Step-by-step roadmap to becoming a data scientist

by Timur Bikmukhametov

Table of Contents

Motivation

Learning data science was never easy. In 2015, when all the buzz started, there was not much material except books and some occasional videos. Now, the situation is the opposite, there is SO MUCH material, so it is hard to choose where you should go.

The goal of the roadmap

is to provide a list of resources that are enough to be a solid junior+/middle data scientist starting from zero. Some parts will be useful even for seniors/leads (if you are a senior and you think you know everything, let me hug you) but this is not the main purpose.

This roadmap is for:

- Data Science beginners who are looking for a practical step-by-step guide.
- Data Scientists who aim to level up skills for a job change or promotion.
- Data Scientists who are looking to refresh their knowledge and prepare for interviews.
- Data Scientists who want to level up skills in a specific domain, e.g. Optimization.

READ THIS BEFORE YOU START

You will never learn enough to be "ready" because Python, machine learning, modeling, and optimization are enormous technical fields. This roadmap will give you the pillars you can use as you grow.

As such, I recommend:

- **If you are a complete beginner,** you cover Python and Introduction to DS/ML blocks and start building a project you are interested in. In parallel, dedicate time to learning things from this roadmap as you go, I guarantee it will help.
- **If you have been working in or studying data science,** choose topics that you feel you need to learn more about. And also build portfolio projects or do some meaningful stuff at work along the way.

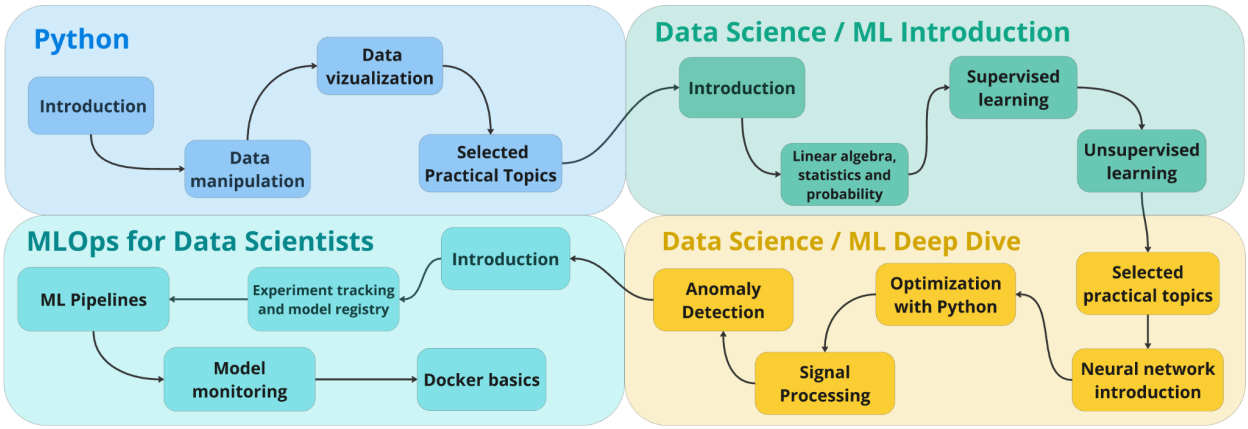
Disclaimers

I have no what-so-over affiliation with any recommended courses. All the materials are advised based on personal experience, the experience of colleagues and friends, and my personal opinion on the content of the courses/books/articles.

Another disclaimer is that you can complete the entire roadmap for absolutely free, however, for some topics, I do recommend spending some money on the paid alternatives that I provide. If you can afford it, please, do that, for a better learning experience.

There are a couple of Coursera courses here. For those, you can ask for financial aid on Coursera and they will provide it, I did that several times back in my student days.

Roadmap Overview



Ok, enough talking. This is the roadmap overview. We will be breaking down each part in detail below.

You will learn things step-by-step, slowly increasing the importance and the difficulty of the topics.

