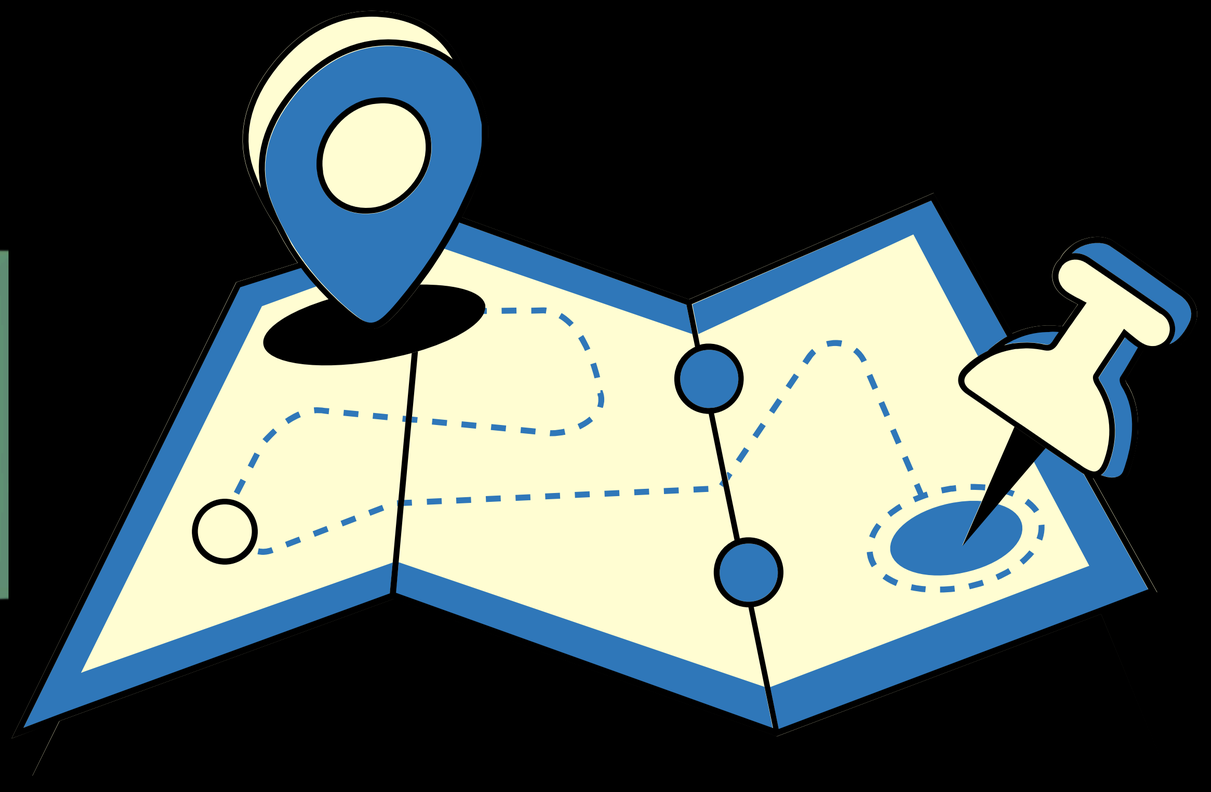
A Beginner to Upper Intermediate

Data Science Roadmap



To Data & Beyond

Youssef Hosni

A Beginner to Upper Intermediate Data Science Roadmap

A Beginner to Upper Intermediate Data Science Roadmap

By: Youssef Hosni

To Data & Beyond

Brief of Contents

1. Introduction to Data Science & Data Methodology
2. Mathematics for Data Science
3. Python Fundamentals
4. Python for Data Science
5. Software Engineering Basics
6. Database & SQL Fundamentals
7. Data Cleaning & Preprocessing
8. Feature Engineering
9. Mastering Machine Learning
10. Deep Learning Fundamentals
11. Generative AI & Large Language Models (LLMs) Fundamentals
12. Machine Learning Operations (MLOps)
13. Building Your Data Science Portfolio
14. Getting Ready for the Market

Table of Contents

Brief of Contents **3**
Table of Contents **4**
About this book **8**
About the author **9**
1. Introduction to Data Science & Data Methodology **11**
1.1. What is Data Science? 11
1.2. Data Science Methodology 12
1.3. Data Science for Business Innovation 13
1.4. 5 Business Basics for Data Scientists 14
1.5. Optional Resources 14
1.6. Action Points 14
2. Mathematics for Data Science **15**
2.1. Mathematics for Machine Learning and Data Science Specialization 16
2.2. Mathematics Interview Questions & Answers 17
3. Additional Resources 18
3.1. Mathematics for Machine Learning Book 18
3.2. Practical Statistics For Data Scientists 19
3.3. Matrix Methods In Data Analysis, Signal Processing, And Machine Learning — MIT 20
3. Python Fundamentals 22
3.1. Python Fundamentals Skills 22
3.2. Compulsory Resources 23
3.2.1. The Complete Python Developer 23
3.3. Optional Resources 24

3.3.1. Modern Python Cookbook — Third Edition 25
3.4. Putting it into Action 26
4. Python for Data Science 28
4.1. Python for Data Science 28
4.2. Compulsory Resources 29
4.2.1. Python for Data Science and Machine Learning Bootcamp 29
4.2.2. Applied Data Science with Python Specialization 30
4.3. Additional Resources 31
4.3.1. Python Data Science Handbook 31
5. Software Engineering Basics for Data Science 33
5.1. Software Engineering for Data Scientists Book 33
5.2. Version Control 35
5.3. Infrastructure & Resource Management 36
5.4. Linux Basics & Shell Scripting 37
6. Database & SQL Fundamentals 39
6.1. SQL Basics 39
6.2. Intermediate & Advanced SQL 40
6.3. Database for Data Science 41
6.4. SQL Case Studies 41
6.5. SQL Practice 42
6.6. Advanced SQL Learning Material 43
6.6.1. Advanced Database Systems—CMU 43
6.6.2. SQL Database Programming 43
7. Data Cleaning & Preprocessing 46
7.1. Data Exploration 46
7.2. Data Cleaning & Data Preprocessing 47
7.3. Optional Learning Resources 47
7.3.1. Bad Data 47
7.3.2. Data Wrangling with Python 49
Table of contents: 51
7.4. Putting it into Action 51
8. Feature Engineering 52
8.1. Feature Engineering For Machine Learning 52
8.2. Machine Learning with Imbalanced Data 53
8.3. Feature Selection for Machine Learning 55
8.4. Additional Resources 56
8.4.1. Python Feature Engineering Cookbook 56
8.4.2. Feature Engineering and Selection 58
8.4.3. Feature Engineering for Machine Learning 60
9. Mastering Machine Learning 62
9.1. Machine Learning Specialization 62

9.2. Additional Resources 63
9.2.1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow 63
9.3. Putting it into Action 64
10. Deep Learning Fundamentals 66
10.1. Deep Learning Specialization 66
10.2. Deep Learning for Natural Language Processing 67
10.3. Deep Learning for Computer Vision 68
Course Information: 69
10.4. Additional Resources 70
10.4.1. Stanford University CS231n: Deep Learning for Computer Vision 70
10.4.2. CS224n: Natural Language Processing with Deep Learning 71
10.4.3. The Deep Learning Book 71
10.4.4. Deep Learning with Python Book 72
10.5. Putting it into Actions 73
11. Generative AI & Large Language Models (LLMs) Fundamentals 75
11.1. Generative AI with Large Language Models 75
11.2. Prompt Engineering Guide 76
11.3. Learn RAG From Scratch 77
11.4. Fine Tuning LLM Models—Generative AI Course 78
11.5. Additional Resources 79
11.5.1. Important Books 79
1. Build LLM from Scratch 79
2. Hands-on Large Language Models 80
3. LLM Engineering Handbook 81
11.5.2. Important University Courses 82
1. Advanced NLP—Carnegie Mellon University 82
2. Recent Advances on Foundation Models—University of Waterloo 83
3. Large Language Model Agents—University of California, Berkeley 83
11.6. Putting it into Action 84
12. Machine Learning Operations (MLOps) Fundamentals 86
12.1. Version Control for Machine Learning 86
2. Continuous Integration & Continuous Delivery (CI/CD) Tools 87
12.3. Infrastructure & Resource Management for Machine Learning 88
12.4. Machine Learning Monitoring & Observability Tools 89
12.5. Managing Machine Learning Projects & Pipelines 90
12.6. Machine Learning Security & Compliance Tools 91
12.7. Putting it into Action 91
13. Building Your Data Science Portfolio 93
13.1. Importance of Having a Data Science Portfolio Project 93
13.2. Select a Domain of Interest 94
Actions: 95

13.3. Prioritize Your Interest Based on the Market Demand 95
13.4. Define Important Case Studies In the Market 96
13.5. Choose Different Case studies 97
13.6. Brainstorm Data Science Solutions 98
Actions: 99
13.7. Determine Success Metrics 99
Actions: 99
13.8. Collect the Dataset 100
13.9. Clean & Prepare the Data 101
Actions: 101
13.10. Train & Evaluate the Model 102
13.11. Make them end-to-end 102
12. Publish & Talk About It 103
14. Getting Ready for the Market 105
14.1. Starting a Career in Data Science: Project Portfolio, Resume, and Interview Process—Course 105
14.2. Data Science Interview Pro—YouTube Channel 106
14.3. Ace the Data Science Interview—Book 107
14.4. Top 30 Generative AI Interview Questions and Answers for 2025 108
Afterword 110

About this book

Whether you're a recent graduate or a professional looking to make a career change, the field of Data Science and AI offers a wide range of exciting and lucrative opportunities.

In this book, I will provide you with a comprehensive guide that will provide you with a clear and actionable plan for building the skills and knowledge you need to succeed in this growing field. By following the steps outlined in this roadmap, you'll be well on your way to a successful and rewarding career in Data Science & AI.

This roadmap will take you to an upper intermediate level, and I truly believe you can land a job and start your career after finishing it. However, to go to an advanced level, you will need to take more in-depth courses, books, and research papers.

For each learning step, there will be compulsory material, optional material, and action points to make sure you put what you have learned into action. Also, there will be an estimated time for each of the learning resources in hours so you can calculate the time needed to finish this roadmap depending on your pace.

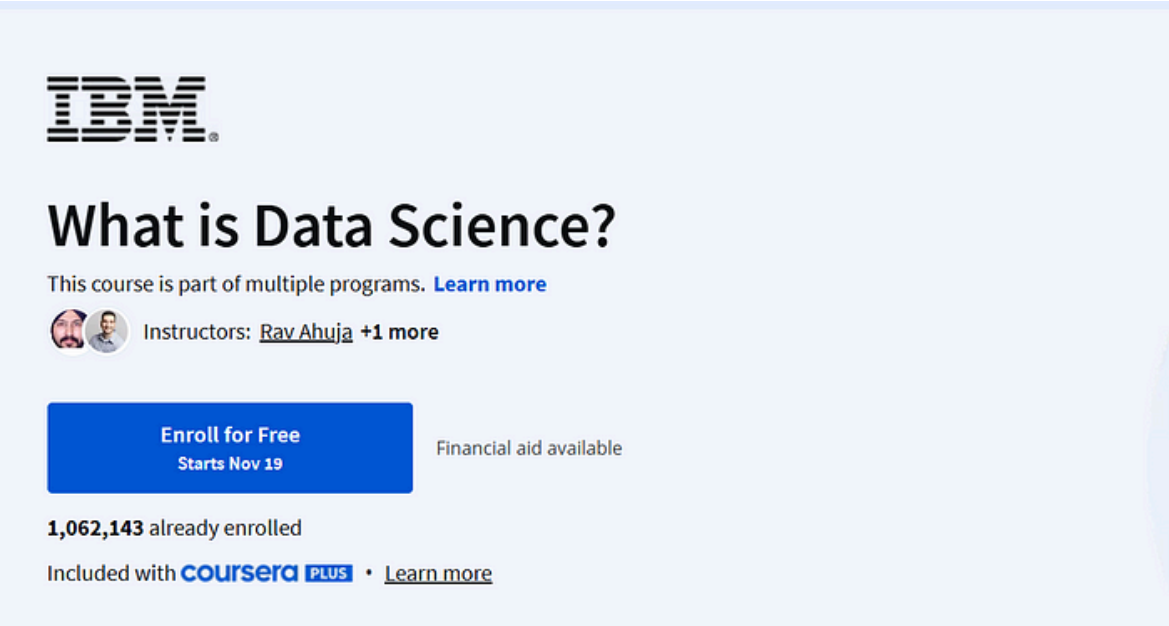
About the author

Youssef Hosni is a data scientist and machine learning researcher who has been working in machine learning and AI for more than half a decade. In addition to being a researcher and data science practitioner, Youssef has a strong passion for education. He is known for his leading data science and AI blog, newsletter, and eBooks on data science and machine learning.

Youssef is a senior data scientist at Ment focusing on building Generative AI features for Ment Products. He is also an AI applied researcher at Aalto University working on mutlimodat agents and their applications for next generation smart cities . Before that, he worked as a researcher in which he applied deep learning and computer vision techniques to medical images.

1. Introduction to Data Science & Data Methodology

1.1. What is Data Science?



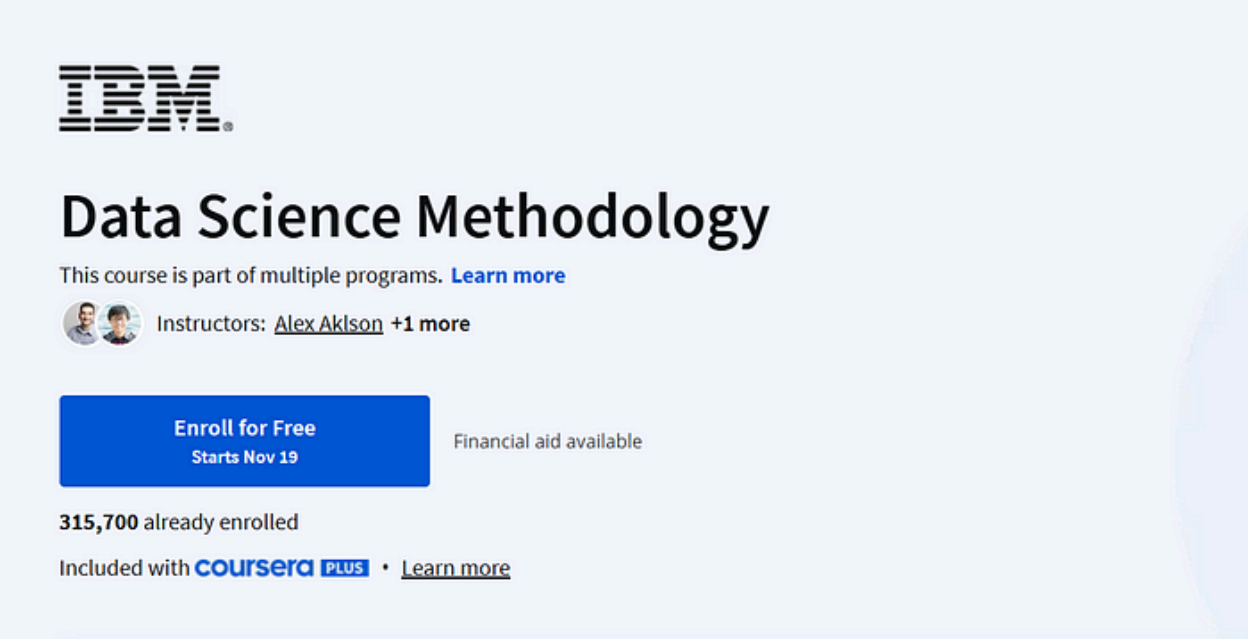
The first resource on this list is **What is Data Science?** by IBM available on Coursera. In today's world, we use Data Science to find patterns in data and make meaningful, data-driven conclusions and predictions.
This course is for everyone and teaches concepts like how data scientists use machine learning and deep learning and how companies apply data science in business. You will meet several data scientists, who will share their insights and experiences in data science. By taking this introductory course, you will begin your journey into this thriving field.
The course consists of four modules:

- Defining Data Science and What Data Scientists Do
- Data Science Topics Module
- Applications and Careers in Data Science
- Data Literacy for Data Science

Resource Information:

- Course Link
- Cost: 40$ (Financial aid available)
- Course Provider: IBM & Coursera
- Duration: 10 Hours
- Instructor: Alex Aklson & Rav Ahuja

1.2. Data Science Methodology



The second resource on this list is the Data Science Methodology course by IBM on coursera. In this course, you will learn and then apply this methodology that you can use to tackle any Data Science scenario.

You'll explore two notable data science methodologies, Foundational Data Science Methodology, and the six-stage CRISP-DM data science methodology, and learn how to apply these data science methodologies. Most established data scientists follow these or similar methodologies for solving data science problems.

Begin by learning about forming the business/research problem Learn how data scientists obtain, prepare, and analyze data. Discover how applying data science methodology practices helps ensure that the data used for problem-solving is relevant and properly manipulated to address the question.

Next, learn about building the data model, deploying that model, data storytelling, and obtaining feedback You'll think like a data scientist and develop your data science methodology skills using a real-world inspired scenario through progressive labs hosted within Jupyter Notebooks and using Python.

The course consists of four modules:

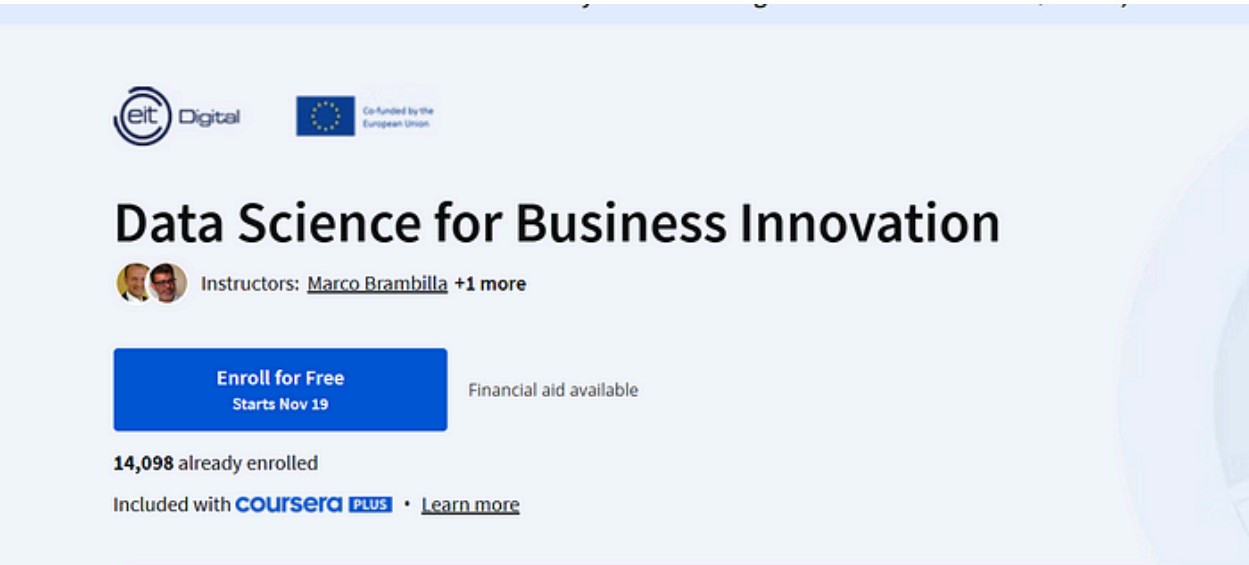
- From Problem to Approach and From Requirements to Collection
- From Understanding to Preparation and From Modeling to Evaluation
- From Deployment to Feedback and Final Evaluation

- Final Project and Assessment

Resource Information:

- Course Link
- Cost: 40$ (Financial aid available)
- Course Provider: IBM & Coursera
- Duration: 15 Hours
- Instructor: Alex Aklson & Polong Lin

1.3. Data Science for Business Innovation



The third course on our list is Data Science for Business Innovation provided by EIT Digital & Politecnico di Milano. This course is your chance to learn all about Data Science for Business innovation and future-proof your career.

The Data Science for Business Innovation nano-course is a compendium of the must-have expertise in data science for executives and managers to foster data-driven innovation. The course explains what Data Science is and why it is so hyped.

From a more technical perspective, the course covers supervised, unsupervised, and semi-supervised methods, and explains what can be obtained with classification, clustering, and regression techniques.
It discusses the role of NoSQL data models and technologies and the role and impact of scalable cloud-based computation platforms. All topics are covered with example-based lectures, discussing use cases, success stories, and realistic examples.

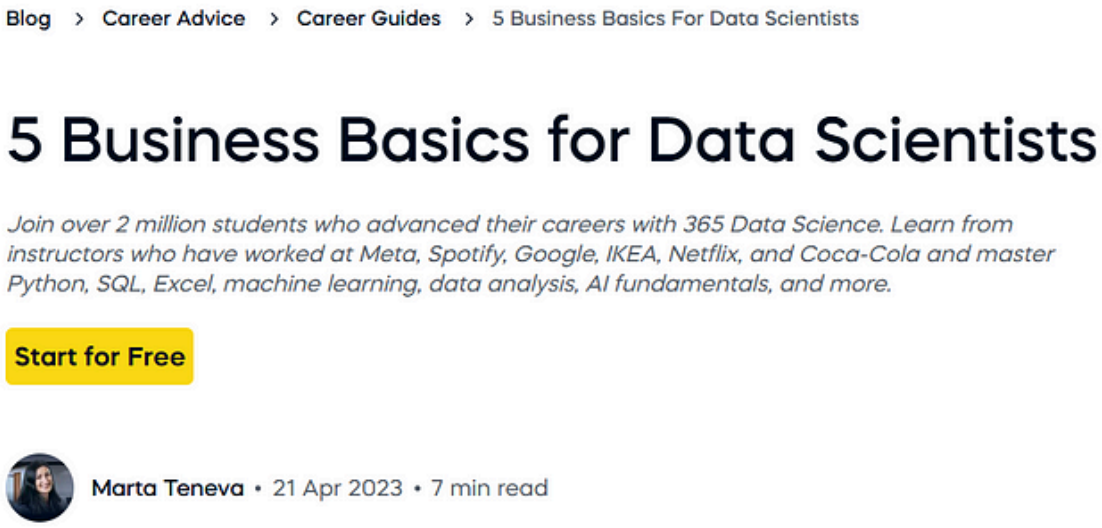
The course consists of four modules:

- Introduction to Data-driven Business
- Terminology and Foundational Concepts
- Data Science Methods for Business
- Challenges and Conclusions

Resource Information:

- Course Link
- Cost: 40$ (Financial aid available)
- Course Provider: EIT Digital & Politecnico di Milano
- Duration: 10 Hours
- Instructor: Marco Brambilla & Emanuele Della Valle

1.4. 5 Business Basics for Data Scientists



The fourth and last compulsory resource is the 5 Business Basics for Data Scientists article by 365 DataScience. This article covers 5 basic business concepts essential for data scientists.

1.5. Optional Resources

In this section, I will recommend additional resources so that you can learn more about data science methodology and business for data science. They are optional so I will add a variety of resources and you can choose the most convenient for you:

- Chapter 1 and Chapter 2 from the Designing Machine Learning Systems book
- Chapter 1 and Chapter 2 from the Data Science for Business book

1.6. Action Points

Finally, the action points in this learning step will help you understand your learning and implement the learning outcomes. Since this learning step focused on introducing you to data science and data science work the action points is to help you translate this into real understanding.

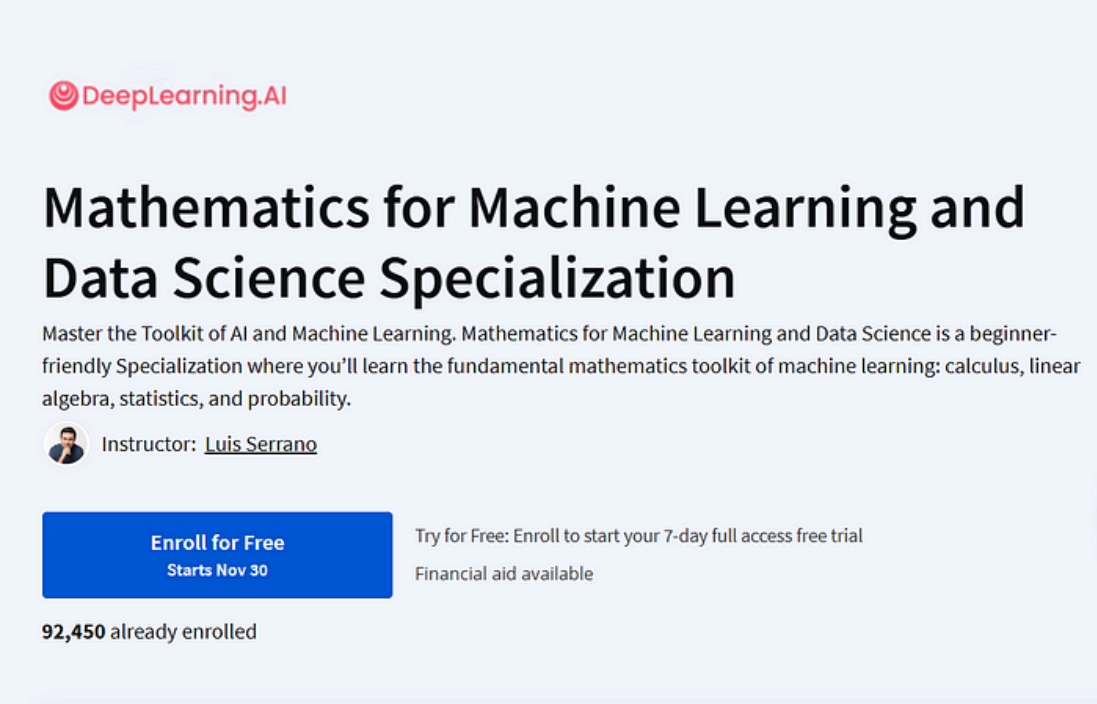
Here are my suggested action points:

- Create a LinkedIn account if you have not
- Create a professional Medium account if you have not created one before
- Choose a certain data science use case and research adn design the project life cycle
- Write and publish a blog post about this data science project lifecycle.
- Publish three LinkedIn posts based on your course notes. You can also break down your blog posts into short-form social media posts.

