Comprehensive Course, Python & Data Science: Machine Learning & AI Oriented

This comprehensive course is designed to guide you through the foundational skills, core concepts, and advanced techniques required to excel in data science with a focus on machine learning (ML) and artificial intelligence (AI). This course is structured to be beginner-friendly but will gradually move towards advanced topics.

Course Outline

- Getting Started with Python: A Comprehensive Guide

Basic Programming Concepts:

- **Variables and Data Types**

    - Variables: Containers for storing data values.
    - Data Types: Understand the different data types that can be stored in variables, including integers, floats, strings, and booleans.

- **Operators**

    - Arithmetic Operators: Perform basic mathematical operations like addition, subtraction, multiplication, and division.

    - Comparison Operators: Compare values to determine equality or inequality.

    - Logical Operators: Combine multiple conditions using logical operators such as AND, OR, and NOT.

- **Control Flow**

    - Conditional Statements

        - If-Else Statements: Control the flow of the program based on conditions. Learn how to execute different code blocks based on whether a condition is true or false.

- **Loops**

    - For Loops: Iterate over a sequence of elements, such as a list or range of numbers.

- While Loops: Repeat a code block as long as a specified condition is true.

- Functions
  - Defining Functions

    - Function Definition: Create reusable blocks of code that perform specific tasks.

    - Parameters and Return Values: Learn how to pass data to functions and return results.

  - Lambda Functions

    - Anonymous Functions: Use lambda functions for short, single-use functions.
- Data Structures

  - **Lists**

    - Lists: Ordered, mutable collections of items. Learn how to add, remove, and access elements in a list.

  - **Tuples**

    - Tuples: Ordered, immutable collections. Understand the differences between lists and tuples and when to use each.

  - Sets

    - Sets: Unordered collections with no duplicate elements. Learn set operations like union, intersection, and difference.

  - **Dictionaries**

    - Dictionaries: Collections of key-value pairs. Learn how to store and access data using unique keys.

Object-Oriented Programming (OOP)

- **Classes and Objects**

    - Classes: Blueprints for creating objects. Understand the concepts of attributes and methods.

    - Objects: Instances of classes. Learn how to create and manipulate objects.

- **Inheritance**

    - Inheritance: Derive new classes from existing ones to reuse and extend functionality.

- Modules and Packages
    - **Importing Modules**

        - Standard Library: Utilize Python's extensive standard library for various tasks, such as math operations and file handling.

    - **Creating Modules**

        - Custom Modules: Organize your code into reusable modules and import them into your projects.

Working with Files

- **File Operations:**

    - Reading and Writing Files: Learn how to open, read, write, and close files in Python.

- **Error and Exception Handling**

    - Try-Except Blocks

    - Handling Exceptions: Write robust code by handling errors gracefully using try-except blocks.

Useful Libraries

- **Numpy**

- Numerical Operations: Perform advanced mathematical operations using the Numpy library.

- **Pandas**

    - Data Manipulation: Handle and analyze data efficiently with the Pandas library.

- **Matplotlib**

    - Data Visualization: Create visualizations to represent your data graphically.

- **Scikit-Learn**

    - **Machine Learning:** Implement basic machine learning algorithms using Scikit-Learn.

- **TensorFlow**
    - **Deep Learning:** Build and train deep learning models with TensorFlow.

Advanced Topics

Decorators

- **Function Enhancement:** Use decorators to modify the behavior of functions or methods.

Generators

- **Efficient Iteration:** Create functions that return an iterable set of items, allowing you to iterate over large datasets efficiently.

Practice Projects

- **Beginner Projects**

    - **Calculator:** Build a simple calculator that performs basic arithmetic operations.

    - **To-Do List Application:** Create an application to manage a list of tasks.

    - **Web Scraper:** Write a script to extract data from websites.

- **Intermediate Projects**

  - **Data Analysis with Pandas:** Analyze datasets and extract insights using Pandas.

  - **Machine Learning Model:** Develop a basic machine learning model using Scikit-Learn.

  - **Web Application:** Build a simple web application using Flask or Django.

- Advanced Projects

  - **Deep Learning Model:** Train a deep learning model for image or text classification using TensorFlow.

  - **NLP Application:** Develop a natural language processing application for sentiment analysis or text generation tasks.