The modules from Python to MLOps are generic and applicable to any classical data science topics with some modules covering NLP and Computer Vision.

I recommend that you compute the first 2 modules (Python and ML intro) and slowly start building your own small project. Then grow this project as you go through the ML deep dive and MLOps modules.

1. Python

Life is too short, learn Python. Forget R or S or T or whatever other programming language letters you see. And for God's sake, no Matlab in your life should exist.

Ok, fine, but how to start?

Step 1. Remember that you will never know Python perfectly. Just admit it, move on, and do not make imposter syndrome hit you.

Step 2. We go to the point.

1.1 Introduction

Here, I propose 2 versions: free and paid. If you ask me, I would go for the paid version because I think those courses are a bit better structured and easier to start with. They have a built-in Python editor, so do not even need to install Python. Pure learning. This is especially useful if are a complete beginner.

However, if you are tight on money, feel free to start with free courses, they are also great and easy to follow.

Paid version:

Course 1: Basic Python from CodeAcademy

Link: https://www.codecademy.com/learn/learn-python-3

This is a perfect intro to Python. I sucked at programming a lot when I was starting but this course took me through. Look no further for the start, just start with this.

Course 2: Python Programming Skill Track from DataCamp

Link: https://app.datacamp.com/learn/skill-tracks/python-programming

This course goes deeper into software engineering and coding principles with Python examples. It covers a useful topic on efficient code in Python in general and in Pandas specifically which is useful for data scientists. Finally, it considers test-driven development which you will need if you write a robust machine learning product.

The course will NOT teach you everything you need (none does) but it should give you a good start and a basic overview of the main principles you will use in your projects.

Free version:

Course 1: Course from futurecoder.io

Link: https://futurecoder.io/

This course is similar to the CodeAcademy course from the paid version and even has a similar online programming editor. This is a great start for a complete beginner.

Course 2: Course from Dave Gray

Link: https://www.youtube.com/watch?v=qwAFL1597eM

Out of tens of free video courses, I found this one to be well-explained with clear language. There will be some reparations from Course 1 above, but this is a good thing.

Course 3: Mini-project course from freeCodeCamp

Link: https://www.youtube.com/watch?v=8ext9G7xspg

I would suggest that if you are a beginner, you spend some time and do several projects from this course to get more confident with the language. The explanations are very clear and I believe you will get more confident before moving to using Python for data analysis.

1.2 Data manipulation

Now that you are familiar with Python, we can go to what data scientists use it for.

Step 1: Kaggle Pandas course with exercises

Link: https://www.kaggle.com/learn/pandas

This is a great start with Pandas. Go cell by cell of the notebooks and **I HIGHLY RECOMMEND** (in fact, I insist), complete exercises.

Step 2: Data manipulation topic in awesome mlcourse.ai by Yury Kashnitsky and Co.

Link: https://mlcourse.ai/book/topic01/topic01_intro.html

Another perfect intro Pandas material. Overall, mlcourse.ai is a great data science course and if you are eager **(you should be!)**, complete the entire course. But you will see the selected topics from it along the roadmap.

Step 3: Data manipulation with Numpy

Link: https://github.com/ageron/handson-ml2/blob/master/tools_numpy.ipynb

Another essential library to know is Numpy which has a lot of capabilities, so it is hard to fully know it. But this tutorial gives a great overview of the most essential Numpy methods.

Step 4 (optional): Pandas exercises repo by Guilherme Samora

Link: https://github.com/guipsamora/pandas_exercises

If still in doubt about Pandas, there is an awesome resource with many Pandas exercises. I would skip it but I am sure some beginners would love to practice more. So, there is no need to look where to do that, enjoy this one.

1.3 Data visualization

Data visualization is a critical part of data scientist work, so you must know how to make good plots to tell a story. There are plenty of tools, some of them are generic and some of them are domain-specific. I recommend learning the generic ones and then you can learn specific ones if you need them.