2. Mathematics for Data Science

In the second chapter of this book, you will learn the mathematical foundations needed to work in data science and understand machine learning and deep learning foundations. These mathematical foundations include statistics and probability, linear algebra, and calculus and optimization.

2.1. Mathematics for Machine Learning and Data Science Specialization



Mathematics for Machine Learning and Data Science is a foundational online program created by DeepLearning.AI and taught by Luis Serrano. In machine learning, you apply math concepts through programming. And so, in this specialization, you'll apply the math concepts you learn using Python programming in hands-on lab exercises. As a learner in this program, you'll need basic to intermediate Python programming skills to be successful.

Many machine learning engineers and data scientists need help with mathematics, and even experienced practitioners can feel held back by a lack of math skills.
This Specialization uses innovative pedagogy in mathematics to help you learn quickly and intuitively, with courses that use easy-to-follow visualizations to help you see how the math behind machine learning actually works.
It is recommended that you have a high school level of mathematics (functions, basic algebra) and familiarity with programming (data structures, loops, functions, conditional statements, debugging). Assignments and labs are written in Python but the course introduces all the machine learning libraries you'll use.

By the end of this Specialization, you will be ready to:

- Represent data as vectors and matrices and identify their properties like singularity, rank, and linear independence
- Apply common vector and matrix algebra operations like the dot product, inverse, and determinants
- Express matrix operations as linear transformations
- Apply concepts of eigenvalues and eigenvectors to machine learning problems including Principal Component Analysis (PCA)
- Optimize different types of functions commonly used in machine learning
- Perform gradient descent in neural networks with different activation and cost functions
- Identify the features of commonly used probability distributions
- Perform Exploratory Data Analysis to find, validate, and quantify patterns in a dataset
- Quantify the uncertainty of predictions made by machine learning models using confidence intervals, margin of error, p-values, and hypothesis testing.
- Apply common statistical methods like MLE and MAP

The specialization consists of three courses:

- Linear Algebra for Machine Learning and Data Science
- Calculus for Machine Learning and Data Science
- Probability & Statistics for Machine Learning & Data Science

Resource Information:

- Course Link
- Cost: 40$/Course (Financial aid available)
- Course Provider: Deep Learning.ai
- Duration: 1.5 months (At 20 hours/week)
- Instructor: Luis Serrano

2.2. Mathematics Interview Questions & Answers

After you studied the main mathematical foundations, it is important to explore the interview questions on this topic, especially the questions related to probability and statistics as they are common in data science interviews.
There are a lot of resources that cover this topic and here are my recommendations:

- Top Important Probability Interview Questions & Answers for Data Scientists [Mathematical Questions]
- Top Important Probability Interview Questions & Answers for Data Scientists [Conceptual Questions]
- Statistics Interview Questions & Answers for Data Scientists
- Chapters 6 & 7 from Ace the Data Science Interview Book

3. Additional Resources

3.1. Mathematics for Machine Learning Book

The first additional resource is the Mathematics for Machine Learning book by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. This book is a great choice to learn the mathematics needed to understand basic machine learning algorithms in general.

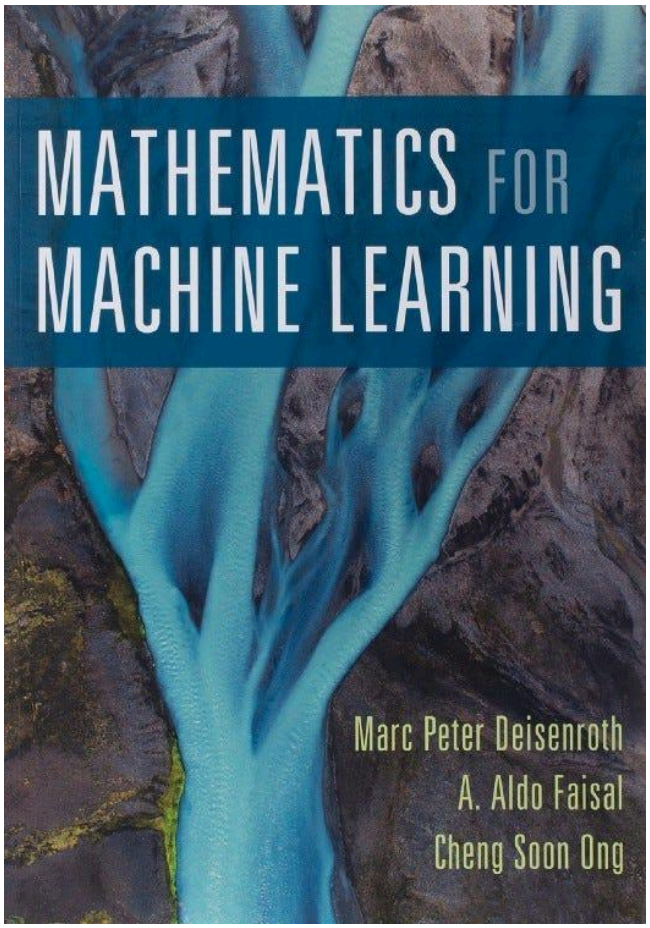


Table of Contents:

Part I: Mathematical Foundations

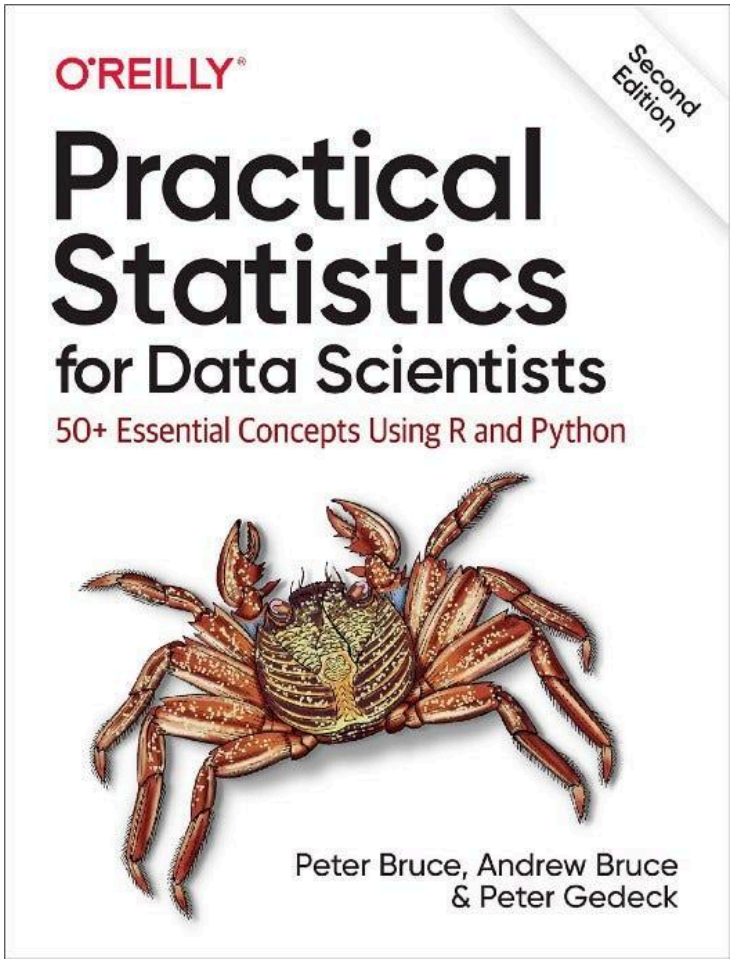
1. Introduction and Motivation
2. Linear Algebra
3. Analytic Geometry
4. Matrix Decompositions
5. Vector Calculus
6. Probability and Distribution
7. Continuous Optimization

Part II: Central Machine Learning Problems

1. When Models Meet Data
2. Linear Regression
3. Dimensionality Reduction with Principal Component Analysis
4. Density Estimation with Gaussian Mixture Models
5. Classification with Support Vector Machines

3.2. Practical Statistics For Data Scientists

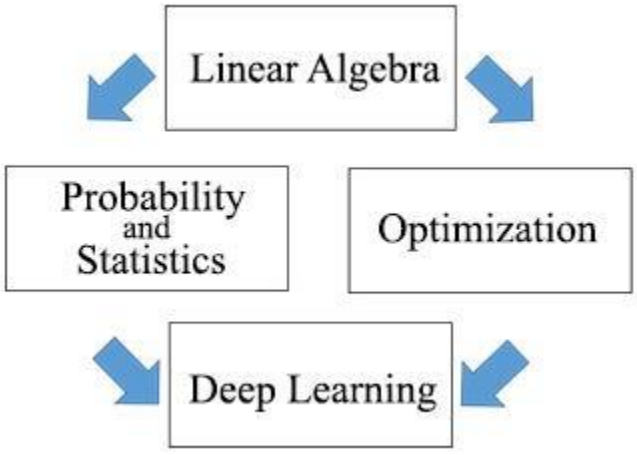
The fourth book is [Practical Statistics For Data Scientists] by Peter Bruce, Andrew Bruce, and Peter Gedeck. This book is a perfect choice to build a strong statistical foundation in a practical way.



One of the best data science statistics books is Practical Statistics for Data Scientists. This text covers a wide variety of statistical procedures used in data science, avoiding the most frequent errors.
The authors begin by explaining how data science begins with exploratory data analysis. They then move on to important topics like random sampling, experimental design, regression, classification methods, and statistical machine-learning methods that learn from data.
Whether you have R programming experience, this book is one of the best books for data science statistics. You will gain the statistical perspective that is needed to perform data scientist duties.
Table of Contents:

- Exploratory Data Analysis
- Data and Sampling Distributions
- Statistical Experiments and Significance Testing
- Regression and Prediction
- Classification
- Statistical Machine Learning
- Unsupervised Learning

3.3. Matrix Methods In Data Analysis, Signal Processing, And Machine Learning — MIT



Linear algebra concepts are key for understanding and creating machine learning algorithms, especially as applied to deep learning and neural networks. This course reviews linear algebra with applications to probability, statistics, and optimization–and above all a full explanation of deep learning.
Covered Topics:

- The Column Space of A Contains All Vectors Ax

- Multiplying and Factoring Matrices
- Orthonormal Columns in Q Give Q′Q=I
- Eigenvalues and Eigenvectors
- Positive Definite and Semidefinite Matrices
- Singular Value Decomposition (SVD)
- Eckart-Young: The Closest Rank k Matrix to A
- Norms of Vectors and Matrices
- Four Ways to Solve Least Squares Problems
- Survey of Difficulties with Ax=b
- Minimizing ‖x‖ Subject to Ax=b
- Computing Eigenvalues and Singular Values
- Randomized Matrix Multiplication
- Low-Rank Changes in A and Its Inverse
- Matrices A(t) Depending on t, Derivative = dA/dt
- Derivatives of Inverse and Singular Values
- Rapidly Decreasing Singular Values
- Counting Parameters in SVD, LU, QR, Saddle Points
- Saddle Points Continued, Maxmin Principle
- Definitions and Inequalities
- Minimizing a Function Step by Step
- Gradient Descent: Downhill to a Minimum
- Accelerating Gradient Descent (Use Momentum)
- Linear Programming and Two-Person Games
- Stochastic Gradient Descent
- Structure of Neural Nets for Deep Learning
- Backpropagation: Find Partial Derivatives
- Computing in Class [No video available]
- Computing in Class (cont.) [No video available]
- Completing a Rank-One Matrix, Circulants!
- Eigenvectors of Circulant Matrices: Fourier Matrix
- ImageNet is a Convolutional Neural Network (CNN), The Convolution Rule
- Neural Nets and the Learning Function
- Distance Matrices, Procrustes Problem
- Finding Clusters in Graphs
- Alan Edelman and Julia Language

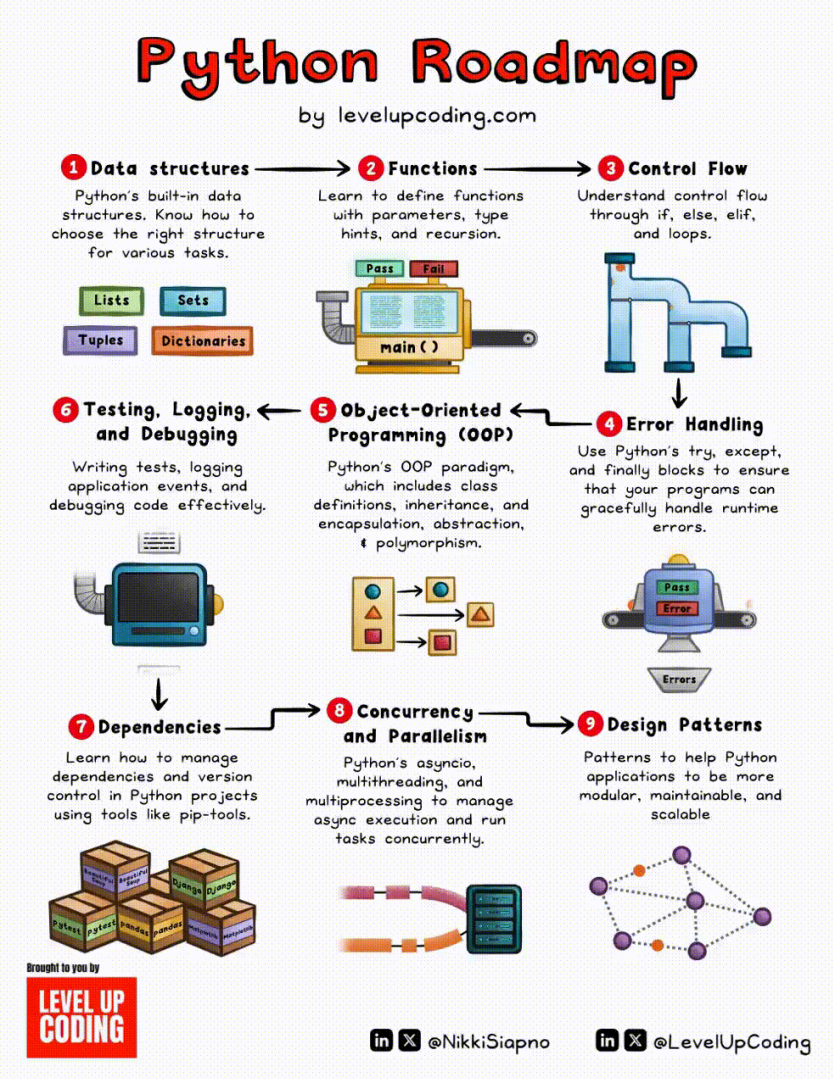
Resource Information:

- Course page
- Lecturer: Prof. Gilbert Strang
- Estimated Duration: 40 hours

3. Python Fundamentals

In the third chapter of this book, you will learn the Python foundations needed to work in data science or the tech field in general. These Python foundations include Data structures, Functions, Control flow, Error handling, Testing, logging, debugging, dependency management, Design patterns, and Object-oriented programming.

3.1. Python Fundamentals Skills



Python is a fundamental skill for your data science career. There are a lot of topics under Python Umbrella. However, I believe there are 9 main skills that you should master before the next steps.

1. Data structures: Data structures are the building blocks of software. Python's built-in data structures include lists, dictionaries, sets, and tuples. Knowing when to use each one ensures optimal performance for specific tasks.
2. Functions: Learn to define functions with parameters, type hints, and recursion. This will make your code more reusable and maintainable.
3. Control flow: Understand conditional statements (if, else, elif) and loops. These are the building blocks of logic in your code.
4. Error handling: Handle runtime errors gracefully using try, except, and finally blocks. This ensures your program can handle unexpected conditions without crashing.
5. Object-oriented programming (OOP): Dive into OOP concepts such as classes, inheritance, and encapsulation to structure your code in a modular and maintainable way.
6. Testing, logging, and debugging: Writing tests, logging events, and debugging are essential to maintaining high-quality code.
7. Dependencies management: Learn to manage dependencies and versions using tools like pip-tools. This is essential for maintaining consistent environments.
8. Concurrency and parallelism: Explore asyncio, multithreading, and multiprocessing to handle tasks efficiently and boost performance.
9. Design patterns: Implement design patterns to create modular, scalable, and maintainable code that aligns with best practices.

3.2. Compulsory Resources

3.2.1. The Complete Python Developer

This comprehensive, project-based course will introduce you to all of the modern skills of a Python developer (Python 3). Along the way, we will build over 12 real-world projects to add to your portfolio. (You will get access to all the code from the 12+ projects we build so that you can put them on your portfolio right away.)

The curriculum will be very hands-on as we walk you from start to finish to become a professional Python developer. We will start from the very beginning by teaching you Python basics and programming fundamentals, and then go into advanced topics and different career fields in Python so you can get real-life practice and be ready for the real world.

The topics covered in this course are:

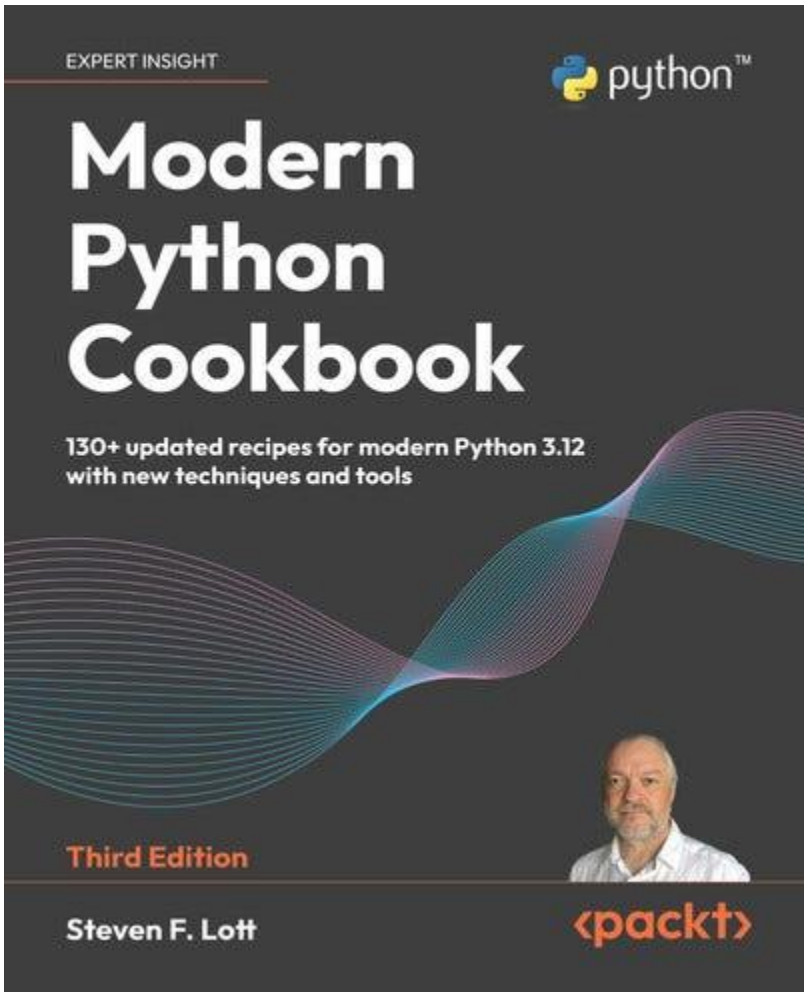
- Programming Fundamentals
- Python Basics
- Python Fundamentals
- Data Structures
- Object Oriented Programming with Python
- Functional Programming with Python
- Lambdas
- Decorators
- Generators
- Testing in Python
- Debugging
- Error Handling
- Regular Expressions
- Comprehensions
- Modules
- Virtual Environments
- Developer Environments (PyCharm, Jupyter Notebooks, VS Code, Sublime Text + more)
- File Processing: Image, CSV, PDFs, Text + more
- Web Development with Python
- Machine Learning with Python
- Data Science with Python
- Automation with Python and Selenium
- Scripting with Python
- Web Scraping with Python and BeautifulSoup
- Image Detection
- Data Visualizations
- Kaggle, Pandas, NumPy, scikit-learn
- Email and SMS with Python
- Working with APIs (Twitter Bot, Password Checker, Translator)

Resource Information:

- Course Link
- Course Provider: Udemy
- Duration: 1 month (At 20 hours/week)
- Instructor: Andrei Neagoie

3.3. Optional Resources

3.3.1. Modern Python Cookbook — Third Edition



The third edition of [Modern Python Cookbook] provides an in-depth look into Python 3.12, offering more than 140 new and updated recipes that cater to both beginners and experienced developers.

This edition introduces new chapters on documentation and style, data visualization with Matplotlib and Pyplot, and advanced dependency management techniques using tools like Poetry and Anaconda.

With practical examples and detailed explanations, this cookbook helps developers solve real-world problems, optimize their code, and get up to date with the latest Python features. What you will learn:

- Master core Python data structures, algorithms, and design patterns
- Implement object-oriented designs and functional programming features
- Use type matching and annotations to make more expressive programs

- Create useful data visualizations with Matplotlib and Pyplot
- Manage project dependencies and virtual environments effectively
- Follow best practices for code style and testing
- Create clear and trustworthy documentation for your projects

3.4. Putting it into Action

The last step in this learning step is to put everything you have learned so far into action. At this point, you are ready to build a functioning system in Python and add it to your GitHub profile and your portfolio.

A potential idea is to build a Task Management System that allows users to manage their to-do lists, set reminders, and analyze their productivity. Include advanced features like a web interface, database integration, API, and parallel task execution.
Here is how you will put the learning outcomes into action:

1. Data Structures:
   - Use lists to store tasks, dictionaries for user data, and sets to manage unique task tags.
   - Example: A dictionary where keys are dates and values are lists of tasks.

2. Functions:
   - Define reusable functions for adding, editing, and deleting tasks.
   - Implement recursion for nested task categories (e.g., subtasks).

3. Control Flow:
   - Use conditional statements for user actions like marking tasks as complete or overdue.
   - Implement loops to display tasks and handle repetitive operations.

4. Error Handling:
   - Gracefully handle errors like invalid input or database connection failures using try and except.
   - Ensure logging for failed API calls or tasks not saved.

5. Object-Oriented Programming (OOP):
   - Create classes for tasks, users, and reminders. Use inheritance to differentiate between task types (e.g., personal, work).
   - Encapsulate logic for reminders and task analytics within objects.

6. Testing, Logging, and Debugging:
   - Write unit tests for core functionalities (e.g., task addition).
   - Implement logging to record user actions and system errors.
   - Debug with tools like pdb or IDE-integrated debuggers.

7. Dependencies Management:

- Use pip-tools or a requirements.txt file to manage dependencies (e.g., Flask/Django for the web interface, SQLAlchemy for the database).

8. Concurrency and Parallelism:
    - Use asyncio to handle reminders and notifications in the background.
    - Explore multithreading to manage multiple user sessions.

9. Design Patterns:
    - Apply patterns like MVC (Model-View-Controller) to separate logic, data, and user interaction.
    - Use a Singleton pattern to manage the database connection.

After finishing the project, it is always a good idea to publish it on different social media channels and your GitHub profile. You can also write a step-by-step guide on Medium or through a video on YouTube.

4. Python for Data Science

In the Fourth article of this series, you will learn the fundamentals of Python for Data Science to efficiently handle the entire data lifecycle, from preprocessing and visualization to advanced machine learning and deep learning tasks.

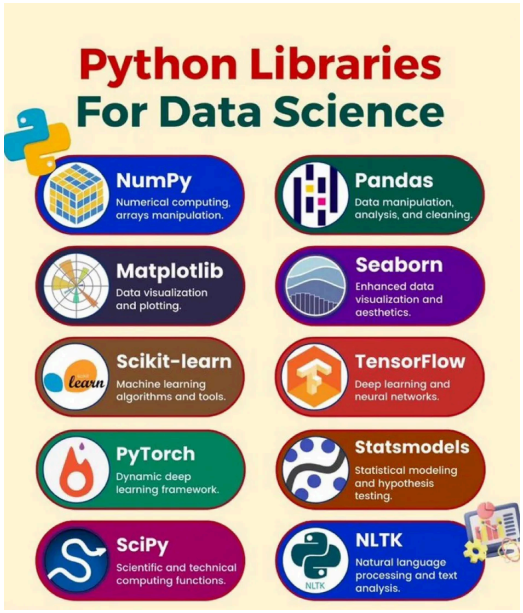
These fundamentals include learning how to use key libraries like NumPy and Pandas to simplify numerical computations and data manipulation, while Matplotlib and Seaborn enable insightful data visualizations. For machine learning, scikit-learn provides a robust framework to build, train, and evaluate models, making it ideal for predictive analytics.

4.1. Python for Data Science

Python is a cornerstone of modern data science, renowned for its simplicity, versatility, and extensive library ecosystem. It empowers data scientists to efficiently handle the entire data lifecycle, from preprocessing and visualization to advanced machine learning and deep learning tasks.

Key libraries like NumPy and Pandas simplify numerical computations and data manipulation, while Matplotlib and Seaborn enable insightful data visualizations. For machine learning, scikit-learn provides a robust framework to build, train, and evaluate models, making it ideal for predictive analytics.

As data grows in complexity and scale, tools like Dask and PySpark allow Python to process massive datasets seamlessly. With its accessibility and powerful capabilities, Python continues to be an indispensable tool for extracting meaningful insights from data.



4.2. Compulsory Resources

4.2.1. Python for Data Science and Machine Learning Bootcamp

This comprehensive course will be your guide to learning how to use the power of Python to analyze data, create beautiful visualizations, and use powerful machine-learning algorithms! With over 100 HD video lectures and detailed code notebooks for every lecture, this is one of the most comprehensive courses for data science and machine learning on Udemy!

