Machine Learning

06048203

Grading Criteria

Criteria	Description	Weight
Assignments	Completeness and correctness of programming assignments	25%
Mid term Project	Quality of the machine learning project, including problem formulation and execution	40%
Final Exam	Performance in exams, assessing theoretical knowledge and problem-solving skills	35%

Introduction to Python, NumPy, and Matplotlib

01

Python: A high-level programming language known for its simplicity and readability.

02

NumPy: A powerful library for numerical computing in Python, providing support for large, multi-dimensional arrays and a collection of mathematical functions.

03

Matplotlib: A popular data visualization library in Python, offering a wide range of plotting options.

Importing

Importing NumPy: Begin by importing the NumPy library into your Python script or Jupyter Notebook.

- import numpy as np

Creating

Creating arrays:

- np.array(): Create an array from a Python list or tuple.
- np.zeros(): Create an array filled with zeros.
- np.ones(): Create an array filled with ones.
- np.arange(): Create an array with a range of values.

Introduction to Matplotlib

Matplotlib: A popular data visualization library in Python.

Why use Matplotlib?

- Create various types of plots: line plots, scatter plots, bar plots, histograms, etc.
- Customize plots with labels, titles, legends, colors, and styles.
- Visualize and communicate data effectively.

- Importing Matplotlib: Begin by importing the Matplotlib library.
    - Example: import matplotlib.pyplot as plt
- Plotting a basic line graph:
    - Example:pythonCopy code

    ```x = [1, 2, 3, 4, 5]
    y = [2, 4, 6, 8, 10]
    plt.plot(x, y)
    plt.xlabel('X-axis')
    plt.ylabel('Y-axis')
    plt.title('Line Plot')
    plt.show()```
- Customizing plots:
    - Labels: plt.xlabel('x_label') and plt.ylabel('y_label')
    - Titles: plt.title('Plot Title')
    - Legends: plt.legend(['line1', 'line2'])
    - Color and style options: plt.plot(x, y, 'r--')
- Other types of plots:
    - Scatter plots: plt.scatter(x, y)
    - Bar plots: plt.bar(x, height)
    - Histograms: plt.hist(data, bins)
    - Pie charts: plt.pie(data, labels)
    ```