Intro

Data visualization topic in mlcourse.ai:

https://mlcourse.ai/book/topic02/topic02_intro.html

This course is a nice start before going into details of each library.

Deeper dive

Below is the list of resources for each library. You can go through the list of plots to understand what kind of plots you can make in general and try several of them. Then, when doing the projects, always go back there and check how to make what you want.

Matplotlib

- **Examples gallery:** https://matplotlib.org/stable/gallery/index.html
- **List of notebooks with examples:**

  https://github.com/javedali99/python-data-visualization/tree/main/Matplotlib

Seaborn

- **Example gallery:** https://seaborn.pydata.org/examples/index.html
- **List of notebooks with examples:**

  https://github.com/javedali99/python-data-visualization/tree/main/Seaborn

Plotly

Plotly is good when you need interactive plots. For me, it was useful when I had to go through multivariate time series data, so I could just scroll the plots instead of re-plotting

them when I needed to see a new timeframe of data. Also, Plotly is a core of Dash that you might want to use when creating your project portfolio.

- **Example gallery:** https://plotly.com/python/
- **List of notebooks with examples:**

  https://github.com/javedali99/python-data-visualization/tree/main/Plotly

1.4 Selected Practical Topics

Learn this as you go.

You got some Python exposure. Now, let's see what else is important to learn as you go. You do not necessarily need to go through it at the beginning but I recommend you to come back when you start working on your portfolio or work projects.

Topic 1: Python environments and how to set it up with Conda

Link:
https://whiteboxml.com/blog/the-definitive-guide-to-python-virtual-environments-with-conda

You need to know what an environment is and how to work with it. There is an endless debate about which environment manager is the best. I recommend starting with Conda and changing later if required.

Topic 2: Demystifying methods in Python

Link: https://realpython.com/instance-class-and-static-methods-demystified/

I highly recommend looking at this article because you will often see and use different Python methods and this article gives an excellent explanation.

Topic 3: Python clean code tips and formatting.

- **Clean code tips, part 1:** https://github.com/zedr/clean-code-python

This is a perfect summary of the great book by Robert Martin on how to write clean code. I have not gone through everything myself, but I will, I promise. **I highly recommend to read and apply it as you go.**

- **Clean code tips, part 2:** https://realpython.com/python-pep8/

A shorter version on a similar topic. By the way, RealPython is always a great resource, their article quality is very high. As long it is RealPython material, I always consider it as best practice.

- **Formatting code with black:**

  https://www.python-engineer.com/posts/black-code-formatter/

Black makes it easy for you to format your code following PEP principles, so it is a great tool to use in your projects.

- **Code linting with flake8 and pylint:**

  https://www.jumpingrivers.com/blog/python-linting-guide/

Linting helps to improve the code readability and make it less buggy, so applying this in your projects will help you to make your code more robust.

Topic 4: Python imports

Link: https://realpython.com/python-import/

For me, at the beginning imports across a project bigger than 1 file was confusing and I believe going through this article would help you to understand how you can organize your Python project and import modules across it.

Topic 5: Python decorators

Link: https://realpython.com/primer-on-python-decorators/

When you start looking at open-source code libraries, you start seeing weird things right before functions or class methods starting with “@”. This is called a decorator, a concept widely used in Python. I bet that you will not use it often, but it is handy to get familiar with it to understand others’ code.

2. Data Science / ML introduction

Now we are getting to the “most interesting“ part.

My opinion is that **you need to know the basics well** to be a solid data scientist. It does not mean to be a nerd, but a good understanding of the main principles will help you in both the job and succeeding in an interview.

In the roadmap, I suggest you get to know only the most often used algorithms but you have to know them very well. Using this knowledge, you can then proceed with other algorithms.

Now, let's go.

2.1. Introduction

Andrew Ng course:

https://www.coursera.org/learn/machine-learning?specialization=machine-learning-introduction

This is a perfect course to get an overview of what machine learning is and what are the two most common problems that are solved by ML: regression and classification. Do not go over the tons of other intro courses, take this.