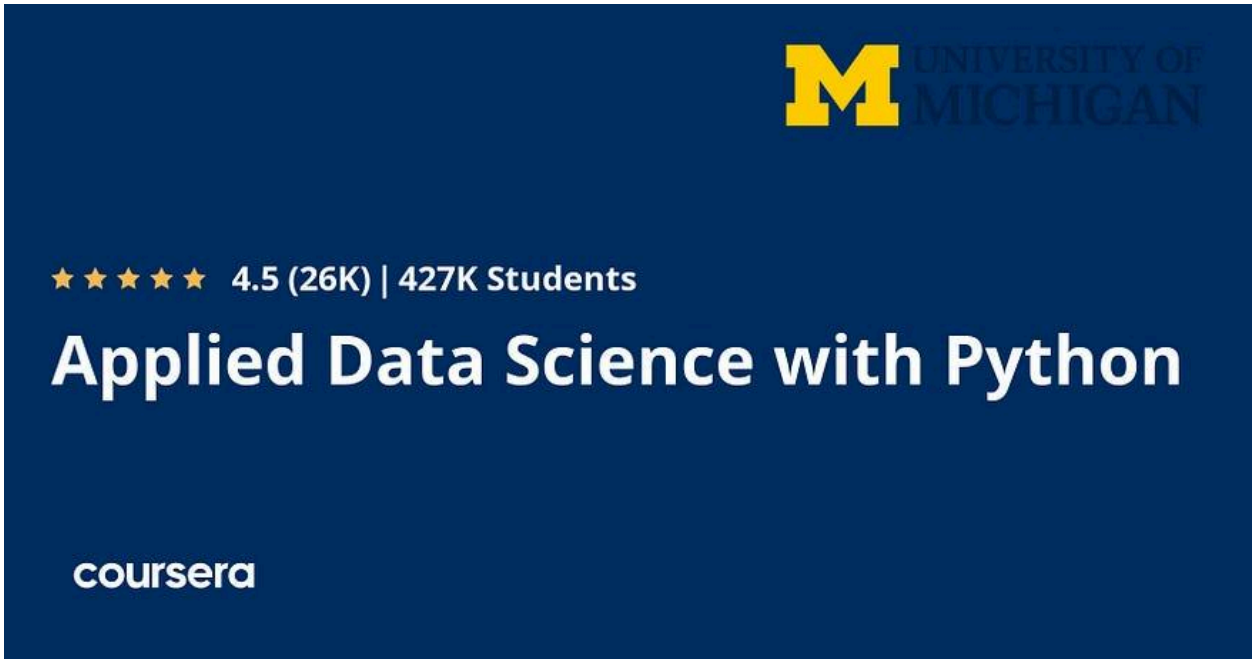
In this course, you will learn how to program with Python, create amazing data visualizations, and use Machine Learning with Python! Here are just a few of the topics we will be learning:

- Programming with Python
- NumPy with Python
- Using pandas Data Frames to solve complex tasks
- Use pandas to handle Excel Files
- Web scraping with Python
- Connect Python to SQL
- Use matplotlib and Seaborn for data visualizations
- Use plotly for interactive visualizations
- Machine Learning with SciKit Learn, including:
- Linear Regression
- K Nearest Neighbors
- K Means Clustering
- Decision Trees
- Random Forests
- Natural Language Processing
- Neural Nets and Deep Learning
- Support Vector Machines

Course Information:

- Course Link
- Cost: 119 $
- Course Provider: Udemy
- Duration: 1 month (At 20 hours/week)
- Instructor: Jose Portilla

4.2.2. Applied Data Science with Python Specialization



The 5 courses in this University of Michigan specialization introduce learners to data science through the Python programming language. This skills-based specialization is intended for learners who have a basic Python or programming background and want to apply statistical, machine learning, information visualization, text analysis, and social network analysis techniques through popular Python toolkits such as pandas, matplotlib, scikit-learn, nltk, and networks to gain insight into their data.

- Introduction to Data Science in Python (course 1)
- Applied Plotting, Charting & Data Representation in Python (course 2)
- Applied Machine Learning in Python (course 3)
- Applied Text Mining in Python (course 4)
- Applied Social Network Analysis in Python (course 5)

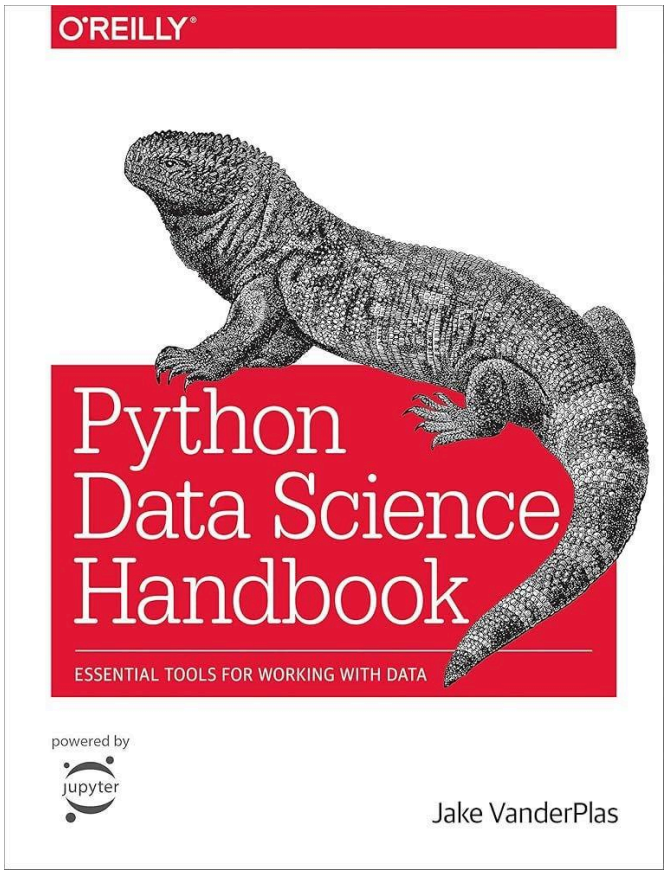
The first three courses should be taken in order and prior to any other course in the specialization. After completing those, courses 4 and 5 can be taken in any order. All 5 are required to earn a certificate.

Course Information:

- Course Link
- Cost: 40$/Course (Financial aid available)
- Course Provider: University of Michigan
- Duration: 1.5 months (At 20 hours/week)
- Instructor: Christopher Brooks

4.3. Additional Resources

4.3.1. Python Data Science Handbook



The final resource on our list is the Python Data Science Handbook which is optional if you want to increase your knowledge on this topic. For many researchers, Python is a first-class tool mainly because of its libraries for storing, manipulating, and gaining insight from data.

Several resources exist for individual pieces of this data science stack, but only with the Python Data Science Handbook do you get them all — IPython, NumPy, Pandas, Matplotlib, Scikit-Learn, and other related tools.

Working scientists and data crunchers familiar with reading and writing Python code will find this comprehensive desk reference ideal for tackling day-to-day issues: manipulating, transforming, and cleaning data; visualizing different types of data; and using data to build statistical or machine learning models. Quite simply, this is the must-have reference for scientific computing in Python. With this handbook, you'll learn how to use:

- IPython and Jupyter: provide computational environments for data scientists using Python
- NumPy: includes the ndarray for efficient storage and manipulation of dense data arrays in Python
- Pandas: features the DataFrame for efficient storage and manipulation of labeled/columnar data in Python
- Matplotlib: includes capabilities for a flexible range of data visualizations in Python
- Scikit-Learn: for efficient and clean Python implementations of the most important and established machine learning algorithms

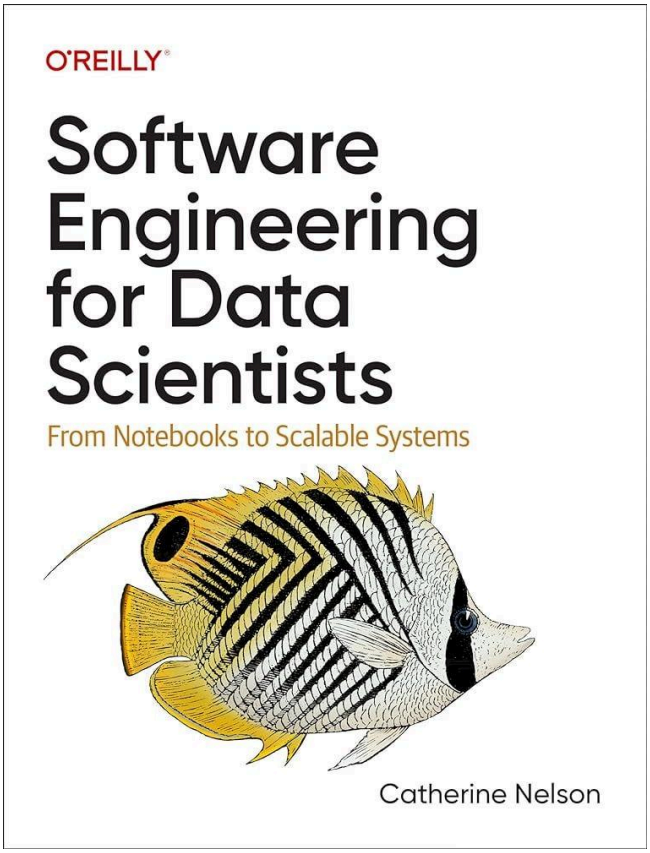
5. Software Engineering Basics for Data Science

In the fifth chapter of this book, you will learn the fundamentals of **Software Engineering Basics for Data Science**. The resources cover software engineering and clearly explain how to apply the best practices from software engineering to data science, Version Control, Infrastructure & Resource Management, and Linux Basics & Shell Scripting.

Data scientists are software engineers first and foremost. They may not be coding machine learning models or natural language processing algorithms on a day-to-day basis, but their work as data scientists requires software engineering and programming skills to apply all the data science project life cycles on the data.

In addition to that, they should be able to understand users' needs and develop solutions for those needs, which is essential for any data scientist working in an organization.

5.1. Software Engineering for Data Scientists Book



Data science happens in code. The ability to write reproducible, robust, scaleable code is key to a data science project's success — and is essential for those working with production code. This practical book Software Engineering for Data Scientists is one of the very few books that bridges the gap between data science and software engineering and clearly explains how to apply the best practices from software engineering to data science.

Examples are provided in Python, drawn from popular packages such as NumPy and pandas. If you want to write better data science code, this guide covers the essential topics that are often missing from introductory data science or coding classes, including how to:

- Understand data structures and object-oriented programming
- Clearly and skillfully document your code
- Package and share your code
- Integrate data science code with a larger code base
- Learn how to write APIs
- Create secure code
- Apply best practices to common tasks such as testing, error handling, and logging
- Work more effectively with software engineers
- Write more efficient, maintainable, and robust code in Python
- Put your data science projects into production

The book covers the following topics:

1. What Is Good Code?
2. Analyzing Code Performance
3. Using Data Structures Effectively
4. Object-Oriented Programming and Functional Programming
5. Errors, Logging, and Debugging
6. Code Formatting, Linting, and Type Checking
7. Testing Your Code
8. Design and Refactoring
9. Documentation
10. Sharing Your Code: Version Control, Dependencies, and Packaging
11. APIs
12. Automation and Deployment
13. Security
14. Working in Software

5.2. Version Control

The second step in this learning roadmap is to master version control/ and how it is used in machine learning. This includes understanding tools such as Git and how to use them to track changes to your code and models.

Working in production demands data scientists and machine learning engineers to know version control. Since you will be working in cooperation with other data scientists, data engineers, and software engineers, therefore it is important to be able to share your code and update it and also to follow up on their updates in a professional way.

Learning Resources:

- Git and GitHub Tutorial For Beginners | Full Course

5.3. Infrastructure & Resource Management

The third step is to learn about infrastructure and resource management tools for machine learning. This includes understanding how to provision and manage computing resources for training and deploying machine learning models and how to scale machine learning pipelines. Examples of infrastructure and resource management tools include:

1. **Kubernetes**: This open-source system allows you to automate the deployment, scaling, and management of containerized applications. It can be particularly useful for managing machine learning workflows, as it allows you to easily scale up or down as needed.
2. **Docker**: It is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow you to package an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package. This makes it easier to run the application on any other machine because everything it needs is contained in the package. Docker is often used in conjunction with container orchestration tools like Kubernetes to manage the deployment and scaling of containerized applications. It is also commonly used for developing and testing machine learning applications, as it allows you to create isolated environments with specific dependencies and packages.
3. **AWS & Amazon SageMaker**: This fully managed service from Amazon Web Services (AWS) simplifies the process of building, training, and deploying machine learning models. It includes tools for resource management, such as the ability to select the right hardware and automatically scale up or down as needed.

Overall, there are many tools available to help with infrastructure and resource management for machine learning, and the right choice for you will depend on your specific needs and preferences.
Learning Resources:

- Kubernetes: Kubernetes Tutorial for Beginners [FULL COURSE in 4 Hours]
- Docker: Docker Tutorial for Beginners [FULL COURSE in 3 Hours]
- AWS Zero to Hero — AWS Simplified
- AWS SageMaker: Amazon (AWS) Sagemaker Full Course | Getting Started

5.4. Linux Basics & Shell Scripting

Linux basics and scripting are essential for data scientists, as Linux powers most servers, cloud platforms, and tools like Hadoop and Docker. Mastering commands for file management, text processing, and networking helps efficiently handle datasets, automate workflows, and configure environments.

Scripting, particularly with Bash, enables automation of repetitive tasks, integration of tools, and reproducibility of data workflows. Tasks like downloading datasets, preprocessing, and scheduling jobs with cron become seamless with scripting. These skills enhance productivity,

ensure reproducibility, and are critical for deploying scalable machine-learning projects in production.
If you're new to Linux, this beginner's course is for you. You'll learn many of the tools used every day by both Linux SysAdmins and the millions of people running Linux distributions like Ubuntu on their PCs. In this learning step, you will teach you how to navigate Linux's Graphical User Interfaces and powerful command line tool ecosystem.

Learning Resources:

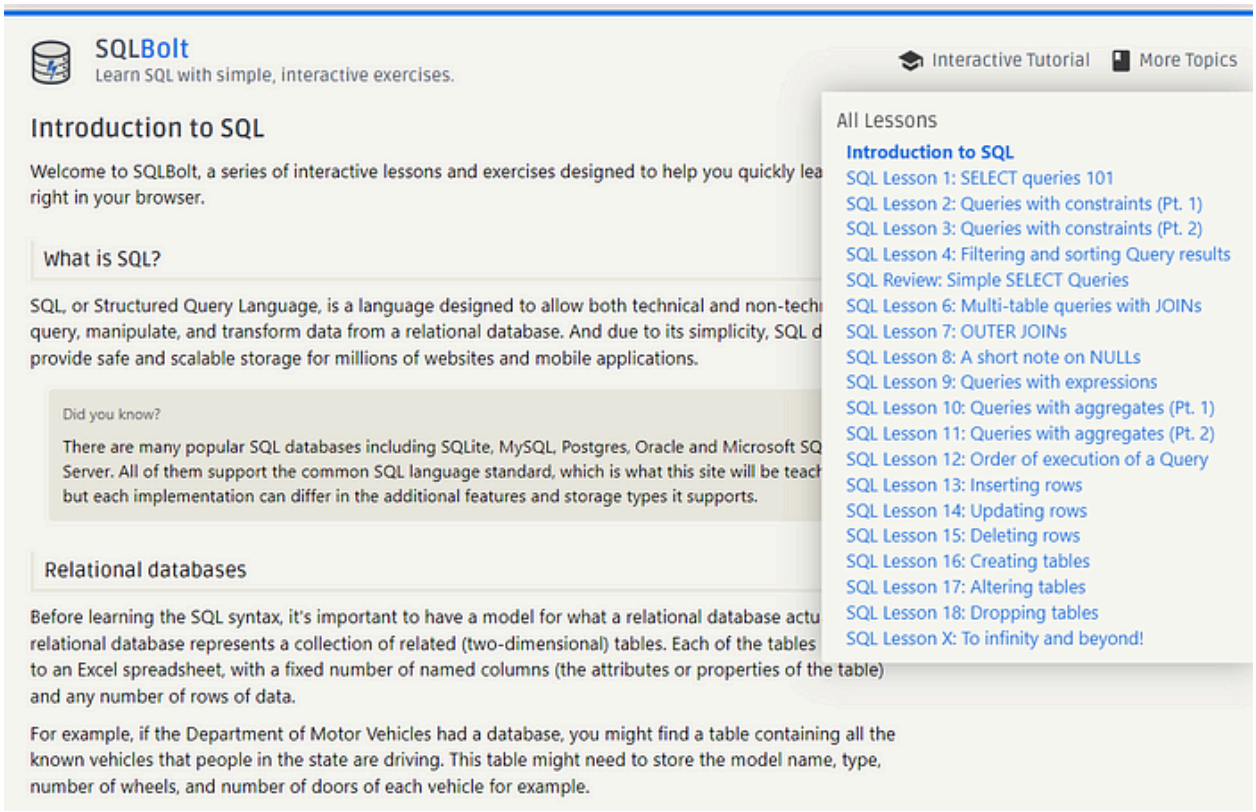
- Introduction to Linux — Full Course for Beginners

6. Database & SQL Fundamentals

In today's data-driven world, the ability to analyze and manipulate large amounts of data has become a critical skill for data scientists. SQL (Structured Query Language) is a powerful tool that allows data professionals to extract valuable insights from vast amounts of data. However, mastering SQL can be a challenging task, especially for those new to the field. This comprehensive guide aims to bridge the gap between novice and advanced SQL skills for data scientists.

From the basics of SQL syntax to advanced techniques such as optimizing queries and data modeling, this guide provides a step-by-step approach to mastering SQL for data science. Whether you are a beginner or an experienced data professional, this guide will equip you with the knowledge and skills needed to become a SQL master and excel in your data science career.

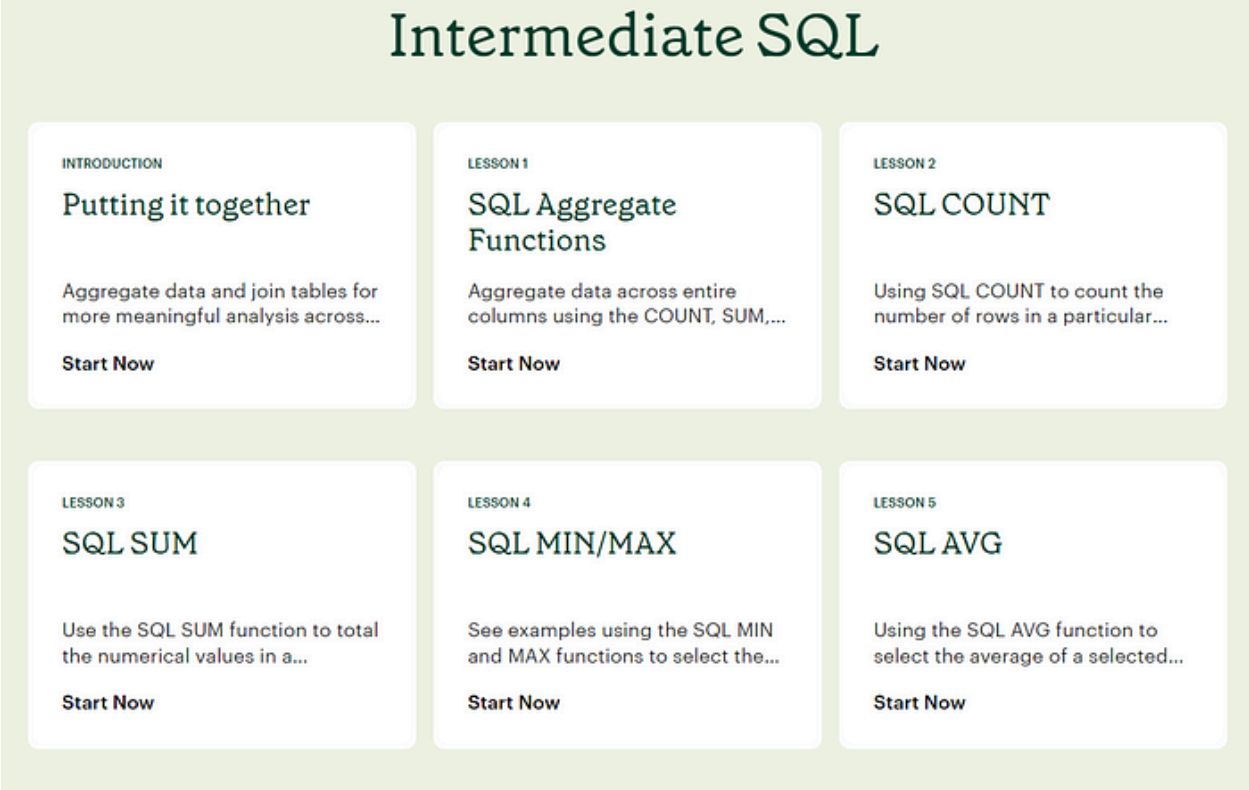
6.1. SQL Basics



First, you will start by learning the SQL basics commands which include SELECT, WHERE, JOINS, Aggregate Functions (Count, Sum, AVG, etc.), and table commands such as (create, delete, insert, etc.) There are many resources to learn these basics but I recommend SQL Bolt.

The lessons are designed to be interactive and hands-on, allowing users to practice writing and executing SQL queries in a real-world setting. This makes SQL Bolt a great resource for anyone looking to learn SQL, whether you're a beginner or an experienced developer looking to brush up on your skills.
It has 19 lessons I believe you can finish all of them in one day or two and by that, you will have the SQL basic knowledge and it will now be time for learning intermediate topics.

6.2. Intermediate & Advanced SQL



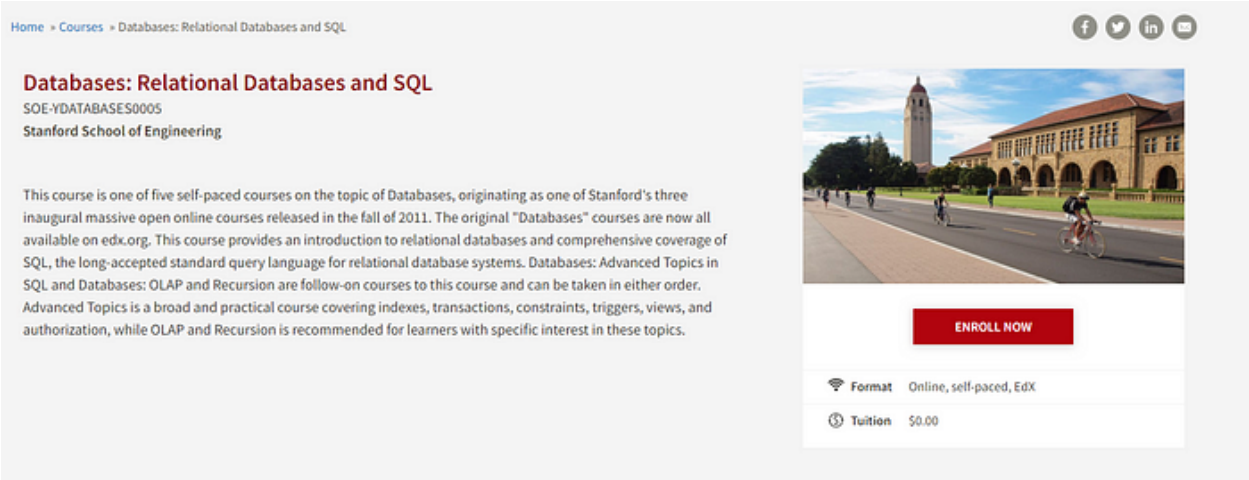
Now it is time to learn more advanced SQL concepts such as Subqueries, Window functions, SQL for data wrangling, and more. A great place to learn these concepts is Mode SQL. They have 4 levels of SQL lessons:

- Basics (14 lessons)
- Intermediate (20 Lesson)
- Advanced (8 Lessons)
- SQL Analytics Training (7 Lessons)

Since the basics were covered before you can skip them or you can take them again if you would like to revise the basics and have a deeper understanding of them. However, you have to take the three remained levels to cover most of the important SQL commands and concepts.

After finishing the basics and the advanced level, it will be good to have a basic understanding of databases. Although as a data scientist, you might not have to deal with it directly it will give you a new perspective and understanding that will help you build better solutions.

6.3. Database for Data Science



After studying SQL commands and statements it will be important to have a basic understanding of databases and their main characteristics and concepts. My recommendation is the Stanford Databases: Relational Databases and SQL Course. This course is one of five self-paced courses on the topic of Databases, originating as one of Stanford's three inaugural massive open online courses released in the fall of 2011.

This course provides an introduction to relational databases and comprehensive coverage of SQL, the long-accepted standard query language for relational database systems. Databases: Advanced Topics in SQL and Databases: OLAP and Recursion are follow-on courses to this course and can be taken in either order.

Advanced Topics is a broad and practical course covering indexes, transactions, constraints, triggers, views, and authorization, while OLAP and Recursion are recommended for learners with a specific interest in these topics.

6.4. SQL Case Studies

Now you are ready for practicing. I would recommend starting with practicing real case studies. A great place to do so is the **8-week SQL challenge by Danny Ma**.

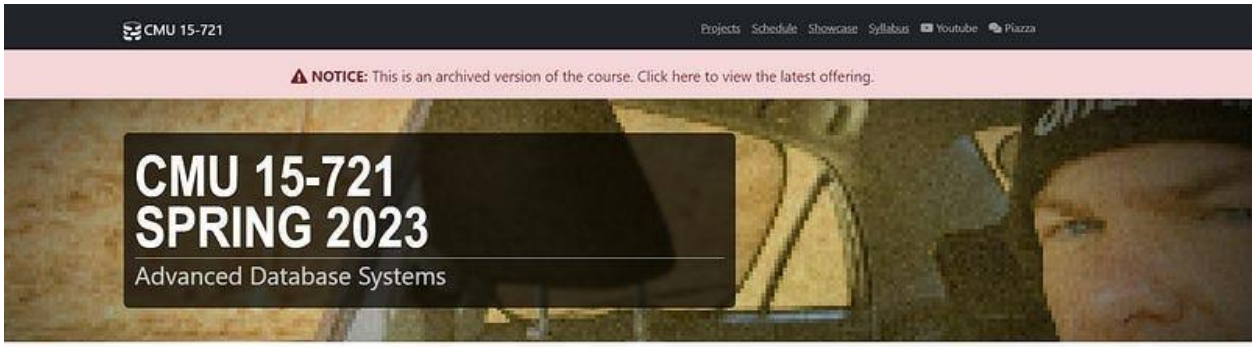
6.5. SQL Practice

Finally, you can start to solve SQL questions that you will expect to meet in a data science interview and also in practice. There are a lot of resources to do so. Here are a few good options:

- **Leetcode**
- **Stratascratch**
- **DataLemur**

6.6. Advanced SQL Learning Material

6.6.1. Advanced Database Systems—CMU



This Advanced Database Systems course is a comprehensive study of the internals of modern database management systems. It will cover the core concepts and fundamentals of the components that are used in large-scale analytical systems (OLAP).

