100 Days Python Challenge for Data Science

Phase 1: Basics (Days 1-25)

Focus: Python fundamentals, data structures, basic programming concepts.

1. Python setup & environment (Jupyter, VSCode)
2. Python syntax, variables, data types
3. Basic input/output, print formatting
4. Operators and expressions
5. Conditional statements (if, else, elif)
6. Loops (for, while)
7. Functions: definition, arguments, return values
8. Lists and list operations
9. Tuples and tuple operations
10. Dictionaries and dictionary operations
11. Sets and set operations
12. String manipulation and formatting
13. List comprehensions
14. Exception handling (try, except)
15. File handling (read/write text files)
16. Modules and packages (import basics)
17. Introduction to virtual environments
18. Lambda functions and map/filter/reduce
19. Basic debugging and using print statements
20. Introduction to Python standard libraries (math, datetime)
21. Working with JSON and CSV files
22. Introduction to pip and installing packages
23. Understanding Python namespaces and scopes
24. Recursion basics
25. Practice problems combining basics

Phase 2: Intermediate (Days 26-50)

Focus: Data manipulation, libraries, and intermediate programming concepts.

26. Introduction to NumPy arrays and operations
27. Array indexing and slicing in NumPy

28. NumPy broadcasting and vectorization
29. Introduction to Pandas: Series and DataFrames
30. DataFrame indexing, selection, and filtering
31. Handling missing data in Pandas
32. Data aggregation and groupby in Pandas
33. Merging, joining, and concatenating DataFrames
34. Data cleaning and preprocessing basics
35. Working with dates and times in Pandas
36. Introduction to Matplotlib for data visualization
37. Plotting line, bar, and scatter plots
38. Customizing plots (labels, titles, legends)
39. Introduction to Seaborn for statistical plots
40. Histograms, boxplots, and heatmaps in Seaborn
41. Introduction to regular expressions in Python
42. Working with APIs and JSON data
43. Introduction to object-oriented programming (OOP)
44. Classes, objects, methods, and attributes
45. Inheritance and polymorphism basics
46. Decorators and generators
47. Writing and using Python scripts
48. Introduction to logging in Python
49. Profiling and optimizing Python code
50. Practice intermediate-level data manipulation problems

Phase 3: Advanced (Days 51-75)

Focus: Advanced data science libraries, algorithms, and concepts.

51. Introduction to Scikit-learn
52. Data preprocessing with Scikit-learn (scaling, encoding)
53. Train/test split and cross-validation
54. Linear regression from scratch and with Scikit-learn
55. Logistic regression and classification metrics
56. Decision trees and random forests
57. Support Vector Machines (SVM)
58. K-Nearest Neighbors (KNN)
59. Clustering: K-Means and hierarchical clustering

60. Dimensionality reduction: PCA
61. Introduction to feature engineering
62. Handling imbalanced datasets
63. Introduction to natural language processing (NLP)
64. Text preprocessing with NLTK/spaCy
65. Bag of Words and TF-IDF vectorization
66. Sentiment analysis basics
67. Introduction to time series analysis
68. ARIMA and moving averages
69. Introduction to deep learning with TensorFlow/Keras
70. Building a simple neural network
71. Convolutional Neural Networks (CNN) basics
72. Recurrent Neural Networks (RNN) basics
73. Transfer learning introduction
74. Model deployment basics (Flask API)
75. Practice advanced machine learning problems

Phase 4: Projects & Applications (Days 76-100)

Focus: Real-world projects, end-to-end workflows, and portfolio building.

76. Exploratory Data Analysis (EDA) project on a dataset
77. Data cleaning and feature engineering project
78. Build a classification model end-to-end
79. Build a regression model end-to-end
80. Sentiment analysis project on Twitter data
81. Image classification project with CNN
82. Time series forecasting project
83. Clustering project with customer segmentation
84. NLP project: text summarization or chatbot
85. Build a recommendation system
86. Deploy a machine learning model with Flask
87. Create interactive visualizations with Plotly/Dash
88. Automate data pipeline with Python scripts
89. Web scraping project with BeautifulSoup/Scrapy
90. Build a portfolio website with GitHub Pages
91. Write detailed README and documentation for projects

92. Version control with Git: branching and merging
93. Collaborate on an open-source data science project
94. Optimize model performance and hyperparameter tuning
95. Introduction to cloud platforms (AWS/GCP/Azure basics)
96. Containerize your project with Docker
97. Build a dashboard for data visualization
98. Prepare a data science project presentation
99. Reflect on learning and write a blog post
100. Plan next steps: advanced topics or specialization