Note: *by default, Coursera is not free but you can ask for financial aid and they will give you that after consideration. I did that several times back in my student days.*

2.2. Basic probability, statistics, and linear algebra

Each of these fields is massive. So massive that you will never end up learning. In practice, you do not need to know them very deeply, but the main concepts you must know very well.

Linear algebra

Step 1: Videos of 3blue1brown about linear algebra

Link: https://www.3blue1brown.com/topics/linear-algebra?ref=mrdbourke.com

The world is blessed to have 3blue1brown. I almost cried when I watched this video series. This will blow your mind, but most importantly give a good understanding of the main concepts.

Step 2: Tutorial of Python Linear Algebra by Pablo Caceres

Link: https://pabloinsente.github.io/intro-linear-algebra

Another unbelievable resource. Here, you get familiar with how the matrix operations look in Python which is essential if you wanna use book knowledge in practice.

Probability and Statistics

Step 1: Statistics Crash Course by Adriene Hill

Link:
https://www.youtube.com/playlist?list=PL8dPuuaLjXtNM_Y-bUAhblSAdWRnmBUcr

Oh, man, if only I had seen this course when I was learning it. She explains complex concepts in a very easy way, so you can relate the main statistical and probability concepts using the world language and examples. Simply beautiful.

Step 2: Learn Statistics with Python by Ethan Weed

Link: https://ethanweed.github.io/pythonbook/landingpage.html

Now, to get back to practice, this resource makes it simple to apply the learned concept in Python while making knowledge more solid along the way.

2.3 Supervised learning

There are a massive amount of algorithms but you barely use even 20% of them. I propose you learn the following list and then proceed with the rest using the knowledge you get.

There will be some intersections with Andrew Ng's course, but it would not hurt to go a bit deeper and have different implementations and perspectives on the same material.

Linear regression

Intro theory: Nando de Freitas lectures at UBC

- Lecture 1
- Lecture 2

I am a huge fan of Nando de Freitas. I had so many "Aha!" moments watching his lectures that he couldn't imagine. You can watch the whole series and enjoy but here we will pick only some of them.

Python Implementation

- Closed-form linear regression implementation
- Linear regression with gradient descent

These are some great visualizations and Python implementations of the algorithm.

Regularization in linear regression

Regularization is an essential concept to understand and with linear models, you can do it easier. There will be a lot of questions in interviews about it, so make sure you know them.

Step 1: Nando de Freitas lectures at UBC

- Lecture 1
- Lecture 2

Step 2: Visual explanation with code

- Explanation 1
- Explanation 2

Sklearn tutorial with Lasso model

Link:
https://www.kirenz.com/blog/posts/2019-08-12-python-lasso-regression-auto/#lasso-regression-in-python

At the end of the day, you will be using the sklearn implementation of the linear model, so this tutorial will give a good overview of how to do that and how to use the LASSO model effectively.

Logistic regression

Logistic regression is a baseline algorithm for classification tasks. As it is highly related to the linear regression model, you do not need to learn it from scratch but it is important to understand some important concepts about it.

Intro: Logistic regression topic of mlcourse.ai

Link to the topic

Another great work by the mlcourse.ai team. Together with the previous material, you will get everything you need about logistic regression.

Selected topic: odds ratio as weights interpretability

Link: https://mmuratarat.github.io/2019-09-05/odds-ratio-logistic-regression

A deeper overview of the odds ratio that you will always be asked in interviews.

Gradient boosting

This one you have to know by heart, I'm sorry. I give you some good resources to start.

Step 1: Gradient Boosting topic of mlcourse.ai

Link: https://mlcourse.ai/book/topic10/topic10_gradient_boosting.html

I think this is the best topic in the mlcourse.ai. They made it perfect and for a reason.