  - **Full-Stack Web Application:** Create a full-stack web application with a frontend and backend.

- **Additional Resources**

  - **Books**

    - **Python Crash Course by Eric Matthes:** A hands-on introduction to Python.

    - **Learning Python by Mark Lutz:** A comprehensive guide to Python programming.

    - **Fluent Python by Luciano Ramalho:** In-depth coverage of Python's advanced features.

  - **Online Courses**

    - **Python for Everybody by Dr. Charles Severance (Coursera):** A beginner-friendly course covering the basics of Python.

    - **Complete Python Bootcamp by Jose Portilla (Udemy):** An in-depth course covering Python from beginner to advanced levels.

- **Python Data Structures (Coursera):** Focuses on data structures and their applications in Python.

- **Websites and Tutorials**

  - **Real Python:** Offers a variety of tutorials and articles on Python programming.

  - **Python Official Documentation:** Comprehensive documentation for Python's features and libraries.

  - **GeeksforGeeks Python Programming Language:** A collection of tutorials and examples covering Python concepts and applications.

By following this guide and leveraging the resources provided, you will build a strong foundation in Python programming. This will enable you to tackle more complex projects and concepts in data science, machine learning, and beyond. Happy coding!

1. Introduction to Data Science
   a. What is Data Science?
   b. Overview of Data Science
   c. Applications and Impact
   d. Data Science Workflow
2. Getting Started with Python
   a. Installation and Setup
   b. Python Basics: Variables, Control Flow, Functions
   c. Object-Oriented Programming (OOP) Concepts
   d. Lambda Functions
3. Resources:
   a. Book: "Automate the Boring Stuff with Python" by Al Sweigart
   b. Course: "Python for Everybody" by Dr. Charles Severance (Coursera)
   c. Tasks: Solve basic programming problems on platforms like LeetCode or HackerRank
4. Programming Fundamentals
5. Data Structures
   a. Lists, Linked Lists
   b. Queues, Stacks
   c. Arrays, Dictionaries
   d. Graphs, Trees (Binary Search Trees, Red-Black Trees, B-Trees)
   e. Hash Tables
6. Resources:
   a. Book: "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein
   b. Course: "Data Structures and Algorithms Specialization" (Coursera)
7. Algorithms
   a. Sorting and Order Statistics (Insertion, Heapsort, Quicksort, Sorting in Linear Time)

      b. Basic Machine Learning Algorithms: K-Nearest Neighbour, K-means, Linear Regression

      c. Advanced Algorithms: Clustering, Multiplicative-weights Algorithms, Gradient Descent, NP-Completeness, AI Algorithms

8. Resources:
      a. Book: "Algorithms Unlocked" by Thomas H. Cormen
      b. Course: "Algorithms Specialization" (Coursera)

9. Working with Libraries
      a. Python Libraries
      b. Numpy: Numerical Operations
      c. Pandas: Data Manipulation
      d. Matplotlib: Data Visualization

10. Resources:
      a. Book: "Python Data Science Handbook" by Jake VanderPlas
      b. Course: "Applied Data Science with Python" (Coursera)

11. Machine Learning Libraries
      a. Scikit-Learn: Basic Machine Learning Algorithms
      b. Books, Videos, Articles, Blogs, Resources
      c. TensorFlow: Deep Learning
      d. Books, Videos, Articles, Blogs, Resources

12. Resources:
      a. Book: "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron
      b. Course: "Machine Learning with Python" (Coursera)

13. Databases
      a. SQL for Data Science
          i. Introduction to SQL
          ii. Creating Databases
          iii. Understanding Data Types
          iv. Basic Math with SQL
          v. Data Exploration
          vi. Joining Tables in RDBMS
          vii. Advanced Techniques

14. Resources:

      a. Course: "SQL for Data Science" (Coursera)
      b. Book: "Learning SQL" by Alan Beaulieu

15. Mathematics for Data Science
      a. Probability and Statistics
      b. Descriptive Statistics
      c. Probability Distributions
      d. Inferential Statistics
      e. Hypothesis Testing

