

MA722 - Foundations of Machine Learning Algorithms

Lab Problem Sheet - 3

1. Write a python program that perform *LU* decomposition.
 - Define a square matrix A . Decompose A into:
 - A permutation matrix P
 - A lower triangular matrix L
 - An upper triangular matrix U
 - Print the matrices P , L , and U .
 - Verify that the decomposition satisfies the condition: $P * A = L * U$
2. Write a python program that perform *LDU* decomposition by:
 - Manual step-by-step Gaussian elimination method
 - Using `scipy.linalg.lu`
3. Write a Python program to perform Linear Regression on a given dataset.
 - Input: Two lists X and Y containing data points.
 - Implement Linear Regression in two ways:
 - From scratch using Gradient Descent.
 - Using the built-in Linear Regression class from `scikit-learn`.
 - Print the slope and intercept of the line.
 - Plot the data points and the regression line.
4. Consider a dataset with two predictor variables X_1 and X_2 and a response variable Y .
 - (a) Implement Multiple Linear Regression from scratch by:
 - Construct the design matrix X (with a column of ones for the intercept).
 - Compute the regression coefficients (intercept, coefficient of X_1 , coefficient of X_2).
 - Compute the predicted values $\hat{Y}$.
 - (b) Calculate the **coefficient of determination** (R^2):
$$R^2 = 1 - \frac{\sum (Y - \hat{Y})^2}{\sum (Y - \bar{Y})^2}$$
 - (c) Verify the results using `scikit-learn`'s `LinearRegression` model and confirm that the intercept, coefficients, and R^2 match.
 - (d) Plot the dataset points in **3D** and the fitted **regression plane**.
5. Write a Python program to compute the Moore–Penrose inverse (pseudoinverse) of a general $m \times n$ matrix.

- Define a matrix A of size $m \times n$
- Compute the Moore–Penrose inverse A^+ .
- Print both the original matrix and its pseudoinverse.
- Verify at least one of the conditions, for example: $AA^+A = A$