Step 2: Gradient Boosting, deeper dive

- Tutorial by Alexey Natekin
- XGBoost original paper

I personally learned a lot from the original XGBoost paper, but Natekin's paper is very detailed and always great to come back to when you forget things.

Step 3: Demo playground by Alex Rogozhnikov

- Demo 1
- Demo 2

Another genius made a great visualization for us, normal people. By the way, check out his entire blog. It is simply amazing. When get you serious about ML and start learning about Markov Chain Monte Carlo, check it out again. I wish I could be as smart as Alex.

Random Forest

Another crucial algorithm to know by heart. Please, understand the difference between Random Forest and Gradient Boosting, I bet you get this question in 30-40% of the interviews.

Step 1: Lectures by Nando de Freitas

- Lecture 1
- Lecture 2
- Lecture 3

Step 2: Bagging topic on mlcourse.ai

Link: https://mlcourse.ai/book/topic05/topic05_intro.html

k Nearest Neighbours (k-NN)

This algorithm is not very often used but I would encourage you to get to know it because this is a distance-based algorithm that is a bit different from all the above and appears here and there.

Link: https://mmuratarat.github.io/2019-07-12/k-nn-from-scratch

2.4 Unsupervised learning

You have grown up, my friend. You are ready to know how to learn things from data without knowing what is the true label/value. Let's see how.

Clustering

k-Means

This is the default method, you have to know it. Here is a great tutorial from Mustafa Murat ARAT (check out his other tutorials instead of scrolling Instagram)

Link: https://mmuratarat.github.io/2019-07-23/kmeans_from_scratch

DBScan

Another popular method that is useful to have in your arsenal.

Link:
https://github.com/christianversloot/machine-learning-articles/blob/main/performing-dbscan-clustering-with-python-and-scikit-learn.md

Dimensionality reduction

PCA

PCA will be all over. A very useful concept for many applications.

Material from the one and famous Sebastian Rashka:

https://sebastianraschka.com/Articles/2014_pca_step_by_step.html

t-SNE

You never use it but everyone talks about it (just kidding). I used it a couple of times successfully for EDA.

- What is it and how to run it in Python
- How to use t-SNE effectively (with great visualizations)

UMAP

Another great algorithm that you can use to reduce data dimensions.

Link: https://pair-code.github.io/understanding-umap/

3. Data Science / ML Deep dive

3.1 Selected Practical Topics

Feature selection

Link:
https://www.kaggle.com/code/prashant111/comprehensive-guide-on-feature-selection/notebook#Table-of-Contents

Feature selection is one of the most important topics when you really want to improve your model, make it more transparent, and understand the WHYs behind the predictions.

In most of the cases, feature selection depends on the domain knowledge, however, you still have to use some of the methods available. The source above gives a solid overview of them.

Feature importance

Feature importance is another crucial thing you have to know. Here are the sources for different methods.

Linear methods: Chapter 5 of Interpretable Machine Learning book

Link 1: https://christophm.github.io/interpretable-ml-book/limo.html

Link 2: https://christophm.github.io/interpretable-ml-book/logistic.html

Tree-based methods: Youtube Raschka lecture

Link: https://www.youtube.com/watch?v=ycyCtxZ0a9w

Permutation feature importance: Chapter 8 of Interpretable Machine Learning book

Link: https://christophm.github.io/interpretable-ml-book/feature-importance.html

SHAP: SHAP library documentation

Link:
https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%20introduction%20to%20explainable%20AI%20with%20Shapley%20values.html

Model metrics evaluation

Ok, you fit the model but then what? Even more, which metric you choose for your problem? The following links provide a good overview about Pros and Cons of the main regression and classification metrics. You might also often see questions about these metrics in the interview.