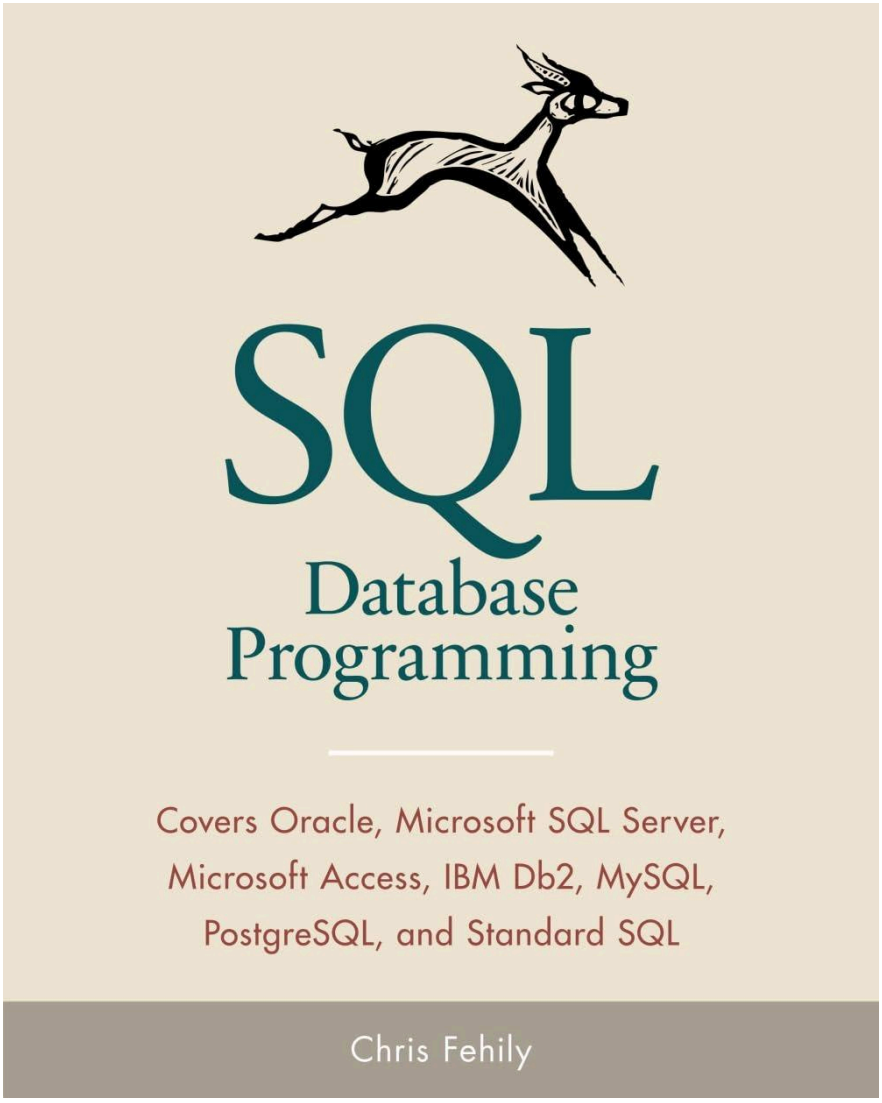
The class will stress both the efficiency and correctness of the implementation of these ideas. The course is appropriate for graduate students in software systems and for advanced undergraduates with dirty systems programming skills.
Upon successful completion of this course, you should be able to:

- Apply and customize state-of-the-art implementation techniques for single-node database management systems following modern coding practices.
- Identify trade-offs among database systems techniques and contrast alternatives for both online transaction processing and online analytical workloads.
- Develop and justify design decisions in the context of a high-performance database system.
- Implement and evaluate complex, scalable database systems, with an emphasis on providing experimental evidence for design decisions.
- Interpret and comparatively criticize state-of-the-art research talks and papers, with emphasis on constructive improvements.

6.6.2. SQL Database Programming

The SQL Database Programming book teaches newcomers SQL, the language of databases, and includes examples of the most widely used database systems.

In all its editions, this book has sold more than 150,000 copies and is popular with end users, students, data scientists, statisticians, epidemiologists, analysts, app developers, webmasters, and hobbyists. Thorough cross-referencing makes it a useful desktop reference for experienced SQL programmers



. The book covers the following topics:

- Covers Oracle Database, Microsoft SQL Server, Microsoft Access, IBM Db2 Database, MySQL, PostgreSQL, and Standard SQL.
- Hundreds of examples of varied difficulty encourage you to experiment and explore.
- Download the sample database and SQL source code to follow along with the examples.
- Organize your database in terms of the relational model.
- Master tables, columns, rows, and keys.
- Retrieve, filter, sort, and format data.
- Use functions and operators to transform and summarize data.
- Create, alter, and drop database tables.
- Answer hard questions by using joins, subqueries, constraints, conditional logic, and metadata.
- Create indexes that speed sorts and searches.
- Use views to secure and simplify data access.

- Insert, update, delete, and merge data.
- Execute transactions to maintain the integrity of your data.
- Avoid common pitfalls involving nulls.
- Troubleshoot and optimize queries.
- Learn advanced techniques that extend the power of SQL.

Table of Contents:

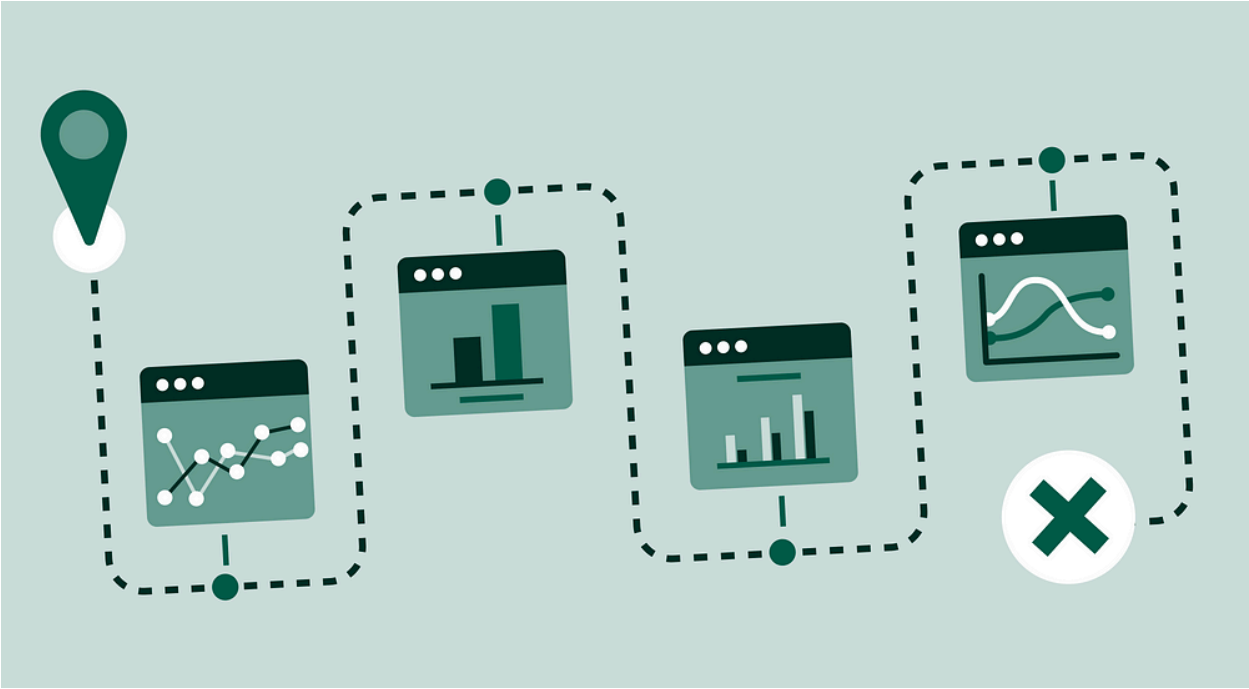
1. Running SQL Programs
2. The Relational Model
3. SQL Basics
4. Retrieving Data from a Table
5. Operators and Functions
6. Summarizing and Grouping Data
7. Joins
8. Subqueries
9. Set Operations
10. Inserting, Updating, and Deleting Rows
11. Creating, Altering, and Dropping Tables
12. Indexes
13. Views
14. Transactions
15. Advanced SQL

7. Data Cleaning & Preprocessing

In the Seventh chapter of this book, you will learn the fundamentals of Data Cleaning & Preprocessing for data science. Data cleaning and preprocessing are one of the most important parts of a data scientist's day. It's something you'll do daily. Being able to clean your data effectively effectively will result in better results with less effort.

The more you know, the better you will understand the data, which will help you to produce better results and be effective at work. There are many courses and books on this topic that you can read to expand your knowledge and skills. I went through most of them and selected the most important ones that will build your fundamentals.

7.1. Data Exploration



So you've got some interesting data—where do you begin your analysis? This learning step will cover the process of exploring and analyzing data, from understanding what's included in a dataset to incorporating exploration findings into a data science workflow.

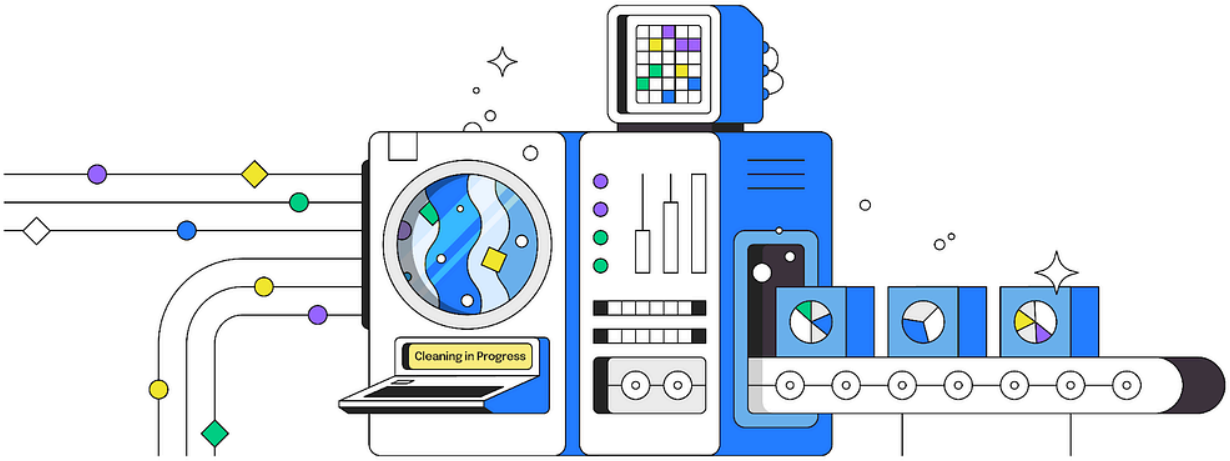
By the end of this learning step, you'll have the confidence to perform your own exploratory data analysis (EDA) in Python. You'll be able to explain your findings visually to others and suggest the next steps for gathering insights from your data!

Learning Resources:

- **Introduction to Statistics in Python** | Expected Duration (2 days) | Datacamp

- **Introduction to Data Visualization with Seaborn** | Expected Duration (2 days) | Datacamp
- **Exploratory Data Analysis in Python** Expected Duration (2 days) | Datacamp

7.2. Data Cleaning & Data Preprocessing



Data cleaning is a key part of data science, but it can be deeply frustrating. Why are some of your text fields garbled? What should you do about those missing values? Why aren't your dates formatted correctly? How can you quickly clean up inconsistent data entry? In this course, you'll learn why you've run into these problems and, more importantly, how to fix them!

In this learning step, you'll learn how to tackle some of the most common data-cleaning problems so you can analyze your data faster. You'll work through five hands-on exercises with real, messy data and answer some of your most commonly-asked data cleaning questions

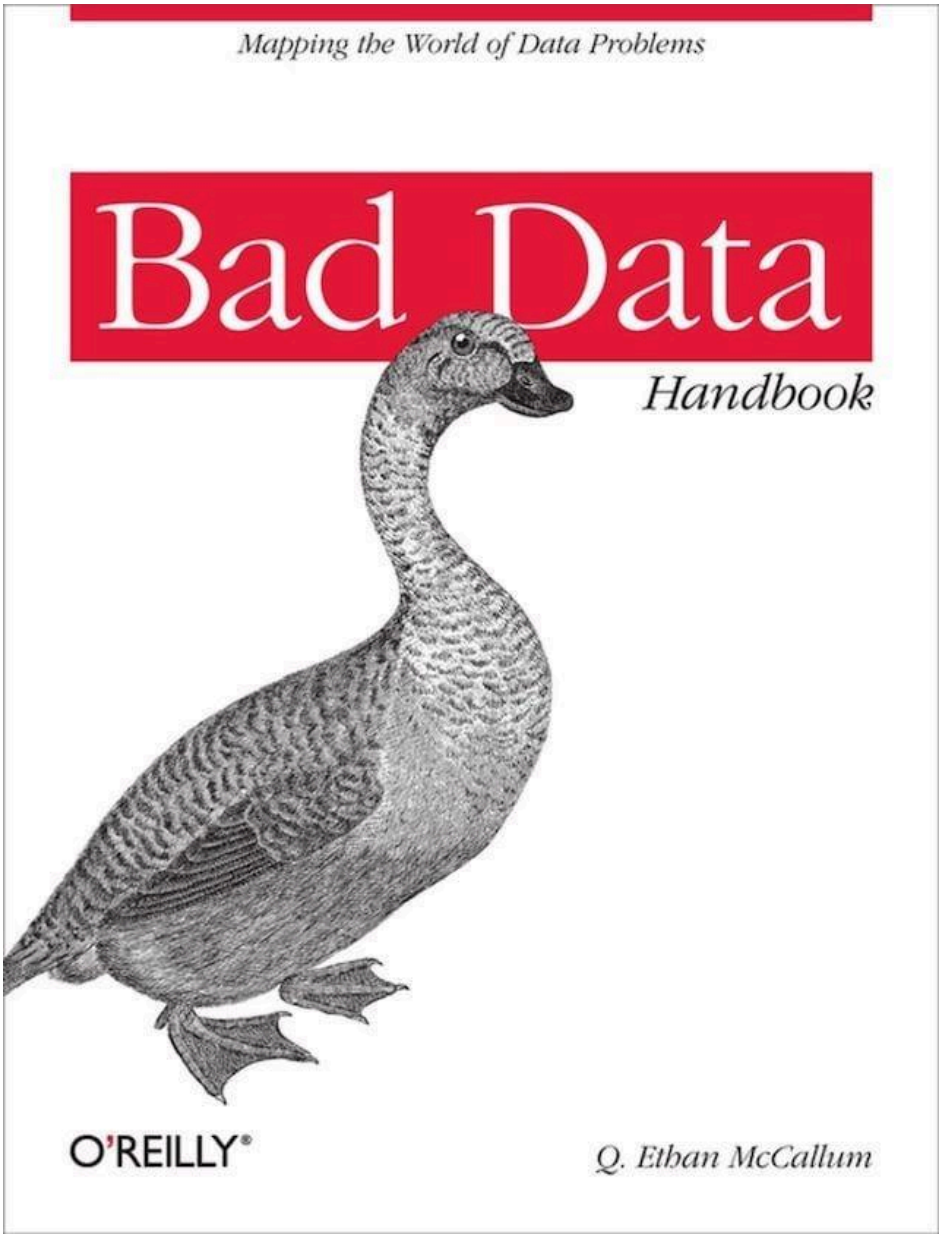
Learning Resources:

- **Cleaning Data in Python | | Expected Duration (2 days) | Datacamp**
- **Data cleaning | Expected Duration (2 days) | Kaggle**

7.3. Optional Learning Resources

7.3.1. Bad Data

The first book in the optional resource is Bad Data Handbook: Cleaning Up The Data So You Can Get Back To Work edited by Q. Ethan Mccallum. This book is a collection of essays by 19 machine learning practitioners and is full of useful nuggets on data preparation and management.



What is bad data? Some people consider it a technical phenomenon, like missing values or malformed records, but bad data includes a lot more. In this handbook, data expert Q. Ethan McCallum has gathered 19 colleagues from every corner of the data arena to reveal how they've recovered from nasty data problems.

Among the many topics covered, you'll discover how to:

- Test drive your data to see if it's ready for analysis
- Work spreadsheet data into a usable form
- Handle encoding problems that lurk in text data
- Develop a successful web-scraping effort
- Use NLP tools to reveal the real sentiment of online reviews

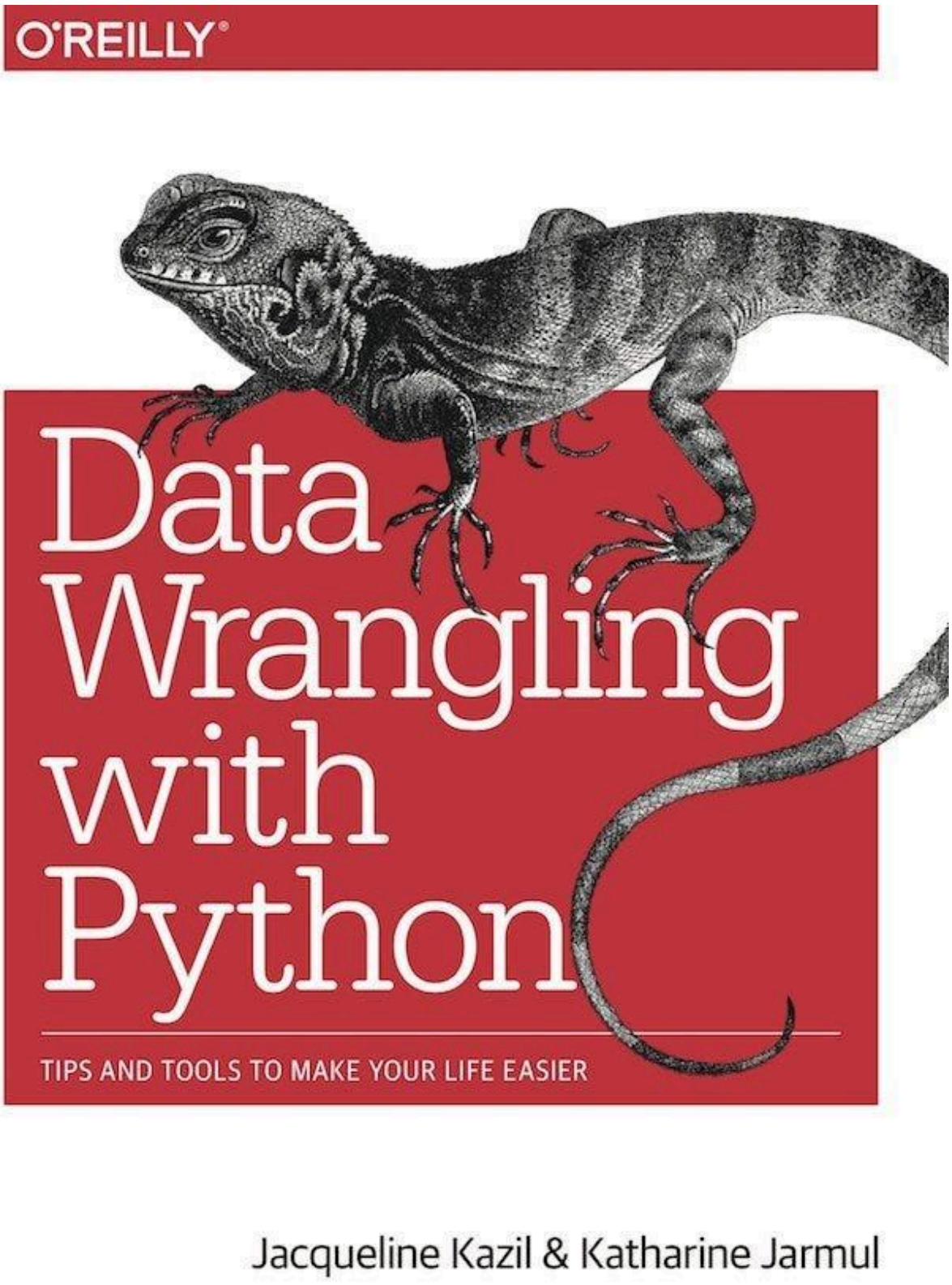
- Address cloud computing issues that can impact your analysis effort
- Avoid policies that create data analysis roadblocks
- Take a systematic approach to data quality analysis

Table of contents:

- **Chapter 01:** Setting the Pace: What Is Bad Data?
- **Chapter 02:** Is It Just Me, or Does This Data Smell Funny?
- **Chapter 03:** Data Intended for Human Consumption, Not Machine Consumption
- **Chapter 04:** Bad Data Lurking in Plain Text
- **Chapter 05:** (Re)Organizing the Web's Data
- **Chapter 06:** Detecting Liars and the Confused in Contradictory Online Reviews
- **Chapter 07:** Will the Bad Data Please Stand Up?
- **Chapter 08:** Blood, Sweat, and Urine
- **Chapter 09:** When Data and Reality Don't Match
- **Chapter 10:** Subtle Sources of Bias and Error
- **Chapter 11:** Don't Let the Perfect Be the Enemy of the Good: Is Bad Data Really Bad?
- **Chapter 12:** When Databases Attack: A Guide for When to Stick to Files
- **Chapter 13:** Crouching Table, Hidden Network
- **Chapter 14:** Myths of Cloud Computing
- **Chapter 15:** The Dark Side of Data Science
- **Chapter 16:** How to Feed and Care for Your Machine-Learning Expert
- **Chapter 17:** Data Traceability
- **Chapter 18:** Social Media: Erasable Ink?
- **Chapter 19:** Data Quality Analysis Demystified: Knowing When Your Data Is Good Enough

7.3.2. Data Wrangling with Python

The second book in the optional resource is Data Wrangling with Python: Tips and Tools to Make Your Life Easier written by Jacqueline Kazil and Katharine Jarmul. The focus of this book is the tools and methods to help you get raw data into a form ready for modeling.



Data wrangling is a more general or colloquial term for data preparation that might include some

data cleaning and feature engineering. In this book, you will learn more about data wrangling in a practical way.

Through various step-by-step exercises, you will learn how to acquire, clean, analyze, and present data efficiently. You will also discover how to automate your data process, schedule file editing and clean-up tasks, process larger datasets, and create compelling stories with the data you obtain. The book will teach you:

- Basic Python syntax, data types, and language concepts
- Work with both machine-readable and human-consumable data
- Scrape websites and APIs to find a bounty of useful information
- Clean and format data to eliminate duplicates and errors in your datasets
- Learn when to standardize data and when to test and script data cleanup
- Explore and analyze your datasets with new Python libraries and techniques

Table of contents:

- Chapter 01: Introduction to Python
- Chapter 02: Python Basics
- Chapter 03: Data Meant to Be Read by Machines
- Chapter 04: Working with Excel Files
- Chapter 05: PDFs and Problem-Solving in Python
- Chapter 06: Acquiring and Storing Data
- Chapter 07: Data Cleanup: Investigation, Matching, and Formatting
- Chapter 08: Data Cleanup: Standardizing and Scripting
- Chapter 09: Data Exploration and Analysis
- Chapter 10: Presenting Your Data
- Chapter 11: Web Scraping: Acquiring and Storing Data from the Web
- Chapter 12: Advanced Web Scraping: Screen Scrapers and Spiders
- Chapter 13: APIs
- Chapter 14: Automation and Scaling
- Chapter 15: Conclusion

7.4. Putting it into Action

The last step is to put what you have learned into action. Although most of the resources have guided exercises and projects. However, it is important to go into the wild and apply the learned techniques to a real dataset.

The best and most direct approach is to go to kaggle to check the **knowledge & get started with competitions,** then choose two or three competitions that you are interested in and start working with the data to explore, clean and preprocess. Also, it will be very helpful to check the most upvoted notebooks in this competition as you will go through very high-level data cleaning and preprocessing pipelines.

8. Feature Engineering

In the eighth chapter of this book, you will learn the fundamentals of Data Cleaning & Preprocessing for data science. Feature engineering is one of the most important parts of a data scientist's day. It's something you'll do regularly and it's an essential step before training any machine learning model.

Being able to clean your data effectively and engineering the features effectively will result in better results with less effort and computational power. I believe that the more you know, the better you will understand the data, which will help you to produce better results and be effective at work.

In this chapter, I'll share essential courses to help you master feature engineering. I'll also include additional resources for further learning, should you want to dive deeper into the topic.

8.1. Feature Engineering For Machine Learning

The first course is **Feature Engineering For Machine Learning by Train in Data**. In this course, you will learn missing data imputation, encoding of categorical features, numerical variable transformation and discretization, feature extraction, and more.

Feature engineering is the process of using domain knowledge and statistical methods to create features that make machine learning algorithms work effectively.

Feature engineering is key in applied machine learning. Raw data is almost never suitable for training machine learning models. In fact, data scientists devote a lot of effort to data analysis, data preprocessing, and feature extraction, to create better features to train predictive models.

While most online courses will teach you the basics of feature engineering, like imputing variables with the mean or transforming categorical features using one-hot encoding, this course will teach you all of that and much more.

