LAB 1: SUPERVISED LEARNING – REGRESSION

(Duration: 2 sessions)

Learning Objectives :

- ✓ Understand the basics of linear regression (simple & multivariate)
- ✓ Learn dataset preparation (train-test split, feature scaling, and data generation)
- ✓ Identify key challenges (overfitting, underfitting, multicollinearity)
- ✓ Apply model evaluation techniques (MSE, R^2, cross-validation)
- ✓ Explore regularization methods (Ridge & Lasso) to improve model performance

Exercise 1: Understanding Linear Relationships

1. Generate a dataset where y is linearly dependent on X (e.g., y=5X+3 with some noise).
2. Plot the dataset and visually determine if a linear relationship exists.
3. Compute the correlation coefficient between X and y. What does it tell you?

Exercise 2: Training a Simple Linear Regression Model

1. Split the dataset into 80% training and 20% testing sets.
2. Train a simple linear regression model.
3. Extract the slope (coefficient) and intercept of the model.
4. Interpret these values: What do they represent?

Exercise 3: Evaluating Model Performance

1. Predict the test set values and calculate:
   a. Mean Squared Error (MSE)
   b. Mean Absolute Error (MAE)
   c. R^2 score
2. What do these metrics indicate about the model's performance?
3. Plot the actual vs. predicted values.
4. Optional: Create a residuals plot (residuals vs. predicted values).

Exercise 4: Multivariate Linear Regression & Data Preparation

1. Generate a synthetic dataset with 3 features and 1 target using `sklearn.datasets.make_regression` (add noise=20, with 1000 samples). Then multiply all data of the first feature by 500, and divide data of the second feature by 2000.
2. Split the data into 70% train and 30% test.
3. Apply feature scaling using `StandardScaler` and train a multivariate linear regression model.
4. Compare the model's coefficients before and after scaling. Does feature scaling impact model predictions or just coefficient values?
5. Evaluate using MSE and R^2.

Exercise 5: Identifying Underfitting and Overfitting in Polynomial Regression

1. Generate Synthetic Data:
   - Create a dataset with 3 input features (X1, X2, X3) and a target y using the following formula, with Gaussian Noise (mean=0, std=2) and number of samples = 500.

$$y = 2X_1^2 + 4X_1X_2 - 3X_3 + 5\sin(X_3) + \text{noise}$$

(Note: This ensures the true relationship can be captured by a degree=2 polynomial model).

2. Split the dataset into 70% training and 30% testing.
3. Train Multiple Models:
   a. Model A: Linear Regression (degree=1).
   b. Model B: Polynomial Regression (degree=2).
   c. Model C: Polynomial Regression (degree=10).
4. Evaluate Model Performance:
   a. Calculate Mean Squared Error (MSE) and R^2 for both training and test sets.
   b. Compare results across models.
5. Plot Learning Curves for Model A (Linear), Model B (Poly Degree=2), and Model C (Poly Degree=10)
6. Optional:
   a. What is Cross-Validation for? When do we use it?
   b. Use Cross-Validation to evaluate the models A, B and C.

Exercise 6 : Regularization Ridge and Lasso

1. Generate Synthetic Data (use the same equation)
2. Train Models:
   a. Model A: Polynomial regression (degree = 10, no regularization).
   b. Model B: Ridge regression (degree = 10, tuned α).
   c. Model C: Lasso regression (degree = 10, tuned α).
3. Optimize α (λ) for Ridge & Lasso using Cross-Validation:
   - Use GridSearchCV to find the best α for Lasso and Ridge (Try α values: [0.001, 0.01, 0.1, 1, 10, 100])
4. Compute Mean Squared Error (MSE) and R^2 for all models.
   - Compare Ridge vs. Lasso: How many polynomial features does each model keep?