Regression metrics: H2O blog tutorial

Link: https://h2o.ai/blog/2019/regression-metrics-guide/

Classification metrics: Evidently AI blog tutorial

Link: https://www.evidentlyai.com/classification-metrics

Cross-validation

Cross-validation is important to understand to effectively avoid overfitting. It was covered partly in some of the above resources, but this guide gives a very good overview with code examples, including consideration of time series validation problems.
Link: https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right

3.2 Neural networks introduction

There are tons of resources on neural networks. It is THE hottest topic. Especially with all the buzz with LLM. In my opinion, to get an intro into the topic, Andrew Ng's specialization is still great. He goes step-by-step and I guarantee you will understand the concept. From that, you can go deeper depending on the domain you are interested in.

It has 5 courses in it, so take a deep breath.

Link: https://www.coursera.org/specializations/deep-learning

3.3 Optimization with Python

Optimization is a relatively hard, heavy-math topic. But it is used in many practical applications. I highly advise you to steadily learn this topic, as it will open great career opportunities.

Introduction to mathematical optimization with Python

Link: https://indrag49.github.io/Numerical-Optimization/

This is an AWESOME resource on numerical optimization. Clear examples in Python with mathematical derivations of the basics.

Bayesian Optimization

Bayesian optimization is a set of optimization methods that allow optimization of black-box functions using input-output sampling. It is often used to tune machine learning models' hyperparameters, but it can also be used to optimize process systems or any other systems that are hard to model.

Source 1: Awesome playground with theory explanation by distill.pub

Link: https://distill.pub/2020/bayesian-optimization/

Source 2: Tutorial with deep theory dive by Nando de Freitas and Co.

Link: http://haikufactory.com/files/bayopt.pdf

Optimization with SciPy

There are various optimization Python libraries you can use for optimization. SciPy is very often used for it. If you come across the need to use SciPy for this, look at these resources:

Link 1: https://caam37830.github.io/book/03_optimization/scipy_opt.html

Link 2:

https://towardsdatascience.com/introduction-to-optimization-constraints-with-scipy-7abd44f6de25

Link 3: https://jiffyclub.github.io/scipy/tutorial/optimize.html#

Link 4: https://people.duke.edu/~ccc14/sta-663-2017/14C_Optimization_In_Python.html

Interactive playground of several optimization methods

Sometimes it is useful to play with parameters and see how the algorithm works. Here is a great playground with a couple of methods.

Link: https://www.benfrederickson.com/numerical-optimization/

Additional resources

Book by Nocedal:

https://www.amazon.ca/Numerical-Optimization-Jorge-Nocedal/dp/0387303030

Extensive list of optimization resources:

https://github.com/ebrahimpichka/awesome-optimization

3.4 Signal processing

Signal processing is often an essential part of an ML projects because you have to be able to filter data from noise outliers and other dirty stuff. On top, if it comes to vibration analysis, you need to be handy with time-frequency domain filters.

Paid source:

I highly recommend a paid course by Mike Cohen. For this price and quality, I consider it essentially free. I have completed the course myself and like it a lot. Since then, I have applied several methods from the course in practice.

Link: https://www.udemy.com/course/signal-processing/?couponCode=2021PM20

Free resources:

If you want fully free alternatives, here are some links on filtering and Fourier transform.

Mean filter

- Example of how to use
- Python implementation

Median filters

- Example of how to use
- Python implementation

Exponential smoothing

- Example of how to use
- Python implementation

Gaussian filter

- Example of how to use
- Python implementation

Fourier transform

- Theory and examples

Low and high pass filters

- Tutorial

3.5 Anomaly detection

In addition to modeling and optimization, another large class of ML problems are related to anomaly detection. I gathered a list of resources to get an overview of the problem and applied methods to solve it.