You will first learn the most popular techniques for variable engineering, like mean and median imputation, one-hot encoding, transformation with logarithm, and discretization.

Then, you will discover more advanced methods that capture information while encoding or transforming your variables, to obtain better features and improve the performance of regression and classification models.

You will learn methods described in scientific articles, used in data science competitions like those hosted by Kaggle and the KDD, and commonly utilized in organizations. What's more, they can be easily implemented by utilizing Python's open-source libraries.

Course Topics:

- How to impute missing values
- How to encode categorical features
- How to transform and scale numerical variables
- How to perform discretization
- How to remove outliers
- How to perform feature extraction from date and time
- How to create new features from existing ones

Course Information:

- Difficulty Level: Beginner
- Duration: 12 hours
- Price: 24 euros
- Instructor: Soledad Galli, PhD

8.2. Machine Learning with Imbalanced Data

The second course is **Machine Learning with Imbalanced Data from Train in Data.** In this course, you will learn to over-sample and under-sample your data and apply SMOTE, ensemble methods, and cost-sensitive learning.

Imbalanced datasets are those typically used in classification problems where one of the target classes is extremely under-represented. When this happens, we talk about a class imbalance. The class with a small number of samples is called the minority class, and the class or classes with plenty of data are called the majority class or classes.
Imbalanced datasets are a common occurrence in data science. Examples of imbalanced datasets are those used for fraud detection or medical diagnosis.

In this course, you will learn multiple methods to improve the performance of machine learning models trained on imbalanced data and decrease the misclassification of the minority class or classes.

Course Topics:

- **Evaluation metrics:** You will learn suitable metrics to assess imbalanced classification models trained with imbalanced datasets. You will learn about the roc-curve and the roc-AUC. You will create a confusion matrix, find true positives, true negatives, false positives, and false negatives, and then use them to calculate other metrics like precision, recall, and the f1-score. You will also learn about specific performance metrics to assess imbalanced classification models, like imbalanced accuracy, among others. Some of these metrics are geared toward binary classification problems. Other metrics can handle multi-class targets out of the box. You will learn when you can use each metric and why in your classification tasks.
- **Resampling techniques:** Next, you will learn about resampling methods, including under-sampling and over-sampling.
  Among the under-sampling methods, you will learn random under-sampling and cleaning methods based on k-nearest neighbors, like tomek links and nearmiss. Among the over-sampling techniques, you will learn random over-sampling and methods that create new data points, like the synthetic minority over-sampling technique (SMOTE) and its variations. SMOTE creates synthetic data, that is, new data, and therefore avoids the mere duplication of samples introduced by random over-sampling. Resampling methods are usually classified as data preprocessing methods because they change the distribution of the training dataset. In particular, the aim of resampling techniques is to create balanced datasets with a similar distribution across the different classes. You will learn how to correctly set up the resampling strategy, modify the training dataset, and leave a test set untouched with the original class distribution, to correctly perform the model validation in a similar setting to how it will be used in the real world.
- **Cost-sensitive learning:** Next, you will learn how to introduce class weights to perform cost-sensitive learning. Cost-sensitive learning uses the original dataset to train the models, without changing the class distribution. It aims to compensate for the misclassification of the minority class by penalizing harder the mistakes the classifier makes when classifying these observations.
- **Ensemble methods:** Finally, we will carry out specific bagging and boosting algorithms designed to handle imbalanced data.
  By the end of the course, you will be able to decide which technique is suitable for your dataset, and/or apply and compare the boost in performance returned by the different methods on multiple datasets

Course Information:

- **Difficulty Level**: Intermediate
- **Duration**: 15 hours
- **Price**: 24 euros
- **Instructor**: Soledad Galli, PhD

8.3. Feature Selection for Machine Learning

The third course on this list is **Feature Selection for Machine Learning from Train in Data**. In this course, you will learn filter, wrapper, and embedded methods, recursive feature elimination, exhaustive search, feature shuffling & more.

Feature selection is the process of identifying and selecting a subset of features from the original data set to use as inputs in a machine learning algorithm.

Data sets usually contain a large number of features. We can use multiple algorithms to quickly disregard irrelevant features and identify those important features in our data.

Feature selection algorithms can be divided into 1 of 3 categories: filter methods, wrapper methods, and embedded methods.

Filter methods comprise basic data preprocessing steps to remove constant and duplicated features and statistical tests to assert feature importance. Wrapper methods wrap the search around the estimator. They use backward and forward selection to examine and identify the best set of features. Embedded methods combine feature selection with the fitting of the classifier or regression model.

In this course, you will learn multiple feature selection techniques, gathered from scientific articles, data science competitions, and my experience as a data scientist, to identify relevant features in your data sets.

Course Topics:

1. Filter methods:

- Chi-square test for categorical variables
- ANOVA for continuous variables and binary or multiclass target variables
- Pearson's correlation for continuous variables in regression
- Information gain
- Mutual information

2. Wrapper methods:

- Forward selection of features
- Backward selection of variables
- Exhaustive search

3. Embedded methods:

- Lasso regularization
- Linear models coefficients
- Feature importance derived from decision trees and random forests

4. Hybrid methods:

- Recursive feature elimination or addition
- How to select features based on changes in model performance after feature shuffling

Course Information:

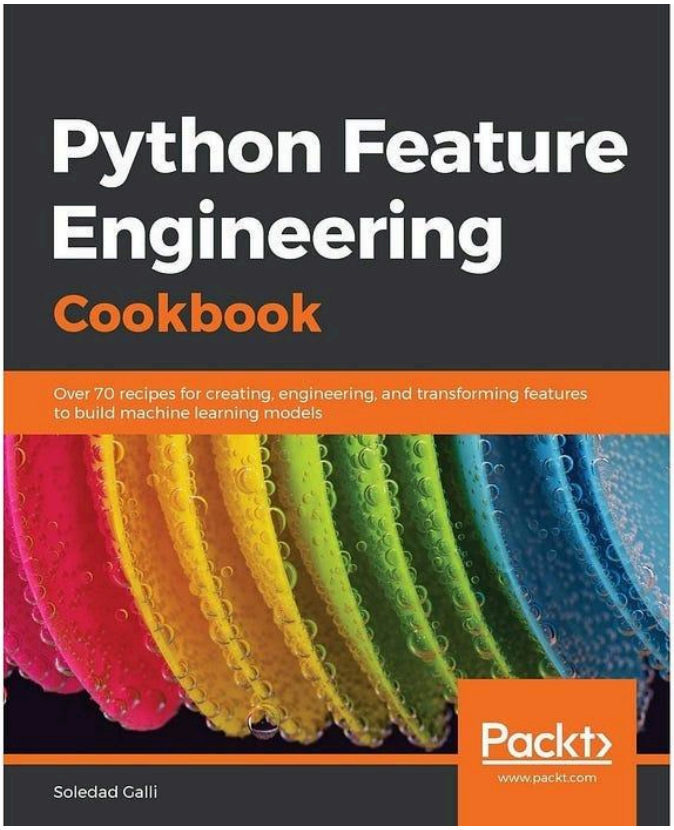
- **Difficulty Level:** Intermediate
- **Duration:** 10 hours
- **Price:** 24 euros
- **Instructor:** Soledad Galli, PhD

8.4. Additional Resources

8.4.1. Python Feature Engineering Cookbook

The first book in the optional resources is the **Python Feature Engineering Cookbook** by **Soledad Galli.** This book is by the same creator as the previous courses. Feature engineering, the process of transforming variables and creating features, albeit time-consuming, ensures that your machine learning models perform seamlessly. This second edition of Python Feature Engineering Cookbook will take the struggle out of feature engineering by showing you how to use open-source Python libraries to accelerate the process via a plethora of practical, hands-on recipes.

This updated edition begins by addressing fundamental data challenges such as missing data and categorical values, before moving on to strategies for dealing with skewed distributions and outliers. The concluding chapters show you how to develop new features from various types of data, including text, time series, and relational databases. With the help of numerous open-source Python libraries, you'll learn how to implement each feature engineering method in a performant, reproducible, and elegant manner.



By the end of this Python book, you will have the tools and expertise needed to confidently build end-to-end and reproducible feature engineering pipelines that can be deployed into production.

This book is for machine learning and data science students and professionals, as well as software engineers working on machine learning model deployment, who want to learn more about how to transform their data and create new features to train machine learning models in a better way.

Table of Contents:

1. Imputing Missing Data
2. Encoding Categorical Variables
3. Transforming Numerical Variables
4. Performing Variable Discretization
5. Working with Outliers
6. Extracting Features from Date and Time
7. Performing Feature Scaling
8. Creating New Features
9. Extracting Features from Relational Data with Featuretools
10. Creating Features from Time Series with tsfresh
11. Extracting Features from Text Variables

8.4.2. Feature Engineering and Selection

The second book on this list is **Feature Engineering and Selection: A Practical Approach for Predictive Models** written by **Max Kuhn** and **Kjell Johnson**. This book describes the general process of preparing raw data for modeling as feature engineering.

The process of developing predictive models includes many stages. Most resources focus on the modeling algorithms but neglect other critical aspects of the modeling process.

This book describes techniques for finding the best representations of predictors for modeling and for finding the best subset of predictors for improving model performance. A variety of example data sets are used to illustrate the techniques along with R programs for reproducing the results. This is a must-own book, even if R is not your primary language. The breadth of the methods discussed is worthy of owning it.

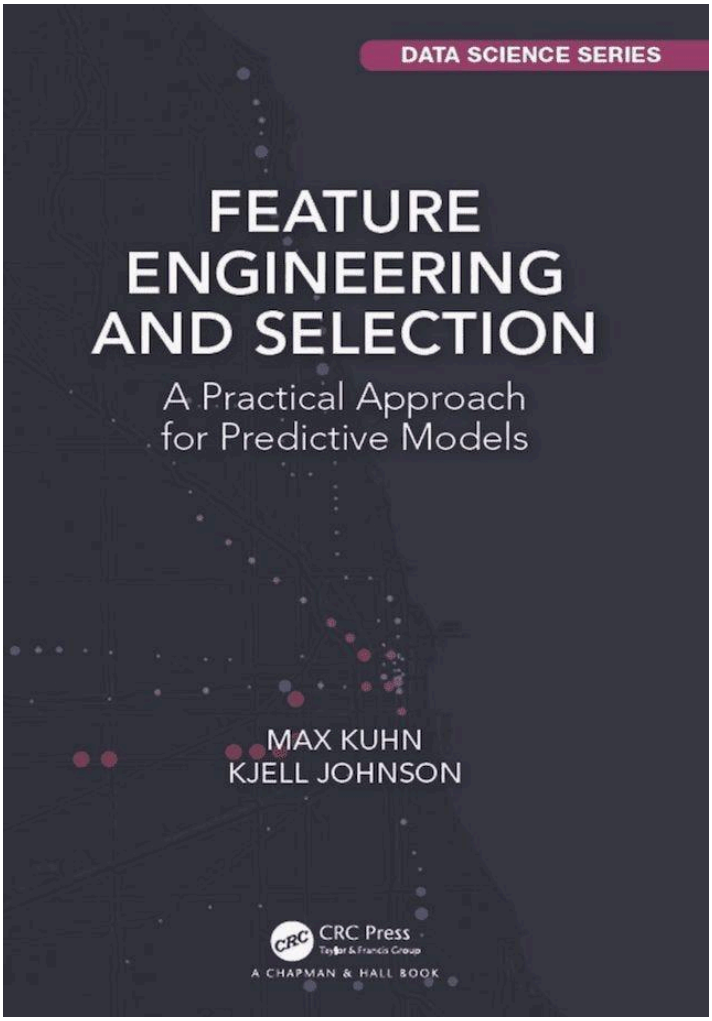
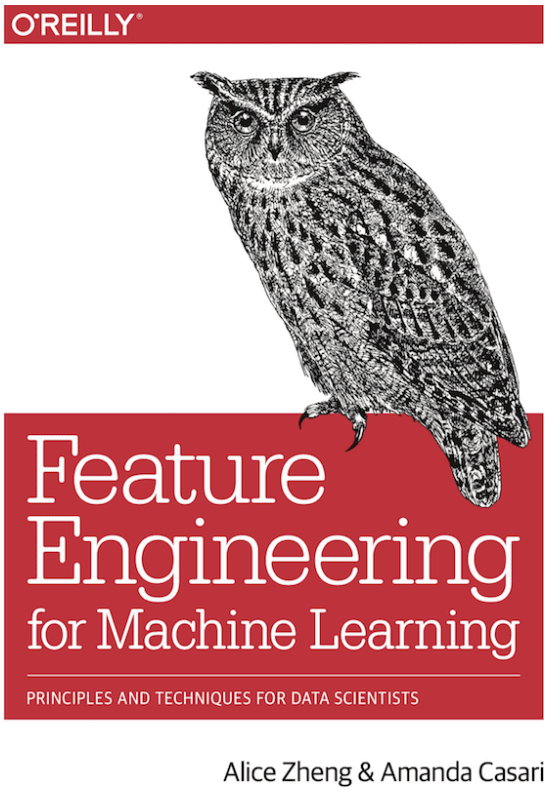


Table of contents:

- Chapter 1: Introduction
- Chapter 2: Illustrative Example: Predicting Risk Ischemic Stroke
- Chapter 3: A Review of the Predictive Modeling Process
- Chapter 4: Exploratory Visualizations
- Chapter 5: Encoding Categorical Predictors
- Chapter 6: Engineering Numeric Predictors
- Chapter 7: Detecting Interaction Effects
- Chapter 8: Handling Missing Data
- Chapter 9: Working with Profile Data
- Chapter 10: Feature Selection Overview
- Chapter 11: Greedy Search Methods
- Chapter 12: Global Search Methods

8.4.3. Feature Engineering for Machine Learning

The final book on this list is Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists by Alice Zheng and Amanda Casari.



In this practical book, you'll learn techniques for extracting and transforming features—the numeric representations of raw data—into formats for machine-learning models. Each chapter

guides you through a single data problem, such as how to represent text or image data. Together, these examples illustrate the main principles of feature engineering.

Rather than simply teach these principles, authors Alice Zheng and Amanda Casari focus on practical application with exercises throughout the book. The closing chapter brings everything together by tackling a real-world, structured dataset with several feature-engineering techniques. Python packages including NumPy, Pandas, Scikit-learn, and Matplotlib are used in code examples.

You'll examine:

- Feature engineering for numeric data: filtering, binning, scaling, log transforms, and power transforms
- Natural text techniques: bag-of-words, n-grams, and phrase detection
- Frequency-based filtering and feature scaling for eliminating uninformative features
- Encoding techniques of categorical variables, including feature hashing and bin-counting
- Model-based feature engineering with principal component analysis
- The concept of model stacking, using k-means as a featurization technique
- Image feature extraction with manual and deep-learning techniques

Table of Contents:

- Chapter 1: Machine Learning Pipeline
- Chapter 2: Fancy Tricks with Simple Numbers
- Chapter 3: Text Data: Flattening, Filtering, and Chunking
- Chapter 4: The Effects of Feature Scaling: From Bag-of-Words to Tf-Idf
- Chapter 5: Categorical Variables: Counting Eggs in the Age of Robotic Chickens
- Chapter 6: Dimensionality Reduction: Squashing the Data Pancake with PCA
- Chapter 7: Nonlinear Featurization via K-Means Model Stacking
- Chapter 8: Automating the Featurizer: Image Feature Extraction and Deep Learning
- Chapter 9: Back to the Future: Building an Academic Paper Recommender

9. Mastering Machine Learning

In the ninth chapter of the book, you will learn the fundamentals of machine learning for data science. In this step of the roadmap, you'll explore key concepts such as supervised and unsupervised learning, model evaluation, and feature engineering.

This guide will introduce essential algorithms, including linear regression, decision trees, and neural networks, along with practical applications using Python libraries like Scikit-Learn. By the end of this learning step of the roadmap, you'll have a solid foundation to build and fine-tune machine learning models, preparing you for more advanced topics in artificial intelligence and deep learning.

9.1. Machine Learning Specialization

The best and most complete resource to master the fundamentals of machine learning is the Machine Learning Specialization by Andrew NG provided by Deep Learning.ai on coursera. The Machine Learning Specialization is a foundational online program created in collaboration between DeepLearning.AI and Stanford Online. This beginner-friendly program will teach you the fundamentals of machine learning and how to use these techniques to build real-world AI applications.

This Specialization is taught by Andrew Ng, an AI visionary who has led critical research at Stanford University and groundbreaking work at Google Brain, Baidu, and Landing.AI to advance the AI field.

This 3-course Specialization is an updated version of Andrew's pioneering Machine Learning course, rated 4.9 out of 5 and taken by over 4.8 million learners since it launched in 2012.

It provides a broad introduction to modern machine learning, including supervised learning (multiple linear regression, logistic regression, neural networks, and decision trees), unsupervised learning (clustering, dimensionality reduction, recommender systems), and some of the best practices used in Silicon Valley for artificial intelligence and machine learning innovation (evaluating and tuning models, taking a data-centric approach to improving performance, and more.)

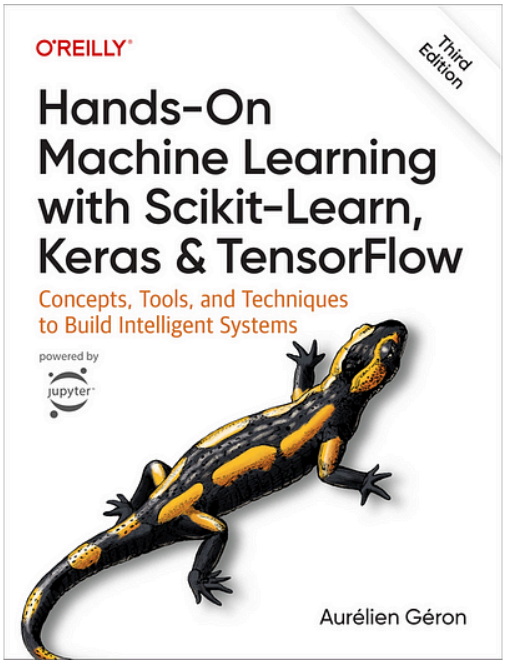
By the end of this Specialization, you will have mastered key concepts and gained the practical know-how to quickly and powerfully apply machine learning to challenging real-world problems. If you're looking to break into AI or build a career in machine learning, the new Machine Learning Specialization is the best place to start.

By the end of this Specialization, you will be ready to:

- Build machine learning models in Python using popular machine learning libraries NumPy and scikit-learn.
- Build and train supervised machine learning models for prediction and binary classification tasks, including linear regression and logistic regression.
- Build and train a neural network with TensorFlow to perform multi-class classification.
- Apply best practices for machine learning development so that your models generalize to data and tasks in the real world.
- Build and use decision trees and tree ensemble methods, including random forests and boosted trees.
- Use unsupervised learning techniques for unsupervised learning: including clustering and anomaly detection.
- Build recommender systems with a collaborative filtering approach and a content-based deep learning method.
- Build a deep reinforcement learning model.

9.2. Additional Resources

9.2.1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow



If you're looking for more resources after completing the course, I highly recommend **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**, particularly the first nine chapters.

The book explores a range of techniques, starting with simple linear regression and progressing to deep neural networks. Numerous code examples and exercises throughout the book help you apply what you've learned. Programming experience is all you need to get started.

The first nine chapters cover the following topics:

1. The Machine Learning Landscape
2. End-to-End Machine Learning Project
3. Classification
4. Training Models
5. Support Vector Machines
6. Decision Trees
7. Ensemble Learning and Random Forests
8. Dimensionality Reduction
9. Unsupervised Learning Techniques

9.3. Putting it into Action

Now you are ready to put your knowledge into action and work on multiple case studies to work on in your domain of interest. It is time now to narrow down your selection. My suggestion is to have at least three solid case studies that cover the basic machine learning tasks, which are regression, classification, and clustering.

To elaborate more, let’s take a practical example. Let’s assume that you are interested in working in the healthcare domain. So after searching, you came up with multiple case studies in this field.

Actions:

- Discover the different areas of AI and know your interests.
- Narrow down your case studies into at least three that cover the basic machine learning tasks and focus on your area of interest in AI.
- Define which case study is solved by which machine learning task.

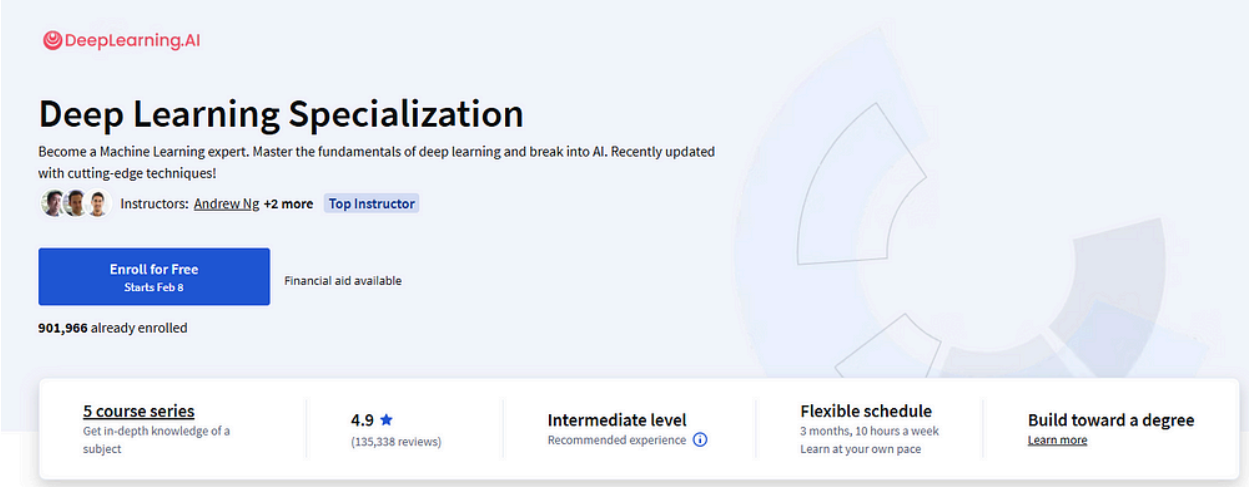
10. Deep Learning Fundamentals

Deep learning has become an essential skill for data scientists, enabling breakthroughs in fields like computer vision, natural language processing, and generative AI. The roadmap introduces carefully curated courses that guide learners from foundational neural networks to specialized applications in NLP and computer vision.

It begins with Andrew Ng's Deep Learning Specialization to establish core concepts, followed by targeted courses on NLP and vision, helping learners gain hands-on experience with architectures like transformers, CNNs, and RNNs.

To further enhance learning, the article includes additional resources such as Stanford's CS231n and CS224n courses, as well as essential deep learning books. By the end, readers will have a structured path to mastering deep learning and applying it in real-world projects.

10.1. Deep Learning Specialization



The first course on the list is the **Deep Learning Specialization** by Andrew Ng. This foundational program will help you understand the capabilities, challenges, and consequences of deep learning and prepare you to participate in developing leading-edge AI technology.

In this Specialization, you will build and train neural network architectures such as Convolutional Neural Networks, Recurrent Neural Networks, LSTMs, and Transformers, and learn how to make them better with strategies such as Dropout, BatchNorm, Xavier/He initialization, and more.

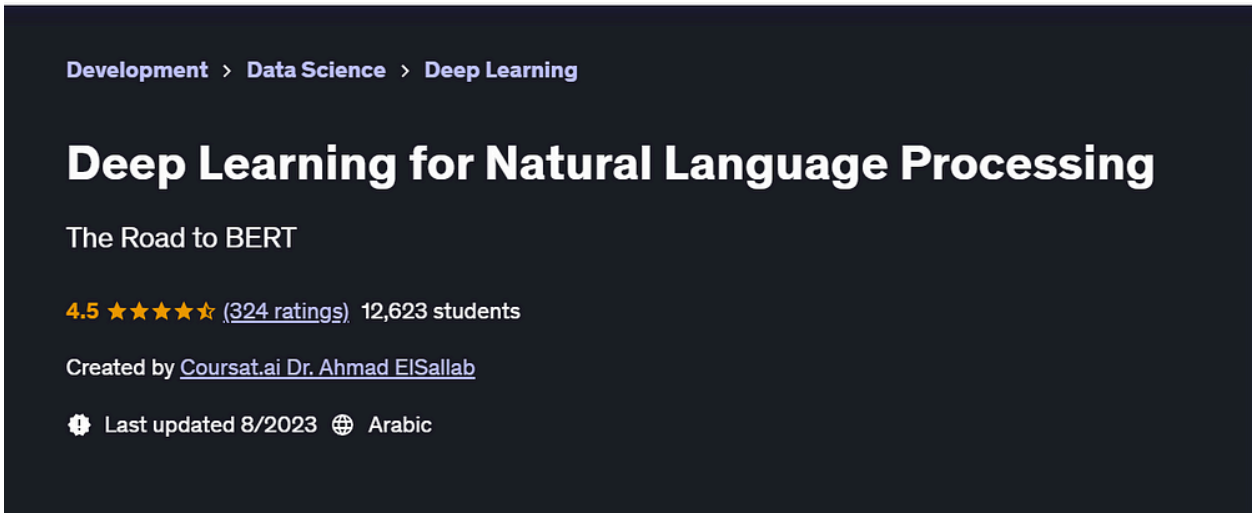
By the end you'll be able to:

- Build and train deep neural networks, implement vectorized neural networks, identify architecture parameters, and apply DL to your applications
- Use best practices to train and develop test sets and analyze bias/variance for building DL applications, use standard NN techniques, apply optimization algorithms, and implement a neural network in TensorFlow
- Use strategies for reducing errors in ML systems, understand complex ML settings, and apply end-to-end, transfer, and multi-task learning
- Build a Convolutional Neural Network, apply it to visual detection and recognition tasks, use neural style transfer to generate art, and apply these algorithms to image video, and other 2D/3D data
- Build and train Recurrent Neural Networks and their variants (GRUs, LSTMs), apply RNNs to character-level language modeling, work with NLP and Word Embeddings, and use HuggingFace tokenizers and transformers to perform Named Entity Recognition and question-answering.

