Data Science Useful Links

Table of Contents

[Python 2](#_Toc145879445)

[Learn 2](#_Toc145879446)

[Design Aphorisms 2](#_Toc145879447)

[Coding Conventions 2](#_Toc145879448)

[OOP 2](#_Toc145879449)

[Regular Expressions 2](#_Toc145879450)

[Projects & Exercises 2](#_Toc145879451)

[Code Visualization 2](#_Toc145879452)

[Machine Learning 3](#_Toc145879453)

[Learning Basics 3](#_Toc145879454)

[EDA 3](#_Toc145879455)

[Algorithms 3](#_Toc145879456)

[Metrics 3](#_Toc145879457)

[Projects 3](#_Toc145879458)

[Feature Engineering 3](#_Toc145879459)

[Interpreting ML models 3](#_Toc145879460)

[Papers 3](#_Toc145879461)

[Practice 3](#_Toc145879462)

[Ops 4](#_Toc145879463)

[Deep Learning 4](#_Toc145879464)

[Visualize 4](#_Toc145879465)

[NLP 4](#_Toc145879466)

[Git & GitHub 4](#_Toc145879467)

[Report Writing & Documentation 4](#_Toc145879468)

[Code Review 5](#_Toc145879469)

[Courses & Other learning 5](#_Toc145879470)

## Python

### Learn

Basics:

<https://www.pythonlikeyoumeanit.com/intro.html>

https://ehmatthes.github.io/pcc/solutions/README.html

Generators:

<https://inmachineswetrust.com/posts/understanding-generators/>

### Design Aphorisms

[PEP 20 -- The Zen of Python | Python.org](https://www.python.org/dev/peps/pep-0020/)

[Type hinting](https://www.youtube.com/watch?v=SMXsIX3PZ5w&ab_channel=mCoding)

### Coding Conventions

<https://lwn.net/Articles/868490/>

<https://pep8.org/>

### OOP

[Introduction to object-oriented programming with Python - Learn | Microsoft Docs](https://docs.microsoft.com/en-us/learn/modules/python-object-oriented-programming/)

[Object-Oriented Programming (OOP) in Python 3 – Real Python](https://realpython.com/python3-object-oriented-programming/)

[What is Object Oriented Programming? OOP Explained in Depth (educative.io)](https://www.educative.io/blog/object-oriented-programming)

### Regular Expressions

<https://pycon2016.regex.training/regex-intro>

### Projects & Exercises

[About - Project Euler](https://projecteuler.net/)

[The Python Challenge](http://www.pythonchallenge.com/)

[The Big Book of Small Python Projects (inventwithpython.com)](https://inventwithpython.com/bigbookpython/)

### Code Visualization

[Python Tutor - Visualize Python, Java, C, C++, JavaScript, TypeScript, and Ruby code execution](https://pythontutor.com/)

## Machine Learning

### Learning Basics

<https://buildingai.elementsofai.com/>

<https://sebastianraschka.com/blog/2021/ml-course.html>

### EDA

<https://services.google.com/fh/files/misc/exploratory_data_analysis_for_feature_selection_in_machine_learning.pdf>

### Algorithms

Tree based: [52.Trees.key (mlvu.github.io)](https://mlvu.github.io/lectures/52.Trees.annotated.pdf)

Dim reduction: <https://www.youtube.com/watch?v=ioXKxulmwVQ&ab_channel=PyData>

### Metrics

<https://neptune.ai/blog/balanced-accuracy>

### Projects

[Building a repeatable data analysis project in jupyter notebook](https://pbpython.com/notebook-process.html)

[CS 229: Machine Learning Final Projects, Autumn 2014 (stanford.edu)](http://cs229.stanford.edu/projects2014.html)

[Building a spam filter](https://inmachineswetrust.com/posts/sms-spam-filter/)

### Feature Engineering

<https://bookdown.org/max/FES/intro-intro.html>

### Interpreting ML models

<https://christophm.github.io/interpretable-ml-book/index.html>

### Papers

[The latest in Machine Learning | Papers With Code](https://paperswithcode.com/)

### Practice

[Interview Query | Interview Questions](https://www.interviewquery.com/questions?searchQuery=&searchQuestionTag=&searchCompany=&ordering=Default&pageSize=20&page=0)

### Ops

<https://github.com/khuyentran1401/Data-science#mlops>

### Dimensionality Reduction

<http://infolab.stanford.edu/~ullman/mmds/ch11.pdf>

<https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf>

## Deep Learning

### Visualize

[alexlenail/NN-SVG: Publication-ready NN-architecture schematics. (github.com)](https://github.com/alexlenail/NN-SVG)

### Zero to Hero

https://github.com/karpathy/nn-zero-to-hero

## NLP

<https://github.com/datascienceid/nlp-resources>

[Practical NLP (book) - Codes](https://github.com/practical-nlp/practical-nlp-code)

## Git & GitHub

[Learn Git Branching](https://learngitbranching.js.org/)

[SuperSimple dev GitHub Course](https://supersimple.dev/courses/git-github-complete-course-20211)

[Tutorial](https://www.youtube.com/watch?v=apGV9Kg7ics&t=1215s&ab_channel=KunalKushwaha)

[Readme Generator](https://rahuldkjain.github.io/gh-profile-readme-generator/)

https://inmachineswetrust.com/posts/git-deep-dive-checkout/

## Report Writing & Documentation

[Home - How to Write a Good Documentation - Library Guides at UC Berkeley](https://guides.lib.berkeley.edu/how-to-write-good-documentation)

[Process Documentation: Definition, Best Practices & How to Do It (helpjuice.com)](https://helpjuice.com/blog/process-documentation)

[Google style docstrings](https://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html)

## Code Review

<https://google.github.io/eng-practices/review/reviewer/>

## Courses & Other learning

[Introduction to Computer Science with Python](https://www.youtube.com/watch?v=nykOeWgQcHM&list=PLUl4u3cNGP63WbdFxL8giv4yhgdMGaZNA)

[Introduction to Computational Thinking in Data Science](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0002-introduction-to-computational-thinking-and-data-science-fall-2016/lecture-videos/)

[Microsoft Learn | Microsoft Docs](https://docs.microsoft.com/en-us/learn/)

[Repository of Machine Learning courses](https://github.com/dair-ai/ML-YouTube-Courses)

<https://www.dataschool.io/start/>

https://workera.ai/