Review of anomaly types and detection methods:

https://dl.acm.org/doi/abs/10.1145/1541880.1541882?casa_token=BBYx70g2C7cAAAAA%3AOUQK0oXeW5sbOF8a0irqYZrWK6-hEWLRjgU31h9nJzDgVki44L9LV_HT8Ez8WcJpiPNVluP95COaJw

Good overview with Python examples

Link: https://neptune.ai/blog/anomaly-detection-in-time-series

Deep learning methods review:

Link: https://arxiv.org/pdf/2211.05244

List of libraries for time series anomaly detection

Link: https://github.com/rob-med/awesome-TS-anomaly-detection

Selected articles:

- https://towardsdatascience.com/anomaly-detection-in-manufacturing-part-1-an-introduction-8c29f70fc68b
- https://towardsdatascience.com/anomaly-detection-in-manufacturing-part-2-building-a-variational-autoencoder-248abce07349

4. MLOps for data scientists

There will always be a debate if data scientists need to know MLOps / machine learning engineering.

I do not want to argue but my opinion is, of course, we have to know solid basics. I intentionally do not distinguish between MLOps and ML engineering here and just use one definition because some things can be on the MLOps side while others are on the ML engineering side.

MLOps is a huge field itself, but you will be a much stronger professional who can build things from end-to-end on a decent level alone or a solid level within a team of ML engineers. In this roadmap, I provide you with basic resources to start, so with some digging, you can do a decent job of preparing your models for deployment.

4.1 Introduction

Alexey Grigoriev with a team created a great intro course on MLOPs and I will be referring to his course several times. I suggest you take the whole course, but here I mention specific topics.

Another amazing resource is the neptune.ai blog. They do a fantastic job! I will be mentioning their resources all the way.

I suggest you take a look at his intro videos that do not concern the course environment setup.

Link: https://github.com/DataTalksClub/mlops-zoomcamp/tree/main/01-intro

4.2 Model registry and experiment tracking

Model registry and experiment tracking are required to develop a consistent workflow of model creation and deployment, especially if you work in a team. Here are some resources to get an intro about it:

Source 1: https://neptune.ai/blog/ml-model-registry

Source 2: https://neptune.ai/blog/ml-experiment-tracking

A hands-on example of how to perform experiment tracking and create a model registry is here:

Link:
https://github.com/DataTalksClub/mlops-zoomcamp/tree/main/02-experiment-tracking

4.3. ML Pipelines

In my opinion, a data scientist MUST know how to create good clear machine learning pipelines. It is a misery to see still people making terrible hard-coded pipelines while there are many pipeline-building tools. I recommend you do the following:

Step 1: Get an intro to what ML pipelines are and what are the available tools for it:

- Link 1: https://neptune.ai/blog/building-end-to-end-ml-pipeline
- Link 2: https://neptune.ai/blog/best-workflow-and-pipeline-orchestration-tools

Step 2: Complete a section of the MLOps zoomcamp with Mage / Prefect (previous years)

Link: https://github.com/DataTalksClub/mlops-zoomcamp/tree/main/03-orchestration

Step 3: Take a very simple dataset and make a new pipeline with any tool you like without guidance

4.4 Model Monitoring

Model monitoring is another essential topic that comes after model development. This is the weakest point of all ML solutions I have seen and built so far.

For learning model monitoring, there is no better place than the blog of evidentlyai.com.

Blog link: https://www.evidentlyai.com/mlops-guides

They also host the model monitoring section on MLOps zoomcamp, so if you are more comfortable with videos, take a look here:

Link: https://github.com/DataTalksClub/mlops-zoomcamp/tree/main/05-monitoring

4.5 Docker basics

Docker is a tool/platform that allows one to build, share, and run container applications that are reproducible in any new environment. Docker is always scary for data scientists because, come on, what the hell is that? I am with you.

However, the truth is that docker is not that difficult, at least at the level that data scientists should know it.

To start, I recommend a great crash course by Nana:

Link: https://www.youtube.com/watch?v=3c-iBn73dDE

4.6 Additional resources

If you feel you want to go deeper into some of the topics, you may want to look at the MLOps roadmap below. However, my warning is that you need to be careful because it is not possible to know everything. So, make sure you understand the basics that you need at your job or to make portfolio projects, and then move on slowly.

Link: https://marvelousmlops.substack.com/p/mlops-roadmap-2024