Course Information:

- **Difficulty Level:** Beginner
- **Duration:** 1 month, 4 hours a day
- **Price:** 40$ (Financial aid available)
- **Instructor:** Andrew Ng

10.2. Deep Learning for Natural Language Processing



The second course is **Deep Learning for Natural Language Processing**, which will help you dive into the world of Natural Language Processing which is the second important step after finishing the fundamentals of deep learning in the first course.

You will learn how Deep Learning has reshaped this area of AI using concepts like word vectors and embeddings, structured deep learning, collaborative filtering, recurrent neural networks, sequence-to-sequence models, and transformer networks.

You will start the journey by going through the traditional pipeline of text pre-processing and the different text features like binary and TF-IDF features with the Bag-of-Words model.

Then you will dive into the concepts of word vectors and embeddings as a general deep learning concept, with a detailed discussion of famous word embedding techniques like word2vec, GloVe, Fasttext, and ELMo.

This will enable us to divert into recommender systems, using collaborative filtering and the twin-tower model as an example of the generic usage of embeddings beyond word representations.

In the second part of the course, you will be concerned with sentence and sequence representations. We will tackle the core NLP of Langauge Modeling, at statistical and neural levels, using recurrent models, like LSTM and GRU.

In the following part, we tackle sequence-to-sequence models, with the flagship NLP task of Machine Translation, which paves the way to talk about many other tasks under the same design seq2seq pattern, like Question-Answering and Chatbots.

We present the core idea of Attention mechanisms with recurrent seq2seq before we generalize it as a generic deep learning concept. This generalization led to the state-of-the-art Transformer Network, which revolutionized the world of NLP, using full attention mechanisms.

In the final part of the course, we present the ImageNet moment of NLP, where Transfer Learning comes into play together with pre-trained Transformer architectures like BERT, GPT 1–2–3, RoBERTa, ALBERT, XLTransforme,r, and XLNet.

Course Information:

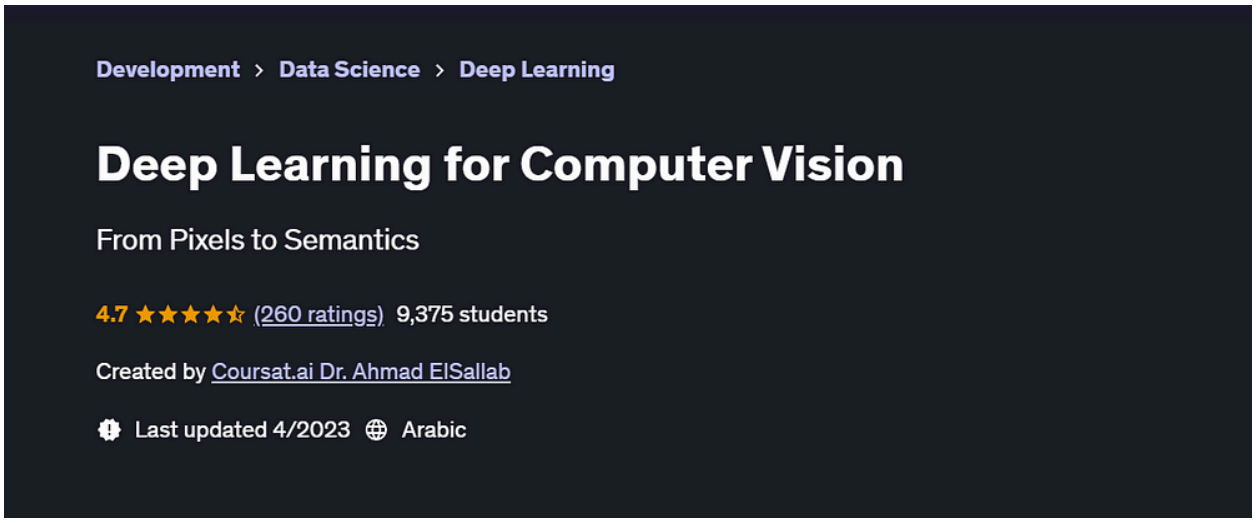
- **Difficulty Level:** Intermediate
- **Duration:** 1 month, 4 hours a day
- **Price:** 20$
- **Instructor:** Ahmed ElSallab

10.3. Deep Learning for Computer Vision

The third course is the **Deep Learning for Computer Vision** course which will help you build computer vision fundamentals. The course consists of three main parts.

The first part covers the essentials of a traditional computer vision pipeline, and how to deal with images in OpenCV and Pillow libraries, including the image pre-processing pipeline like: thresholding, denoising, blurring, filtering, edge detection, contours…etc.

You will build simple apps like Car License Plate Detection (LPD) and activity recognition. This will lead us to the revolution that deep learning brought to the game of computer vision, turning traditional filters into learnable parameters using Convolution Neural Networks.



We will cover all the basics of ConvNets, including the details of the Vanilla architecture for image classification, hyperparameters like kernels, strides, maxpool, and feature map size calculations. Beyond the Vanilla architecture, we also cover the state-of-the-art ConvNet meta-architectures and design patterns, like skip-connections, Inception, DenseNet…etc.

In the second part, we will learn how to use ConvNets to solve practical problems in different situations, with small amounts of data, how to use transfer learning and the different scenarios for that, and finally how to debug and visualize the leaned kernels in ConvNets.

In the last part, we will learn about different CV apps using ConvNets. We will learn about the Encoder-Decoder design pattern. We start with the task of semantic segmentation, where we will build a U-Net architecture from scratch for the Cambridge Video (CAMVID) dataset.

Then we will learn about Object Detection, covering both 2-stage and one-shot architectures like SSD and YOLO. Next, we will learn how to deal with the video data using the Spatio-Temporal ConvNet architectures. Finally, we will introduce 3D Deep Learning to extend ConvNets usage to deal with 3D data, like LiDAR data.

Course Information:

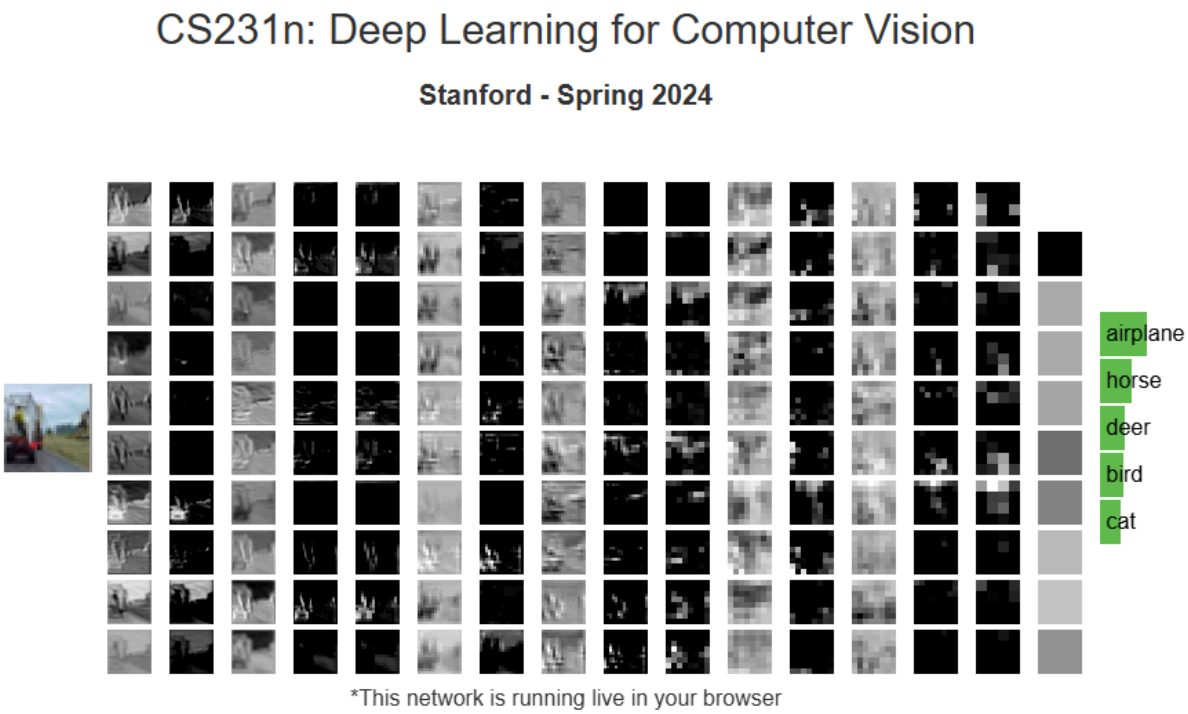
- **Difficulty Level:** Intermediate
- **Duration:** 1 month, 4 hours a day
- **Price:** 20$

- **Instructor:** Ahmed ElSallab

10.4. Additional Resources

If you want to explore specific areas of deep learning beyond the courses listed in this roadmap, here are some high-quality resources that can help you deepen your understanding. These cover theoretical concepts, hands-on projects, research papers, and specialized applications of deep learning.

10.4.1. Stanford University CS231n: Deep Learning for Computer Vision

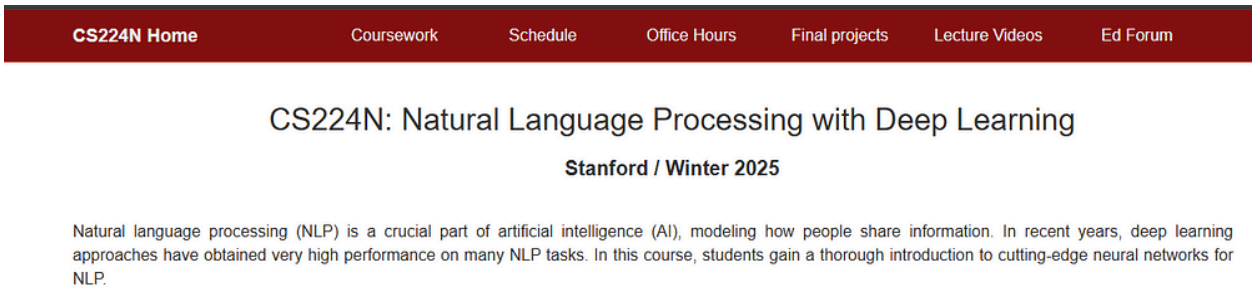


The [Stanford University CS231n: Deep Learning for Computer Vision](#) course is a deep dive into the details of deep learning architectures with a focus on learning end-to-end models for these tasks, particularly image classification.

During the 10-week course, students will learn to implement and train their own neural networks and gain a detailed understanding of cutting-edge research in computer vision.
Additionally, the final assignment will give them the opportunity to train and apply multi-million parameter networks on real-world vision problems of their choice.

Through multiple hands-on assignments and the final course project, students will acquire the toolset for setting up deep learning tasks and practical engineering tricks for training and fine-tuning deep neural networks.

10.4.2. CS224n: Natural Language Processing with Deep Learning



In this course CS224n: Natural Language Processing with Deep Learning students will gain a thorough introduction to both the basics of Deep Learning for NLP and the latest cutting-edge research on Large Language Models (LLMs).

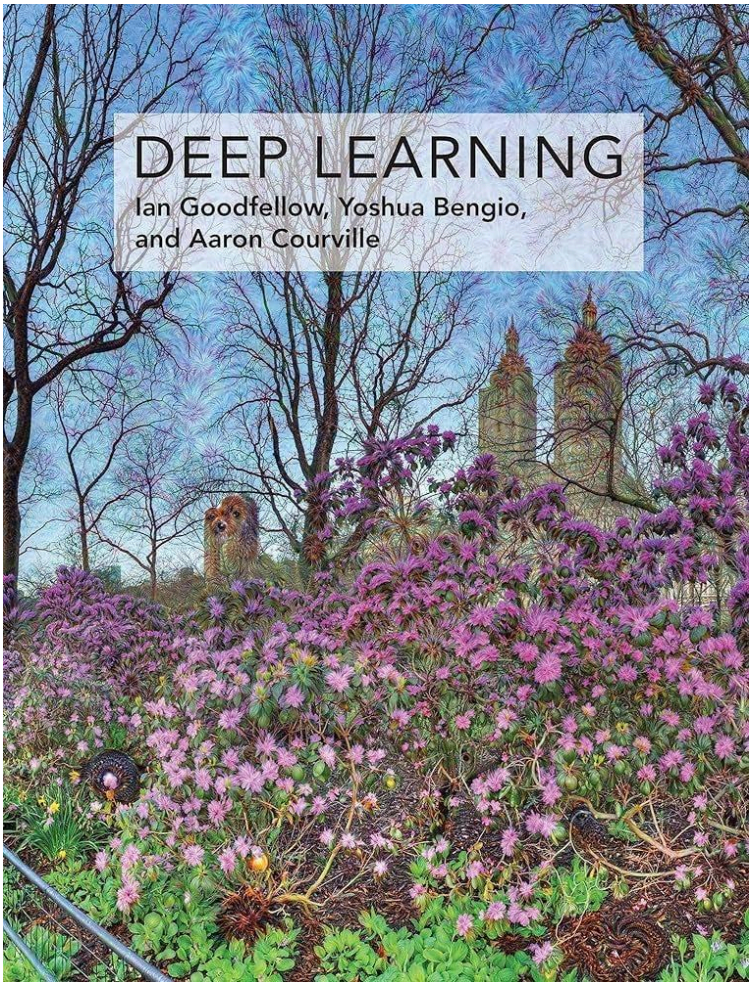
Through lectures, assignments, and a final project, students will learn the necessary skills to design, implement, and understand their own neural network models, using the Pytorch framework.

10.4.3. The Deep Learning Book

The **Deep Learning book** introduces a broad range of topics in deep learning. The text offers mathematical and conceptual background, covering relevant concepts in linear algebra, probability theory and information theory, numerical computation, and machine learning.

It describes deep learning techniques used by practitioners in industry, including deep feedforward networks, regularization, optimization algorithms, convolutional networks, sequence modeling, and practical methodology; and it surveys such applications as natural language processing, speech recognition, computer vision, online recommendation systems, bioinformatics, and videogames.

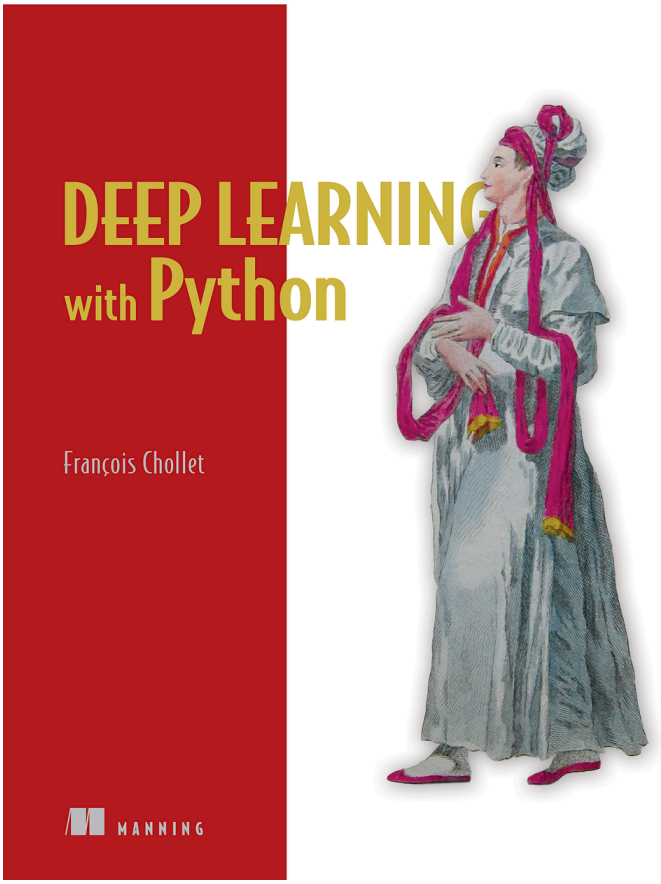
Finally, the book offers research perspectives, covering such theoretical topics as linear factor models, autoencoders, representation learning, structured probabilistic models, Monte Carlo methods, the partition function, approximate inference, and deep generative models.



10.4.4. Deep Learning with Python Book

Deep Learning with Python introduces the field of deep learning using the Python language and the powerful Keras library. Written by Keras creator and Google AI researcher François Chollet, this book builds your understanding through intuitive explanations and practical examples.

You'll explore challenging concepts and practice with applications in computer vision, natural language processing, and generative models. By the time you finish, you'll have the knowledge and hands-on skills to apply deep learning to your projects.



10.5. Putting it into Actions

Now that you've followed a structured roadmap and built a solid foundation in deep learning, it's time to move from theory to practice. The best way to reinforce your learning is by working on real projects, contributing to open-source repositories, and engaging with the deep learning community.

One of the best frameworks for developing deep learning projects is **PyTorch**. It offers flexibility, ease of debugging, and strong community support. **Steps to Develop Your Deep Learning Project Using PyTorch:**

- **Step 1: Define Your Problem**—Choose a real-world challenge that interests you. Examples:
  1. Image Classification (e.g., Dog vs. Cat classification)
  2. Sentiment Analysis (e.g., Classifying movie reviews as positive or negative)
  3. Object Detection (e.g., Identifying cars in a traffic video)
- **Step 2: Collect and Prepare Data**—Use datasets from **Kaggle, Hugging Face Datasets, or OpenAI's Dataset Hub**. Preprocess the data using libraries like OpenCV (for images) or NLTK (for text).

- **Step 3: Build Your Model Using PyTorch**—Implement a deep learning model using PyTorch's `torch.nn.Module`. For example, if you're building an image classifier, you can start with a **pretrained ResNet model** from `torchvision.models`.
- **Step 4: Train and Optimize**—Use **Adam optimizer and CrossEntropy loss** for training. Implement techniques like **learning rate scheduling, dropout, and data augmentation** to improve performance.
- **Step 5: Evaluate the Model**—Measure accuracy, precision, recall, and F1-score using `torchmetrics`. Visualize predictions using Matplotlib.
- **Step 6: Deploy the Model**—Convert the trained model to **TorchScript** and deploy it with **FastAPI or Flask**. Use **Streamlit** if you want to build an interactive web interface.
- **Step 7: Share Your Work**—Publish your project on **GitHub**, write a blog post explaining your approach, and showcase it in your portfolio.

11. Generative AI & Large Language Models (LLMs) Fundamentals

Part 11 of the **Beginner-to-Upper Intermediate Data Science Roadmap for 2025** will cover the fundamentals of Generative AI & Large Language Models (LLMs).

It will cover key topics such as prompt engineering, retrieval-augmented generation (RAG), and fine-tuning LLMs, providing essential resources, including books and university courses, to deepen understanding.

The final section emphasizes hands-on learning by guiding readers through three practical projects: crafting effective prompts, building a RAG-powered search system, and fine-tuning an open-source LLM.

By the end of this chapter, learners will have the knowledge and practical experience to build and deploy generative AI applications in real-world scenarios.

11.1. Generative AI with Large Language Models

In Generative AI with Large Language Models (LLMs), you'll learn the fundamentals of how generative AI works, and how to deploy it in real-world applications.

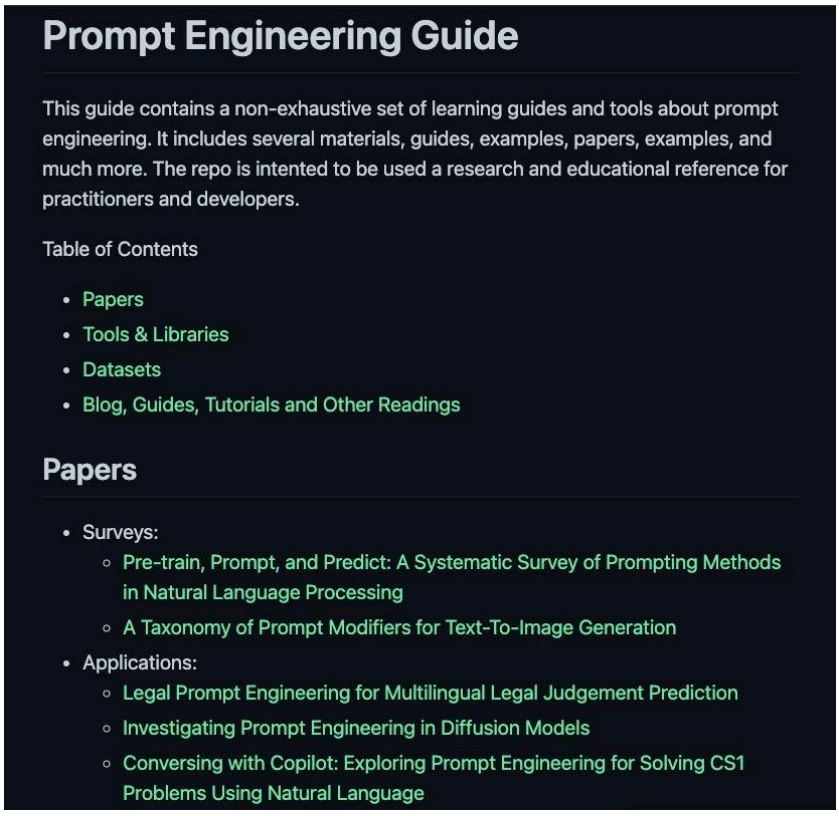
By taking this course, you'll learn to:

- Deeply understand generative AI, describing the key steps in a typical LLM-based generative AI lifecycle, from data gathering and model selection to performance evaluation and deployment.
- Describe in detail the transformer architecture that powers LLMs, how they're trained, and how fine-tuning enables LLMs to be adapted to a variety of specific use cases.
- Use empirical scaling laws to optimize the model's objective function across dataset size, compute budget, and inference requirements.
- Apply state-of-the-art training, tuning, inference, tools, and deployment methods to maximize the performance of models within the specific constraints of your project.
- Discuss the challenges and opportunities that generative AI creates for businesses after hearing stories from industry researchers and practitioners.

Course Details:

- **Course link**
- **Duration**: 1.5 Weeks (4 hours/day)
- **Provider**: Deep Learning.ai on Coursera
- **Price**: 40$ (Financial aid available)

11.2. Prompt Engineering Guide

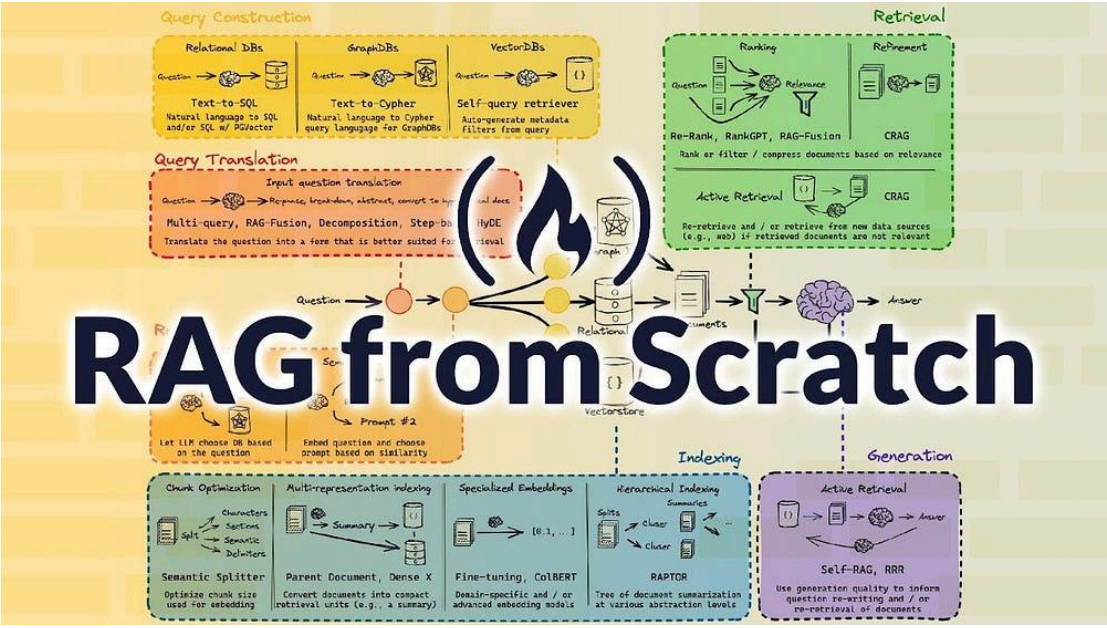


Prompt engineering is a relatively new discipline for developing and optimizing prompts to efficiently use language models (LMs) for various applications and research topics.
Prompt engineering skills help better understand the capabilities and limitations of large language models (LLMs). Researchers use prompt engineering to improve the capacity of LLMs on a wide range of common and complex tasks such as question answering and arithmetic reasoning. Developers use prompt engineering to design robust and effective prompting techniques that interface with LLMs and other tools. Motivated by the high interest in developing with LLMs, Elvis Saravia has created this Prompt Engineering Guide that contains all the latest papers, learning guides, lectures, references, and tools related to prompt engineering for LLMs.

Course Details:

- Course link
- Duration: 2 days (4 hours/day)
- Price: Free

11.3. Learn RAG From Scratch



Learn how to implement RAG from scratch, straight from a LangChain software engineer. This Python course teaches you how to use RAG to combine your own custom data with the power of Large Language Models (LLMs).

Course Contents:

- Indexing
- Retrieval
- Generation
- Query Translation (Multi-Query)

- Query Translation (RAG Fusion)
- Query Translation (Decomposition)
- Query Translation (Step Back)
- Query Translation (HyDE)
- Routing
- Query Construction
- Indexing (Multi Representation)
- Indexing (RAPTOR)
- Indexing (ColBERT)
- CRAG
- Adaptive RAG

Course Details:

- **Course link**
- **Duration**: 1 day (4 hours/day)
- **Price**: Free

11.4. Fine Tuning LLM Models—Generative AI Course

Learn how to fine-tune LLM models. This course will teach you fine-tuning using QLORA and LORA, as well as Quantization using LLama2, Gradient, and the Google Gemma model. This

crash course includes both theoretical and practical instruction to help you understand how to perform fine-tuning.

