

How many projects should I include in my portfolio?

Start with 3-5 projects showcasing your abilities in data cleaning, machine learning, and data visualization. Quality is more important than quantity.

What are some good project ideas for beginners?

Beginner project ideas:

- ✓ Titanic Survival Prediction
- ✓ Stock Price Prediction
- ✓ Image Classification with MNIST
- ✓ Customer Segmentation

Should I create a GitHub repository for my projects?

Absolutely! A GitHub repository is essential for showcasing your code and projects, demonstrating your ability to collaborate and share your work.

Will AI and automation replace data scientists?

While AI and automation are changing industries, data science will evolve. Skilled professionals will always be needed to build and refine AI systems. Automation will handle repetitive tasks, allowing data scientists to focus on higher-level work.

What should I include in my resume as a fresher?

Freshers should focus on projects, certifications, and skills. Mention any Kaggle competitions or open-source contributions.

Final Thoughts

Success in data science requires a combination of technical skills, a strong portfolio, well-developed soft skills, and thorough interview preparation. Consistent work on real-world projects, sharing your learnings, and community engagement provide a competitive edge. Treat interview preparation as an ongoing process of skill development and adaptation to industry trends.