16. Resources:
      a. Book: "Think Stats" by Allen B. Downey

b. Course: "Statistics with R" (Coursera)
   i. Video:
    [Probability](https://youtu.be/Vfo5le26IhY?si=f1VvhwvvU4rYExth)
c. Linear Algebra and Calculus
d. Vectors, Matrices, Eigenvalues, and Eigenvectors
e. Derivatives, Integrals, Optimization
   i. Resources:
      1. Book: "Linear Algebra and Its Applications" by Gilbert Strang
      2. Course: "Mathematics for Machine Learning" (Coursera)
      3. Videos:
      [Linear algebra1](https://youtu.be/JnTa9XtvmfI?si=iqVlSMY2afL4aJFl),
      [Linear Algebra]
      (https://youtu.be/n9jZmymHX6o?si=IuZHMbp0CKak2dLY),
      [Calculus](https://youtu.be/HfACrKJ_Y2w?si=jJahD38mgkH57Skm)

A. Core Machine Learning
  a. Supervised Learning

    i. Regression: Linear Regression, Logistic Regression

    ii. Classification: Decision Trees, Random Forests, Support Vector Machines

    iii. Evaluation Metrics: Precision, Recall, F1-Score, ROC-AUC

    iv. Resources:

      1. Book: "Pattern Recognition and Machine Learning" by Christopher M. Bishop
      2. Course: "Supervised Machine Learning: Regression and Classification" (Coursera)

  b. Unsupervised Learning
    i. Clustering: K-Means, Hierarchical Clustering
    ii. Association Rule Learning: Apriori, Eclat
    iii. Dimensionality Reduction: PCA, LDA, t-SNE
  c. Resources:
    i. Book: "Data Mining: Concepts and Techniques" by Jiawei Han, Micheline Kamber, and Jian Pei
    ii. Course: "Unsupervised Machine Learning" (Coursera)
  d. Reinforcement Learning

    i. Markov Decision Processes
    ii. Q-Learning
    iii. Deep Q-Learning

  e. Resources:

        i.   Book: "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto

B. Course: "Reinforcement Learning Specialization" (Coursera)

C. Deep Learning and AI

    a. Neural Networks

    b. Introduction to Neural Networks

    c. Backpropagation

    d. Hyperparameter Tuning

    e. Convolutional Neural Networks (CNNs)
        i.   Convolutional Layers
        ii.   Pooling Layers
        iii.   CNN Architectures

    f. Recurrent Neural Networks (RNNs)
        i.   RNN Basics
        ii.   Long Short-Term Memory (LSTM)
        iii.   Gated Recurrent Units (GRUs)

    g. Advanced Topics
        i.   Transfer Learning
        ii.   Generative Adversarial Networks (GANs)
        iii.   Transformer Models (BERT, GPT)
        iv.   Resources:

                1. Book: "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

                2. Course: "Deep Learning Specialization" by Andrew Ng (Coursera)

    h. Natural Language Processing (NLP)
        i.   Text Preprocessing
        ii.   Tokenization
        iii.   Stop Words Removal
        iv.   Stemming and Lemmatization

    i. Text Classification and Sentiment Analysis
        i.   Bag of Words, TF-IDF
        ii.   Word Embeddings: Word2Vec, GloVe

D. Advanced NLP
   a. Sequence-to-Sequence Models
   b. Attention Mechanisms
   c. Transformer Models
      i. Resources:
         1. Book: "Speech and Language Processing" by Daniel Jurafsky and James H. Martin

         2. Course: "Natural Language Processing with Classification and Vector Spaces" (Coursera)
E. Data Visualization
   a. Tools for Data Visualization
   b. Tableau: Basics, Advanced Visualizations, Dashboards
   c. Power BI: Data Transformation, Visualizations, Reports

F. Storytelling with Data
   a. Principles of Data Visualization
   b. Crafting Effective Visuals
   c. Data Storytelling
      i. Resources:
         1. Book: "Storytelling with Data" by Cole Nussbaumer Knaflic
         2. Course: "Data Visualization with Tableau" (Coursera)
G. Practical Experience and Projects
   a. Building Projects
      i. Spam Filters
      ii. Recommendation Engines
      iii. Fraud Detection Systems
      iv. Social Network Analysis
H. Internships and Competitions
   a. Participate in Data Science Competitions (e.g., Kaggle)
   b. Apply for Internships to Gain Real-World Experience
I. Portfolio Development
   a. Create a Portfolio Showcasing Your Projects and Skills
   b. Use GitHub, Personal Blogs, or a Dedicated Portfolio Website

Happy learning!