Course Contents:

- Introduction
- Quantization Intuition
- Lora And QLORA In-depth Intuition
- Finetuning With LLama2
- 1 bit LLM In-depth Intuition
- Finetuning with Google Gemma Models
- Building LLm Pipelines With No code
- Fine-tuning With Own Custom Data

Course Details:

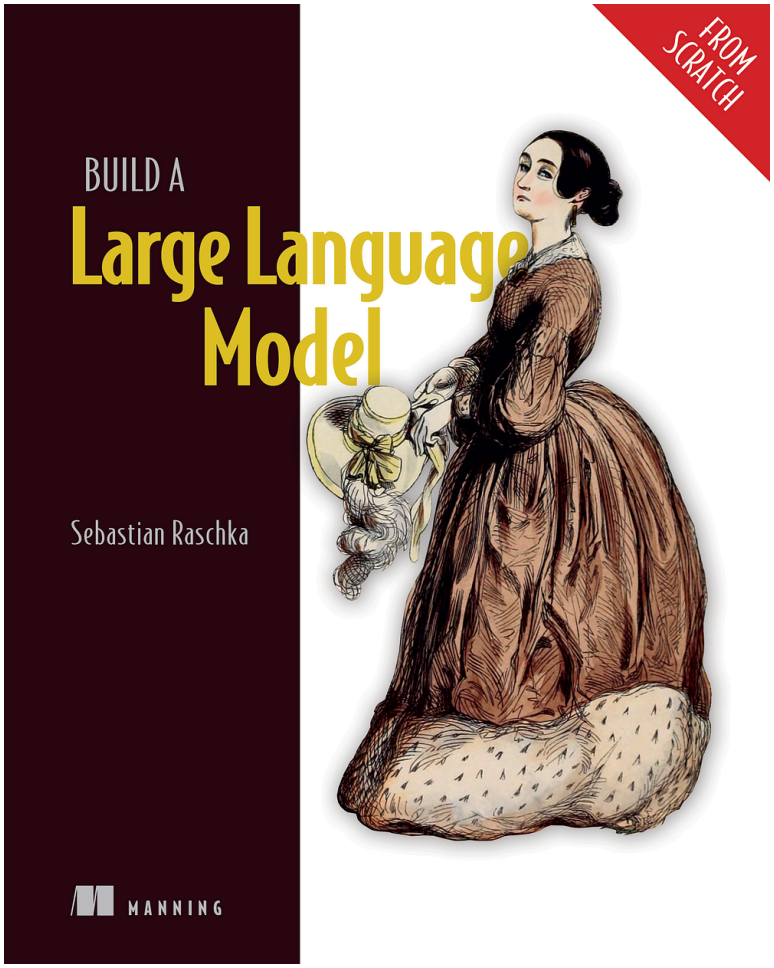
- **Course link**
- **Duration**: 1 day (4 hours/day)
- **Price**: Free

11.5. Additional Resources

11.5.1. Important Books

1. Build LLM from Scratch

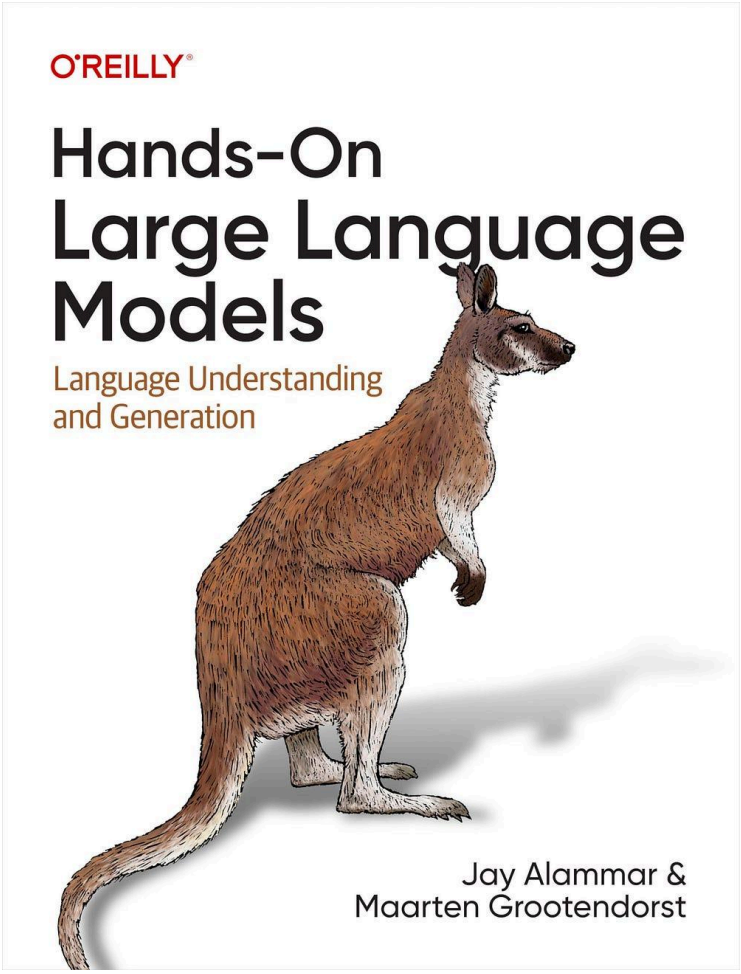
In **Build a Large Language Model (from Scratch)** bestselling author Sebastian Raschka guides you step by step through creating your own LLM. Each stage is explained with clear text, diagrams, and examples. You'll go from the initial design and creation to pretraining on a general corpus, and on to fine-tuning for specific tasks.



2. Hands-on Large Language Models

Through **Hands-on Large Language Models**'s visually educational nature, readers will learn practical tools and concepts they need to use these capabilities today.

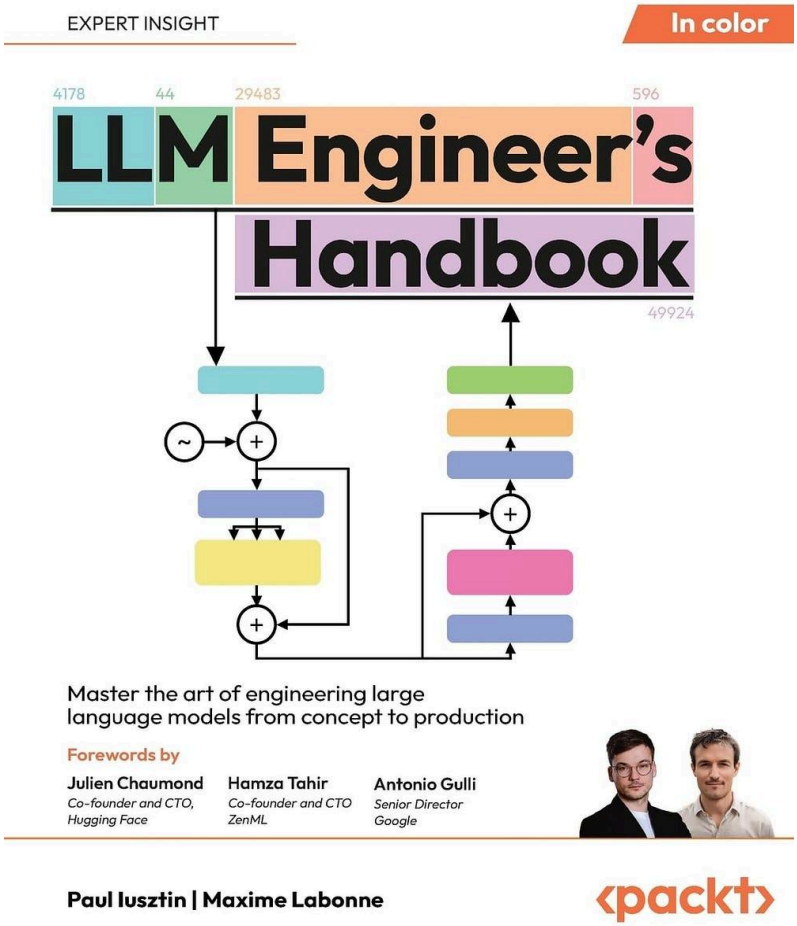
You'll understand how to use pretrained large language models for use cases like copywriting and summarization; create semantic search systems that go beyond keyword matching; and use existing libraries and pretrained models for text classification, search, and clusterings.



3. LLM Engineering Handbook

Throughout the **LLM Engineering Handbook**, you will learn data engineering, supervised fine-tuning, and deployment. The hands-on approach to building the LLM Twin use case will help you implement MLOps components in your own projects.
You will also explore cutting-edge advancements in the field, including inference optimization, preference alignment, and real-time data processing, making this a vital resource for those looking to apply LLMs in their projects.

By the end of this book, you will be proficient in deploying LLMs that solve practical problems while maintaining low-latency and high-availability inference capabilities.
Whether you are new to artificial intelligence or an experienced practitioner, this book delivers guidance and practical techniques that will deepen your understanding of LLMs and sharpen your ability to implement them effectively.



11.5.2. Important University Courses

1. Advanced NLP—Carnegie Mellon University

Carnegie Mellon University Language Technologies Institute

The Advanced NLP course focuses on modern methods using neural networks and covers the basic modeling and learning algorithms required. The class culminates in a project in which students attempt to reimplement and improve upon a research paper on a topic of their choosing.

In the course, we describe fundamental tasks in natural language processing such as syntactic, semantic, and discourse analysis, as well as methods to solve these tasks. The course focuses on modern methods using neural networks and covers the basic modeling and learning algorithms required. The class culminates in a project in which students attempt to reimplement and improve upon a research paper on a topic of their choosing.

2. Recent Advances on Foundation Models—University of Waterloo

CS 886: Recent Advances on Foundation Models is a graduate-level course at the University of Waterloo, focusing on the latest developments in foundation models, including transformers, large language models, and multimodal models.

3. Large Language Model Agents—University of California, Berkeley

In the Large Language Model Agents course, we will first discuss fundamental concepts that are essential for LLM agents, including the foundation of LLMs, essential LLM abilities required for task automation, as well as infrastructures for agent development.
We will also cover representative agent applications, including code generation, robotics, web automation, medical applications, and scientific discovery. Meanwhile, we will discuss the limitations and potential risks of current LLM agents, and share insights into directions for further improvement.
Specifically, this course will include the following topics:

- Foundation of LLMs
- Reasoning
- Planning, tool use
- LLM agent infrastructure
- Retrieval-augmented generation
- Code generation, data science
- Multimodal agents, robotics
- Evaluation and benchmarking on agent applications
- Privacy, safety, and ethics
- Human-agent interaction, personalization, alignment
- Multi-agent collaboration

11.6. Putting it into Action

Now that you've covered the fundamentals of Generative AI and Large Language Models (LLMs), the best way to solidify your understanding is by working on hands-on projects. Below are three key projects that will help you develop practical skills in prompt engineering, retrieval-augmented generation (RAG), and fine-tuning LLMs.

1. Prompt Engineering: Building an AI-Powered Writing Assistant

- **Goal:** Create a prompt-driven AI writing assistant that helps generate blog posts, summaries, or creative content.
- **Tools:** OpenAI's GPT, Claude, or any open-source LLM (Mistral, Llama 3).

Steps:

1. Experiment with different prompt techniques like zero-shot, few-shot, and chain-of-thought prompting.
2. Implement prompt templates to structure responses effectively.
3. Optimize and test prompts for accuracy, coherence, and creativity.

- **Outcome:** A practical understanding of how to craft and refine prompts to get the best results from LLMs.

2. Building a RAG Application: AI-Powered Document Search

- **Goal:** Develop a RAG-based system that retrieves relevant information from a custom knowledge base.
- **Tools:** LangChain, FAISS/LanceDB/ChromaDB, Hugging Face Transformers.

Steps:

1. Load and preprocess domain-specific documents.
2. Split text into chunks and embed them into a vector database.
3. Implement a retrieval pipeline that fetches relevant chunks based on user queries.
4. Integrate an LLM to generate responses based on retrieved context.

- **Outcome:** A working AI-powered search system that efficiently retrieves and summarizes information.

3. Fine-Tuning LLMs: Customizing a Model for a Specific Task

- **Goal:** Fine-tune an open-source LLM on a domain-specific dataset to improve its performance.
- **Tools:** Hugging Face Transformers, PEFT (LoRA/QLoRA), PyTorch.

Steps:

1. Select a base model (e.g., Mistral 7B, Llama 3).

2. Collect and preprocess a task-specific dataset.
3. Use parameter-efficient fine-tuning techniques to adapt the model.
4. Evaluate performance and deploy the fine-tuned model.

- **Outcome:** Hands-on experience in customizing LLMs for specialized use cases.

By completing these projects, you'll transition from theory to practice, gaining the experience needed to build real-world generative AI applications. Whether you aim to enhance your personal projects, contribute to open-source initiatives, or land a job in AI, these projects will set you on the right path.

12. Machine Learning Operations (MLOps) Fundamentals

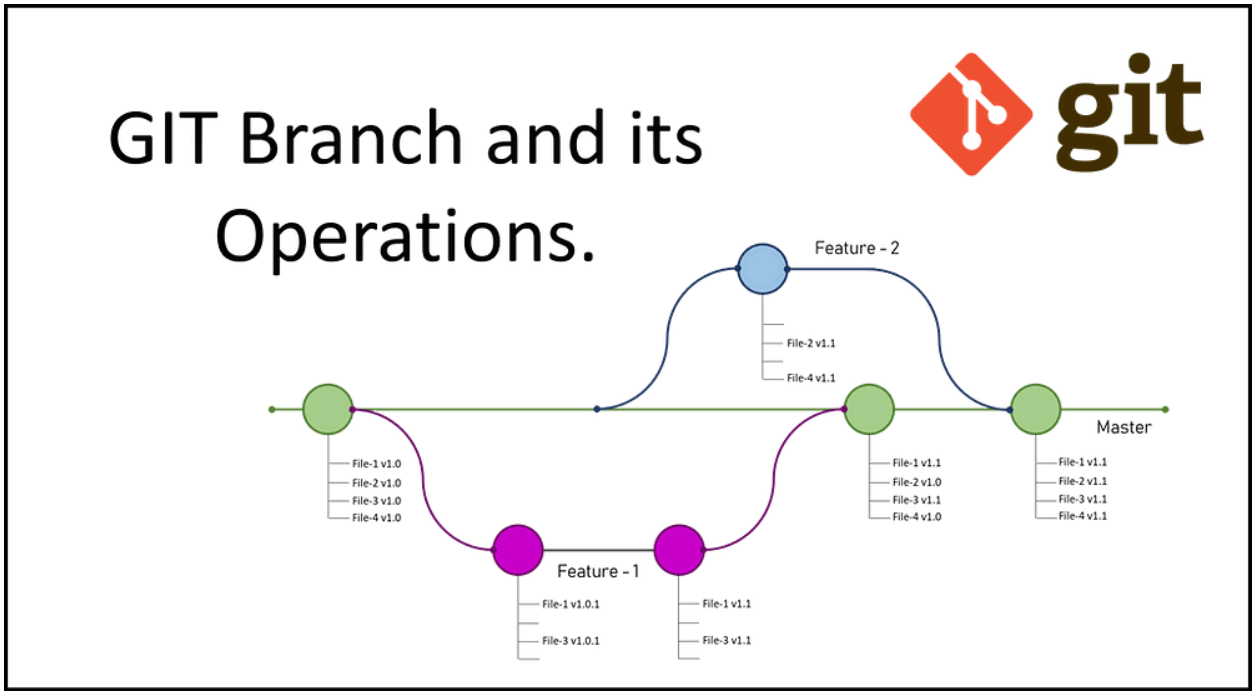
There are many challenges in bringing your machine learning systems into production, which include construction, integrating, testing, releasing, deployment, and infrastructure management.

Therefore it is important to follow good practices and know how to overcome these challenges. MLOps technologies are tools and platforms that help organizations manage and optimize machine learning models' development, deployment, and maintenance.

In the 12th part of this series, we will provide you with a comprehensive guide to learning about MLOps, including the key concepts and skills that you need to master. We will also provide you with a selection of the best free learning resources available online to help you get started on your MLOps journey.

Whether you are new to MLOps or have some experience under your belt, this roadmap will provide you with a clear and structured path to follow to help you become an expert in this exciting and rapidly evolving field. So, let's get started!

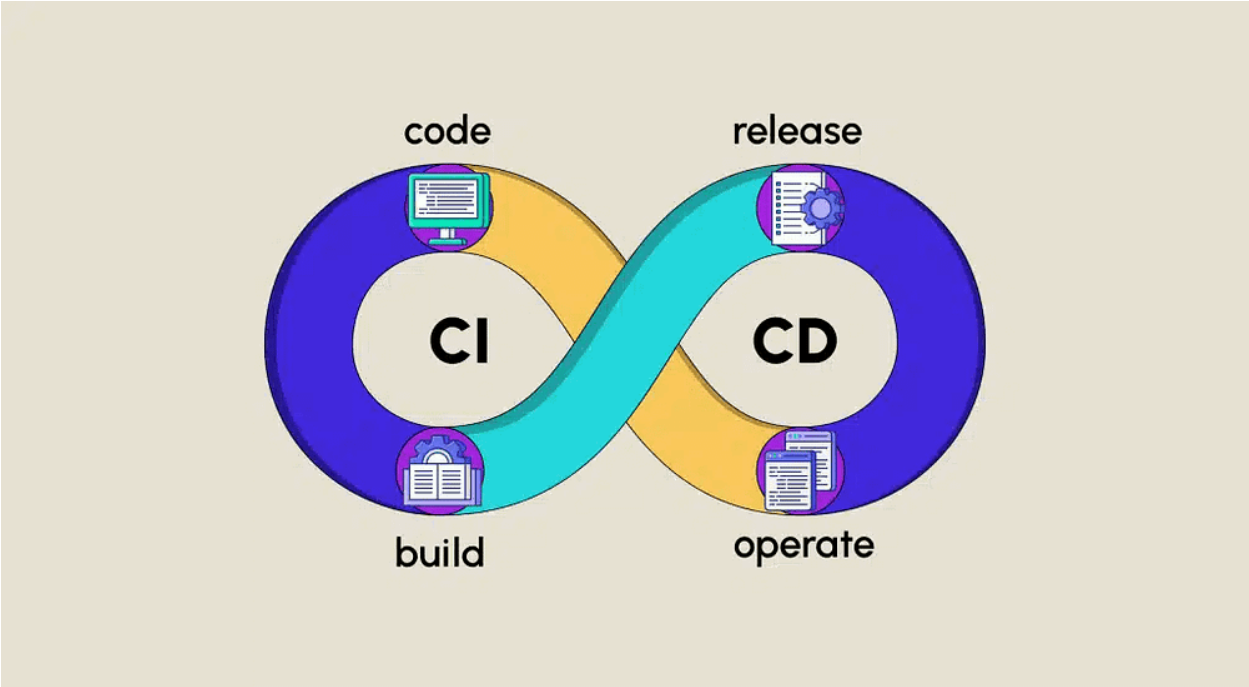
12.1. Version Control for Machine Learning



The first step in this learning roadmap is to master version control/ and how it is used in machine learning. This includes understanding tools such as Git and how to use them to track changes to your code and models.
Working in production demands data scientists and machine learning engineers to know version control. Since you will be working in cooperation with other data scientists, data engineers, and software engineers, therefore it is important to be able to share your code and update it and also to follow up on their updates in a professional way.
Learning Resources:

- **Git and GitHub Tutorial For Beginners | Full Course**
- **Intro to MLOps: Data and Model Versioning**

2. Continuous Integration & Continuous Delivery (CI/CD) Tools



The second step is to learn about continuous integration and continuous delivery (CI/CD) tools and how they can automate building, testing, and deploying machine learning models. Continuous Integration (CI) and Continuous Delivery (CD) are key practices in the field of MLOps that help to automate and streamline the process of building, testing, and deploying machine learning models. CI/CD tools are software tools that assist with these practices by providing automated pipelines for building, testing, and deploying code. Some popular CI/CD tools include:

- **Jenkins:** an open-source CI/CD tool that is widely used and highly configurable.
- **Travis CI:** a popular cloud-based CI/CD tool that is easy to set up and use, particularly for open-source projects.

- **GitLab CI:** a CI/CD tool that is closely integrated with the GitLab version control system, making it easy to set up CI/CD pipelines for projects managed with GitLab.
- **CircleCI:** a cloud-based CI/CD tool that is popular for its easy setup and fast build times.

It's important to choose the right CI/CD tool for your project, taking into consideration factors such as your team's size, the complexity of your project, and your budget.

Learning Resources:

- **Jenkins**: MLOps Tutorial—Building a CI/ CD Machine Learning Pipeline
- **TravisCI**: Getting Started with Travis
- **GitLab**: GitLab CI CD Tutorial for Beginners [Crash Course]
- **CircleCI**: How to Get Started with CircleCI

12.3. Infrastructure & Resource Management for Machine Learning

The third step is to learn about infrastructure and resource management tools for machine learning. This includes understanding how to provision and manage computing resources for training and deploying machine learning models and how to scale machine learning pipelines. Examples of infrastructure and resource management tools include:

1. **Kubernetes**: This open-source system allows you to automate the deployment, scaling, and management of containerized applications. It can be particularly useful for managing machine learning workflows, as it allows you to easily scale up or down as needed.
2. **Docker**: It is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow you to package an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package. This makes it easier to run the application on any other machine because everything it needs is contained in the package. Docker is often used in conjunction with container orchestration tools like Kubernetes to manage the deployment and scaling of containerized applications. It is also commonly used for developing and testing machine

learning applications, as it allows you to create isolated environments with specific dependencies and packages.

3. **Amazon SageMaker**: This fully managed service from Amazon Web Services (AWS) simplifies the process of building, training, and deploying machine learning models. It includes tools for resource management, such as the ability to select the right hardware and automatically scale up or down as needed.
4. **Google Cloud AI Platform:** This cloud-based platform from Google includes tools for building, deploying, and managing machine learning models. It includes features such as automatic scaling and resource management to help you optimize your machine-learning workflow.

Overall, there are many tools available to help with infrastructure and resource management for machine learning, and the right choice for you will depend on your specific needs and preferences.

Learning Resources:

- **Kubernetes**: Kubernetes Tutorial for Beginners [FULL COURSE in 4 Hours]
- **Docker**: Docker Tutorial for Beginners [FULL COURSE in 3 Hours]
- **AWS SageMaker**: Amazon (AWS) Sagemaker Full Course | Getting Started

12.4. Machine Learning Monitoring & Observability Tools

The fourth step is to learn about monitoring and observability tools for machine learning. This includes understanding how to monitor the performance and health of machine learning models

in production and troubleshooting and debugging issues that may arise. Examples of monitoring and observability tools include:

1. **TensorBoard**: This tool, developed by Google, is a web-based visualization tool for machine learning experiments. It allows you to view metrics such as loss and accuracy, as well as visualize the structure of your model.
2. **Prometheus**: This open-source monitoring system is designed to collect and store metrics from your applications and infrastructure. It includes a query language for analyzing the data and creating alerts.
3. **Datadog**: This cloud-based monitoring platform allows you to track metrics, logs, and events from your applications and infrastructure. It includes tools for visualizing and analyzing data, as well as for creating alerts.
4. **ELK Stack**: The ELK Stack is a collection of open-source tools for collecting, storing, and analyzing logs. It includes Elasticsearch for storing and searching logs, Logstash for collecting and processing logs, and Kibana for visualizing and analyzing the data.

Learning Resources:

- **Grafana and Prometheus** : Grafana and Prometheus Crash Course
- **Datadog**: Datadog Tutorials
- **AWS CloudWatch**: AWS CloudWatch Demonstration
- **Weight & Biases**: Weights & Biases Crash Course

12.5. Managing Machine Learning Projects & Pipelines

The fifth step in this roadmap is to learn about tools and platforms for managing machine learning projects and pipelines. This includes understanding how to track and collaborate on machine learning projects and orchestrate and automate machine learning pipelines.

Examples of project management and pipeline orchestration tools include:

1. **Argo**: This is an open-source platform for automating the development and deployment of machine learning pipelines on Kubernetes. It is designed to be flexible and scalable and to allow you to easily build, manage, and monitor machine learning workflows.
2. **Apache Airflow**: Apache Airflow is an open-source platform for managing and scheduling workflows. It is designed to be flexible and extensible and can be used to manage a wide range of workflows, including those for machine learning.
3. **Kubeflow**: This open-source platform is designed to help with the development and deployment of machine learning pipelines on Kubernetes. It includes tools for building, managing, and deploying machine learning pipelines, as well as for monitoring their performance.

Learning Resources:

- **Kubeflow**: Building a Machine Learning Pipeline with Kubeflow | Full Walk-through

- **Apache Airflow**: Airflow Tutorial for Beginners—Full Course in 2 Hours
- **Argo**: ArgoCD Tutorial for Beginners | GitOps CD for Kubernetes

12.6. Machine Learning Security & Compliance Tools

The final point in this roadmap is to learn machine learning security and compliance tools. This involves knowing how to secure sensitive data, maintain data privacy, and comply with legislation and standards. Here are some popular tools:

1. HashiCorp Vault: This is a tool for securely storing and managing secrets, such as passwords, API keys, and certificates. It is designed to be highly secure and to provide a central location for storing secrets that can be easily accessed by applications and users.
2. AWS GuardDuty: This cloud-based threat detection service from Amazon Web Services (AWS) uses machine learning to identify potential security threats to your machine learning systems. It includes features such as real-time monitoring and automatic alerting to help you respond to potential threats.

Learning Resources:

- HashiCorp Vault: HashiCorp Vault Certification
- AWS GuardDuty: AWS GuardDuty Crash Course

12.7. Putting it into Action

Now that you've explored the core fundamentals of MLOps, the next step is to put your learning into action. MLOps is not just about understanding the tools—it's about integrating them into

real-world workflows, automating machine learning lifecycles, and ensuring models are reliable in production. Here's a practical approach to reinforce your knowledge:

1. Build an End-to-End MLOps Pipeline

The best way to apply what you've learned is by designing and implementing an end-to-end MLOps pipeline. This involves:

- **Version Control**: Use Git to track your ML code and data changes.
- **Automate Training & Deployment**: Set up a CI/CD pipeline to train and deploy models automatically.
- **Infrastructure as Code (IaC)**: Use Terraform or Kubernetes to manage resources dynamically.
- **Monitoring & Observability**: Implement logging, metrics, and alerts to monitor model drift and performance.
- **Security & Compliance**: Ensure your workflow adheres to best practices in security and data governance.

