Lab05: Renewable Electricity Output Modeling with Python

Course: Programming for Data Science
Professor: Dr. Carlos J. Costa

March 6, 2026

Objective

In this lab, you will model the share of **renewable electricity output (% of total electricity output)** for multiple countries over time. The lab requires preprocessing, feature selection, handling missing data, and training a regression model to predict renewable electricity output. Each group must use a specific regression algorithm.

Additionally, students must explain how the assigned algorithm works.

Groups and Assigned Regressors

- **Group 1:** Ordinary Least Squares (OLS)
- **Group 2:** Support Vector Regression (SVM)
- **Group 3:** Random Forest Regressor
- **Group 4:** Multi-Layer Perceptron (MLP)
- **Group 5:** AdaBoost Regressor
- **Group 6:** Ridge Regression

Important: Only the assigned algorithm should be used. Using another algorithm will incur penalties.

Dataset

The dataset contains electricity indicators for multiple countries and years. Available at:
`https://github.com/masterfloss/data/raw/refs/heads/main/WBESG6123.xlsx`
Variables:

- `REF_AREA_LABEL`: Country name
- `TIME_PERIOD`: Year
- `INDICATOR_LABEL`: Indicator name
- `OBS_VALUE`: Indicator value

Lab Tasks

1. Load Dataset

- Import required Python libraries (`pandas`, `numpy`, `sklearn`, etc.).
- Read the Excel dataset into a pandas DataFrame.

2. Preprocess Data

1. Rename columns: `REF_AREA_LABEL` → `country`, `TIME_PERIOD` → `year`, `INDICATOR_LABEL` → `indicator`, `OBS_VALUE` → `value`.
2. Keep only columns: `country, year, indicator, value`.
3. Convert from long to wide format using `pivot_table`:

$$index = [country, year], \quad columns = indicator, \quad values = value$$

4. Define the target variable: `"Renewable electricity output (% of total electricity output)"`.
5. Separate predictors (X) and target (y).

3. Feature Selection

- Compute correlation matrix of predictors.
- Remove highly correlated predictors (correlation > 0.9) to avoid multicollinearity.

4. Handle Missing Data

- Drop rows where the target variable is missing.
- Impute missing predictor values using column means.

5. Train/Test Split

- Split data into training and test sets (80/20).
- Use a fixed random seed for reproducibility (e.g., 42).

6. Train Regression Model

- Train the assigned regressor for your group.
- Only OLS requires adding a constant term for statsmodels.
- Generate predictions for the test set.
- Evaluate model performance using R^2, RMSE, and MAE.
- For OLS (Group 1), also provide the model summary including coefficients, p-values, confidence intervals, and overall model significance.

7. Explain the Algorithm

- Provide a brief explanation (1–2 paragraphs) of how your assigned regression algorithm works, including:
  - Main idea
  - Key parameters
  - How it handles prediction
  - Strengths and limitations

8. Refactor to Object-Oriented Programming (OOP)

- Refactor the entire pipeline to use object-oriented programming (OOP).
- Your OOP class should encapsulate:
  - Data reading
  - Feature selection
  - Model training and evaluation

Deliverables

Everything should be presented in a Jupyter Notebook or Python script file, including:

1. Python inicial version, the OOP model class, and evaluation.
2. Accuracy metrics for your model (R^2, RMSE, MAE).
3. Algorithm explanation (1–2 paragraphs).
4. For Group 1 (OLS): printed regression summary.
5. Brief discussion (1 paragraph) of the results and potential predictor importance.

Grading Notes

- Adhering to the assigned regressor is mandatory; deviations will incur penalties.
- Proper preprocessing, multicollinearity handling, and missing data imputation are required.
- Accuracy metrics must be calculated and reported correctly.
- OOP implementation and algorithm explanation are required for full credit.