

Week2 - Introduction to Pandas and Scipy

DS3010: Introduction to Machine Learning Lab

Timing: 02:00 PM to 04:45 PM

Max Marks:

Instructions

1. Submit one .ipynb file containing all answers. The name should be [student_name]_[rollno]_[lab].ipynb
 2. Write the questions in separate text blocks before the answers.
 3. Outputs for all sub-questions should be given and the code should be executable.
 4. Write justifications for your choices where needed.
 5. Ensure that all plots include clear labels and legends for better interpretation.
 6. Use of generative AI tools (such as ChatGPT, Gemini, etc.) is strictly prohibited. Any submission found to contain AI-generated or plagiarized content will receive a score of zero, and disciplinary action.
-

Basics of Pandas

1. (a). Create a DataFrame from the following dictionary:

```
data = {  
    "Name": ["Alice", "Bob", "Charlie", "David", "Eva"],  
    "Age": [25, 30, 35, 40, 45],  
    "Department": ["HR", "Engineering", "Marketing", "Engineering", "HR"],  
    "Salary": [50000, 80000, 60000, 85000, 52000]  
}
```


(b). Display the first 3 rows of the DataFrame.
(c). Print the data types of each column.
(d). Print summary statistics of the numerical columns.
2. (a). Load the Studentsperformance.csv into a Pandas DataFrame.
(b). Display first 5 rows, dataset shape, column names, and data types.
(c). Find how many missing values (if any) are in each column.
(d). Compute mean, median, and standard deviation for Math score.
(e). Identify which subject has the highest average score.
(f). Compute average score per subject grouped by gender.
(g). Calculate the correlation matrix among Math, Reading, and Writing scores.

Basics of SciPy

3. (Hint: use numpy, matplotlib, and scipy.stats)
 - (a). Generate 1000 random samples from a normal distribution with mean = 50 and standard deviation = 10 using an appropriate function from NumPy.
 - (b). Compute the sample mean and sample standard deviation using NumPy functions, and verify how close they are to the actual values (50 and 10).
 - (c). Using Matplotlib, plot a histogram of the generated data. Then, overlay a probability density function (PDF) curve(scipy.stats.norm) using both actual and sample parameters.
 - (d). Label your axes and title the plot appropriately.
4. (Hint: use numpy, matplotlib, and scipy.spatial.distance)
 - (a). Generate two sets A and B, each containing 5 points in 2D space. Each coordinate should be a random integer between 0 and 50.
 - (b). Compute the Euclidean distance between every pair of points, one from set A and one from set B.
 - (c). Identify and print the pair (one from A, one from B) that has the minimum distance between them.
 - (d). Plot both point sets using different colors and highlight the closest pair using a red dashed line.
5. (Hint: use scipy.optimize and matplotlib)
 - (a). You are given two functions, one convex and one non-convex. Define the following functions:

Convex function: $f(x)=2x^2 + 3x + 10$

Non – Convex function : $f(x) = x^4 - 3x^3 + 2$
 - (b). find the value of x that minimizes each function.
 - (c). Print the value of x and the corresponding function value at the minimum.
 - (d). Plot both functions using Matplotlib over a suitable range. On each plot:
 - (a) Show the function curve.
 - (b) Mark the minimum point found.
 - (c) Compare and explain briefly how the optimizer behaves differently for convex vs. non-convex functions