2. Select a Practical Project

To solidify your skills, work on a hands-on project that involves real-world challenges. Here are a few project ideas:

Project 1: MLOps Pipeline for a Fraud Detection Model

- Train a machine learning model for detecting fraudulent transactions.
- Automate data ingestion, preprocessing, and training using Airflow.
- Deploy the model as a REST API with FastAPI or Flask.
- Set up continuous monitoring to detect model degradation.
- Implement drift detection and auto-retraining triggers.

Project 2: Scalable MLOps for NLP Sentiment Analysis

- Fine-tune a transformer-based NLP model for sentiment classification.
- Deploy the model on AWS Lambda or Kubernetes for scalable inference.
- Use MLflow for experiment tracking and model registry.
- Automate testing with unit and integration tests in CI/CD.
- Implement monitoring with Prometheus and Grafana.

Project 3: Computer Vision Model with Continuous Deployment

- Train an image classification model using TensorFlow or PyTorch.
- Deploy the model using Docker and Kubernetes.
- Implement a model shadow deployment strategy to test updates before full rollout.
- Use Grafana to visualize model performance metrics in real-time.
- Set up automatic retraining based on data drift detection.

13. Building Your Data Science Portfolio

In today's job market, having a strong portfolio of data science projects is crucial for landing a job as a data scientist or data analyst. Companies are looking for candidates who can demonstrate their knowledge of data science tools and techniques, as well as their ability to use them to solve real-world problems they are expected to face while working there.

In this chapter, we will discuss the key components of a successful data science portfolio project and provide tips and best practices for building one that will make you stand out to potential employers. Whether you're just getting started in the field or looking to take your career to the next level, this guide will help you create a portfolio that showcases your skills and experience in the best possible light.

13.1. Importance of Having a Data Science Portfolio Project

Having a portfolio project will be a game-changer in your job search. It demonstrates your skills and experience to potential employers or clients. It also provides a way to showcase the projects you've worked on and the results you've achieved, which can be far more effective than simply listing your skills on a resume.

In addition to that, building a portfolio project can also help you improve your hands-on skills and stay current in the field. By working on a variety of projects and tackling new challenges, you can continue to learn and grow as a data scientist.
Finally, you will build a self-brand by publishing this project on different social media channels, which will get you more opportunities and will expand your network.

> ***Building a portfolio of projects, especially one that shows progress over time from simple to complex undertakings, will be a big help when it comes to looking for a job.***
> - Andrew NG -

13.2. Select a Domain of Interest

The first step to building a strong data science portfolio is to focus on a certain domain of interest. Data science and AI, at the end of the day, are tools that are used to solve a problem, improve performance, or automate a certain task. Therefore, it is important to decide which domain you would like to apply your data science skills.

This might depend on your previous experience. If you have working experience in a certain domain, it will be easier to find business problems to work on. It can also be a domain you are interested in and would like to use your skills to make an impact in this domain.

It is important to mention that the more you have experience in this domain, the more you will be able to get unique ideas, and the better your projects will be. In addition to that, it will give you a great advantage in the market and make you stand out. You will have a very good understanding of the data collection process and what it means, and it will also improve your skills in engineering the features of the data.

Actions:

- Choose three domains of interest depending on your experience and research background.
- You should take into consideration your career goal and whether you would like to work in research or in industry.
- You can find more about different domains and how data science and AI are used to solve business problems in this article: **Requirements: Domain Emphases**

13.3. Prioritize Your Interest Based on the Market Demand

The next step after selecting two or three domains of interest is to prioritize and arrange them based on the market needs and demands. For example, I am living in Finland, so if I were to

join this market, I would have to do some market research to understand the market demand and know the companies working in these domains.
Based on my market research, I found that there are a lot of opportunities and demands in the telecommunication domain (Nokia, Elisa, DNA), gaming domain(Unity, Rovio), and the Fintech industry. So I should prioritize my interests based on this when I start to build an end to end project.
Also, currently, Generative AI and LLMs are very active areas of research and a hot topic in the industry. So you can build projects in this AI domain and focus on the overlap between your domain of interest, generative AI, and the market needs.

Actions:

- Select the market you would like to work in.
- Do market research and find the intersection between your interest and the market need.
- Select one domain that meets the previous criteria
- Select the companies that are working in this intersection and are regularly hiring.

13.4. Define Important Case Studies In the Market

Now you have selected the domain of interest that meets your interest and the market demand, and you have good ideas about the companies that are hiring in this domain and the job

requirements. It is time to define the case study or business problem to start building a solution to solve it or answer the business requirements.
To come up with real-world business problems or questions in this domain, you can do further research about the companies above and know what they are working on and what they use data science for.
This can sometimes be found in the job requirements or on the company website. However, if you can not find it there, you can take a further step and start asking data scientists who are working there.
I believe this is a very useful and goal-oriented approach because not only will you be working on similar problems as your dream company is working on, but it will make it easier to get a job offer there.
You will also get a good idea about the tools and the technology stack at these companies, so you can learn them and use them in creating your projects.
Another approach is to interview experts from this domain and ask them about the most important questions or problems they have, and they wish they could use the data to solve them.

Actions:

- Read the recent data science job requirements for the companies you are interested in or read about the projects these companies are currently working on.
- If you did not find much information on the job requirements or the company website, you can contact data scientists who are working there.
- Find three case studies or business problems where they are using data science to solve them.
- Repeat this for different companies you are interested in working there.

13.5. Choose Different Case studies

Now you have multiple case studies to work on in your domain of interest. It is time now to narrow down your selection. My suggestion is to have at least three solid case studies that cover the basic machine learning tasks, which are regression, classification, and clustering.

In addition to that, try to focus on having them solved using different data types, and you narrow your selection and focus more on the data types you are passionate about. For example, if you would like to show your computer vision skills, you can focus more on case studies that need computer vision skills.

To elaborate more, let’s take a practical example. Let’s assume that you are interested in working in the healthcare domain. So after searching, you came up with multiple case studies in this field, and you are interested in the computer vision domain. So here are three case studies that cover different machine learning tasks and are solved using computer vision skills:

- **Regression**: Movement disorder detection using pose estimation.

- **Classification**: Brain tumor detection and classification
- **Clustering**: Alzheimer's disease analysts using clustering

You can also work on generative AI and LLM-based projects and applications. I highly recommend exploring projects in the following areas:

- **Prompt Engineering**: Experiment with designing and optimizing prompts to improve LLM outputs for various tasks like text generation, summarization, and reasoning.
- **RAG Pipeline**: Build Retrieval-Augmented Generation (RAG) systems that combine LLMs with external knowledge sources, such as vector databases, to enhance response accuracy.
- **Agentic Workflow**: Develop autonomous AI agents that can plan, execute, and refine tasks dynamically using tools like LangChain and LangGraph or crewai.
- **LLM Fine-Tuning**: Customize large language models for specific domains by fine-tuning specialized datasets to improve performance and relevance.

These areas provide a solid foundation for mastering LLM applications and building real-world AI solutions.

Actions:

- Discover the different areas of AI and know your interests.
- Narrow down your case studies into at least three that cover the basic machine learning tasks and focus on your area of interest in AI.
- Define which case study is solved by which machine learning task and which data type.

13.6. Brainstorm Data Science Solutions

Now you have defined the case study you would like to work in and defined the business questions for this problem. Do not rush into building the first solution that comes into your mind. Instead, take your time to brainstorm different potential solutions for it, study each of them, and see which one will lead to better results and meet your learning goals and skill set.

After brainstorming the solutions and choosing the suitable one you will need to assess the feasibility and value of potential solutions. This can be done by reviewing the published research papers on this topic or you can discuss with an expert your potential solution and see whether it is reasonable and will achieve the expected results or not. This is also a very critical step as it will save you a lot of time, effort, and future disappointments.

Actions:

- Brainstorm different AI& data science solutions for your business problem
- Evaluate them based on your criteria and select the one that best meets them.
- Validate your potential solution by discussing it with experts.

13.7. Determine Success Metrics

Once you have brainstormed the potential solutions and validated their feasibility and value, it is time to determine the success metrics you aim for for this project and solution.
This includes both machine learning metrics, or what is known as offline metrics, such as (accuracy and F1 score) and business metrics, or what is known as online metrics (revenue, click-through rate).
Machine learning teams are often most comfortable with metrics that a learning algorithm can optimize. However, we may need to stretch outside our comfort zone to come up with business metrics, such as those related to user engagement, revenue, and so on.
Unfortunately, not every business problem can be reduced to optimizing test set accuracy! If you aren't able to determine reasonable milestones, it may be a sign that you need to learn more about the problem.

Actions:

- Study the problem and define the success metrics for your project, both the machine learning and the business metrics.

13.8. Collect the Dataset

Now that your idea is ready, it is up to me to get your hands dirty with data. You need to collect real-world data to answer your business questions or to train the models. It is very important to use a unique dataset that is representative of your problem.
Kindly stay away from well-known datasets such as the Titanic, California house prices, Iris flowers dataset, and similar well-known datasets. They are very good for beginners and for educational projects, but they will harm you if your portfolio projects are with them.

Here are some suggested ways to collect unique datasets to develop your solution based on them:

- Kaggle
- Hugging Face Datahub
- Scrape your own data
- Ask for data
- Use open datasets from universities, NGO organizations, or governmental organizations.

Actions:

- Search for different data sources that can provide you with a unique dataset that can fit your project
- Collect and store the data

13.9. Clean & Prepare the Data

Now you have the data. The next step is to make it ready for machine learning modeling. This includes cleaning the data and applying different feature engineering techniques to it to get the best out of it.

This step is very demanding, especially if your data is real-world data, which might have a lot of missing data, outliers, and other defects that are very common in real-world data.

In addition to that, you will have to explore the data to have a better understanding of it and to guide you when you start to engineer its features and make it ready for the modeling step. Feature engineering will include data preprocessing, feature selection,n dimensionality reduction, and more, depending on your problem and the collected data.

Actions:

- Clean the data
- Explore the data
- Feature engineering to make it ready for modeling

13.10. Train & Evaluate the Model

Now it is time to train the machine learning model using your data. This will include several steps. First, you have to choose the models to use. This will depend on many factors, such as:

- Explainability
- In Memory vs. Out Memory
- Number of features and instances
- Categorical vs. numerical features
- Data normality
- Training speed
- Prediction speed

Next, you will train and evaluate the model. This step includes splitting the data, training the model, choosing suitable and representative evaluation metrics, and hyperparameter optimization.

Actions:

- Select suitable models for your problem
- Split the data
- Train the model
- Choose suitable evaluation metrics
- Optimize the model hyperparameters
- Test your model on the testing data

13.11. Make them end-to-end

The final step is to make your project an end-to-end project. Many people usually stop at the previous. By this, you miss a big chance of making your project a real project, which will help you in your job searching journey and make you stand out from other candidates.

To make your project an end-to-end project, you will need to take your model a further step, deep into production, and integrate it into a web or mobile application. After that, you will start to monitor the model and see how it responds to new data. Based on the model's performance in production, you will have to retrain the model, change it, collect more data, engineer the features in a different and so on.

Actions:

- Deploy the trained model into production
- Integrate the model into a mobile or a web application
- Monitor the model performance
- Iterate

12. Publish & Talk About It

The final step is to publish your project on your GitHub page and create a comprehensive README file for it. Your readme file should contain this:

- Motivation & business problem statement
- How the data was collected
- Main data cleaning steps
- Comprehensive data exploration plots
- Main feature engineering steps
- The model used and why you chose it
- Evaluation metrics and why you chose it
- The model performance
- How to try the model in production

Having a comprehensive readme file for your project is a very important step that a lot of people actually do not focus on. It will make your work and project more valuable and accessible since many people will only read the readme before deciding to go through the code. It also shows your documentation and insight communication skills, which is a critical skills for aspiring data scientists.

Finally, you need to start talking about your project to grab the attention and to show the people what you are capable of. You can record a short video of your project while working in real time and publish it on your social media channels, especially LinkedIn & Twitter, and invite your connections to try it and give feedback on this experience.

You can also write a blog explaining each step and showing the insights you got from the data. You can create a YouTube video explaining the project steps and also show the results and the insights you got from the data, and how you answered the business questions, and how the model works in production.

Actions:

- Upload your project to GitHub & publish it on your professional social media channel
- Push the model to Hugging Face
- Write a comprehensive readme file for your project
- Record a short video of your project to demonstrate how it works
- Invite people to try your project and give you feedback
- Write a blog about your project
- Record a long video explaining the project steps

14. Getting Ready for the Market

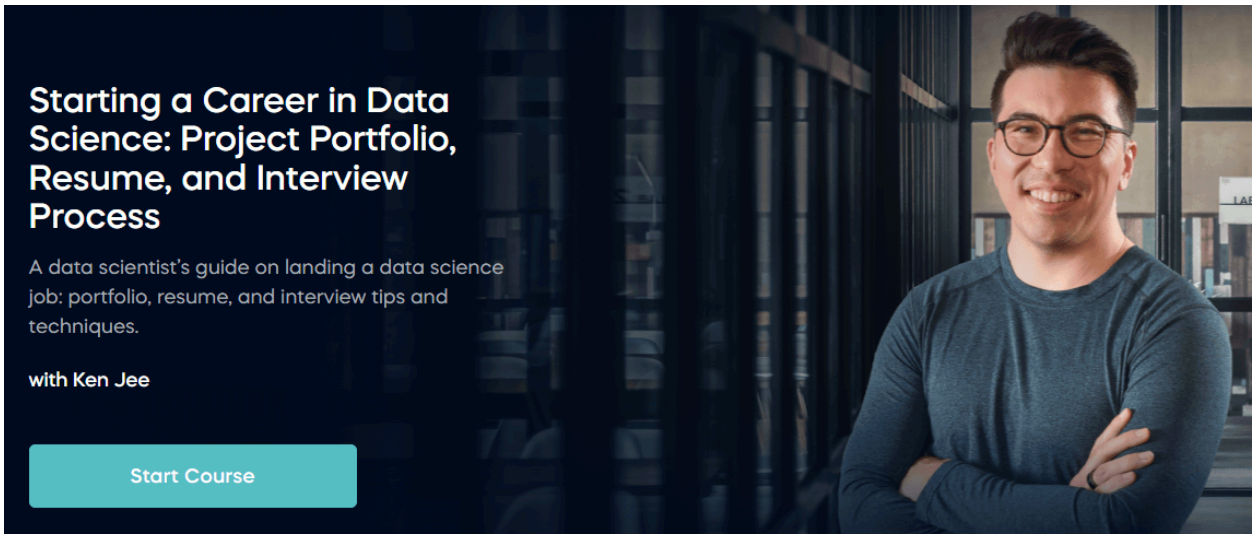
Data science & AI are one of the most in-demand fields, with opportunities available across a range of industries. However, landing a data science role can be a competitive and challenging process, particularly if you are a beginner or have limited experience in the field.

This chapter, the last in our Series "A Beginner-to-Upper Intermediate Data Science Roadmap for 2025," provides you with resources to help you get and prepare for your first data science interview. It outlines four essential resources that you can use to prepare your resume and portfolio, gain knowledge and skills, as well as tips and tricks to help you ace the interview and stand out from the competition.

Whether you are a student or an aspiring data scientist, these resources will provide you with the tools you need to succeed in your data science job search.

14.1. Starting a Career in Data Science: Project Portfolio, Resume, and Interview Process—Course

The first resource is the Starting a Career in Data Science: Project Portfolio, Resume, and Interview Process course by 365 Data Science. This course is taught by Ken Jee. This course provides you with invaluable insights directly from a top-level data scientist with first-hand experience in recruiting skilled individuals.



You will learn everything you need in order to get a competitive edge over other job candidates and start a career in data science:

- How to create your data science project portfolio
- How to build your resume professionally
- How to get an interview through networking

- How to succeed during the phone interview
- How to solve the take-home test
- How to ace the behavioral and technical questions.

The course also offers you resume templates, downloadable materials, helpful infographics, as well as a section on how to optimize your LinkedIn, Github, and Kaggle profiles for recruitment purposes.

I really recommend this course if you have just finished your learning plan, and done enough data science projects, and you are in the process of finding a job. I truly believe this course will give you a huge advantage during this process and will decrease the time you will take to land your first job.

14.2. Data Science Interview Pro—YouTube Channel

The second resource is the **Data Science Interview Pro YouTube channel** by **Emma Ding**. Emma is an experienced data scientist who worked at Airbnb and founder of Data Interview Pro, a website that empowers Data Scientists to land their dream jobs. On her YouTube channel, she provides a lot of valuable information on getting and passing your data science interviews.

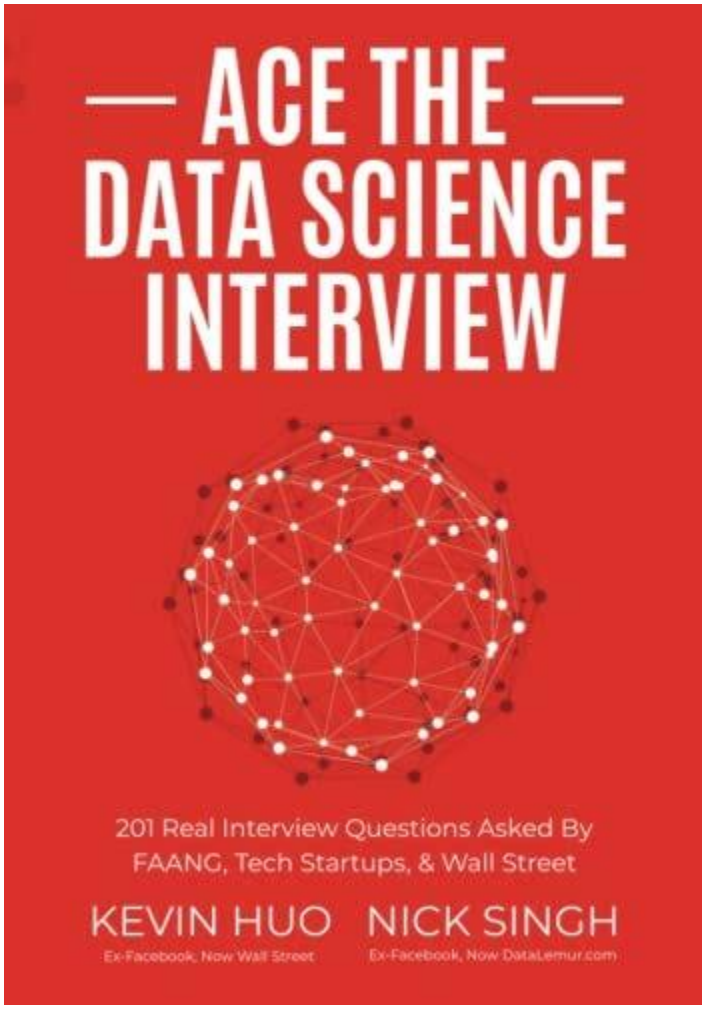
I found her YouTube channel really helpful in knowing what to expect in each type of data science interview in the hiring process. In addition to that, she also provides valuable information on what to expect in technical interviews. In addition to that,t she really provides a comprehensive review of each of the important technical topics and how to pass the interview questions on these topics:

- Machine Learning
- Python
- SQL
- Statistics
- Product case & metrics

I really recommend her channel to refresh your mind with the important concepts before your interviews, and also to know what to expect in each step of the hiring process,s and to know how to pass it.

14.3. Ace the Data Science Interview—Book

The third resource on this shortlist is Ace the Data Science Interview: 201 Real Interview Questions Asked By FAANG, Tech Startups, & Wall Street by Nick Singh (Author), and Kevin Huo (Author). This book is authored by two ex-Facebook employees. Ace the Data Science Interview is the best way to prepare for Data Science, Data Analyst, and Machine Learning interviews, so that you can land your dream job at FAANG, tech startups, or Wall Street.



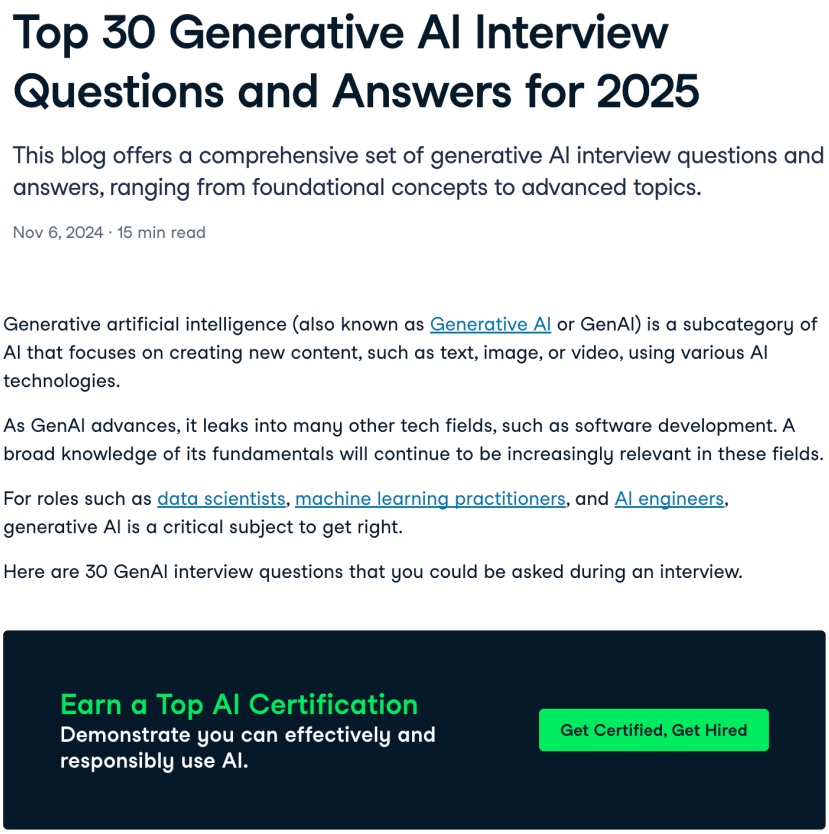
The content of the book can be summarized in the following points:

- 201 real Data Science interview questions asked by **Facebook**, **Google**, **Amazon**, **Netflix**, **Two Sigma, Citadel,** and more—with detailed step-by-step solutions!
- Learn how to break into Data Science, with tips on crafting your **resume**, creating kick-ass **portfolio projects**, sending networking **cold emails**, and better telling your story during **behavioral interviews**

- Questions cover the most frequently-tested topics in data interviews: **Probability**, **Statistics**, **Machine Learning**, **SQL & Database Design**, **Coding (Python)**, **Product Analytics**, and **A/B Testing**
- Each chapter has a brief crash course on the most important concepts and formulas to review
- Learn how to solve open-ended case study questions that combine product sense, business intuition, and statistical modeling skills, and practice with case interviews from **Airbnb, Instagram,** & **Accenture**

I believe this book is a great resource to review the common questions that you will probably meet in your data science interview and have answered in a perfect way.

14.4. Top 30 Generative AI Interview Questions and Answers for 2025



Top 30 Generative AI Interview Questions and Answers for 2025

This blog offers a comprehensive set of generative AI interview questions and answers, ranging from foundational concepts to advanced topics.

Nov 6, 2024 · 15 min read

Generative artificial intelligence (also known as Generative AI or GenAI) is a subcategory of AI that focuses on creating new content, such as text, image, or video, using various AI technologies.

As GenAI advances, it leaks into many other tech fields, such as software development. A broad knowledge of its fundamentals will continue to be increasingly relevant in these fields.

For roles such as data scientists, machine learning practitioners, and AI engineers, generative AI is a critical subject to get right.

Here are 30 GenAI interview questions that you could be asked during an interview.

Generative artificial intelligence is a subcategory of AI that focuses on creating new content, such as text, image, or video, using various AI technologies.

As GenAI advances, it leaks into many other tech fields, such as software development. A broad knowledge of its fundamentals will continue to be increasingly relevant in these fields.

For roles such as data scientists, machine learning practitioners, and AI engineers, generative AI is a critical subject to get right.

This **article** provides you with 30 GenAI interview questions that you could be asked during an interview.

This is my short list of resources that I believe can have a game-changing role in your job-searching journey. I hope you find them beneficial and can help you with the frustrating process of landing a data science job. If you know similar resources, kindly share them with us in the comments section.

Afterword

Thanks for purchasing and reading my book! If you have any questions, feedback or praise, you can reach me at: **Youssef.Hosni95@outlook.com**

You can check my other books on my **website**. I would be happy if you connect with me personally on **LinkedIn**. If you liked my writings, make sure to follow me on **Medium**. You are also welcomed to subscribe to my **newsletter To Data & Beyond** to never miss any of my writings.

What's inside the book?

- Introduction to Data Science & Data Methodology
- Mathematics for Data Science
- Python Fundamentals
- Python for Data Science
- Software Engineering Basics
- Database & SQL Fundamentals
- Feature Engineering
- Mastering Machine Learning
- Deep Learning Fundamentals
- Generative AI & Large Language Models (LLMs) Fundamentals
- Machine Learning Operations (MLOps)
- Building Your Data Science Portfolio
- Getting Ready for the Market

About the Author

Youssef Hosni is a data scientist and machine learning researcher who has worked in machine learning and AI for over half a decade.
In addition to being a researcher and data science practitioner, Youssef has a strong passion for education. He is known for his leading data science and AI blog, newsletter, and eBooks on data science and machine learning.

Youssef is a senior data scientist at Ment focusing on building Generative AI features for Ment Products. He is also an AI applied researcher at Aalto University working on AI agents and their applications. Before that, he worked as a researcher in which he applied deep learning and computer vision techniques to medical images.