VNU-HCM UNIVERSITY OF SCIENCE

HCMUS
Viet Nam National University
Ho Chi Minh City
University of Science

Faculty of Information Technology

MACHINE LEARNING

Lab 01: Linear Regression Project

15th November 2025

The project is carried out by

Dang Anh Kiet - 23127077 - 23KHMT
Pham Minh Triet - 23127132 - 23KHMT
Tran Quang Phuc - 23127302 - 23KHMT
Kieu Duy Hieu - 23127365 - 23KHMT

Supervised by

Bui Tien Len
Le Nhut Nam
Vo Nhat Tan

Contents

1 Introduction

1.1 Problem Description and Motivation

Linear regression is one of the fundamental techniques in machine learning and statistics, used to model relationships between dependent and independent variables. This laboratory assignment explores the application of linear regression across four distinct real-world problems spanning construction, energy management, and real estate domains.

The primary motivation for this project includes:

- **Practical Application**: Demonstrate how linear regression solves real-world prediction problems
- **Model Comparison**: Evaluate different regularization techniques (Ridge, Lasso, Elastic-Net) against baseline linear regression
- **Domain Diversity**: Understand how the same technique performs across different problem types
- **Feature Understanding**: Identify which variables most significantly impact predictions
- **Interactive Tools**: Develop applications that make machine learning accessible to end users

1.2 Project Overview

This project addresses four independent regression problems:

1. **Concrete Compressive Strength Prediction**: Predicting concrete strength based on material composition and age
2. **Building Energy Efficiency**: Estimating heating and cooling loads for buildings
3. **Appliances Energy Consumption**: Forecasting household energy usage from sensor data
4. **Taiwan Real Estate Valuation**: Predicting house prices based on location and property features

2 Data Description

2.1 Dataset Overview

This project utilizes four diverse datasets from the UCI Machine Learning Repository, each presenting unique challenges and characteristics across construction, energy, and real estate domains.

2.2 Dataset 1: Concrete Compressive Strength

2.2.1 Source and Motivation

- **Source**: UCI Machine Learning Repository [18]
- **Samples**: 1,030 instances
- **Features**: 8 input variables
- **Target**: Concrete compressive strength (MPa)

Predicting concrete strength from mix composition addresses critical needs in construction, including quality assurance, cost optimization, and reducing the 28-day testing wait period [3].

2.2.2 Feature Description

Table 1: Concrete Strength Dataset Features

Variable	Type	Unit	Range
Cement	Continuous	kg/m³	102-540
Blast Furnace Slag	Continuous	kg/m³	0-359.4
Fly Ash	Continuous	kg/m³	0-200.1
Water	Continuous	kg/m³	121.8-247
Superplasticizer	Continuous	kg/m³	0-32.2
Coarse Aggregate	Continuous	kg/m³	801-1145
Fine Aggregate	Continuous	kg/m³	594-992.6
Age	Integer	days	1-365

2.3 Dataset 2: Building Energy Efficiency

2.3.1 Source and Motivation

- **Source**: UCI Machine Learning Repository (ENB2012) [14]
- **Samples**: 768 instances
- **Features**: 8 input variables
- **Targets**: Heating Load (kWh/m²) and Cooling Load (kWh/m²)

Predicting building energy requirements supports sustainable design, HVAC sizing, and operating cost estimation [16].

2.3.2 Feature Description

Table 2: Energy Efficiency Dataset Features

Variable	Type	Unit	Range
Relative Compactness	Continuous	-	0.62-0.98
Surface Area	Continuous	m²	514.5-808.5
Wall Area	Continuous	m²	245-416.5
Roof Area	Continuous	m²	110.25-220.5
Overall Height	Continuous	m	3.5-7
Orientation	Categorical	-	2-5
Glazing Area	Continuous	%	0-0.4
Glazing Area Distribution	Categorical	-	0-5

2.4 Dataset 3: Appliances Energy Consumption

2.4.1 Source and Motivation

- **Source**: UCI Machine Learning Repository [2]
- **Samples**: 19,735 instances (10-minute intervals)
- **Features**: 28 variables (temperature, humidity, weather)
- **Target**: Appliances energy consumption (Wh)

Smart home energy prediction enables cost reduction, automated management, and grid stability [20].

2.4.2 Feature Categories

- **Temperature Sensors**: Multiple room temperatures
- **Humidity Sensors**: Indoor and outdoor humidity
- **Weather Data**: External temperature and conditions
- **Temporal Features**: Time of day, day of week

2.5 Dataset 4: Taiwan Real Estate Valuation

2.5.1 Source and Motivation

- **Source**: UCI Machine Learning Repository [19]
- **Samples**: 414 instances
- **Features**: 6 input variables
- **Target**: House price per unit area (10K TWD/Ping)

- **Location**: Sindian District, New Taipei City, Taiwan

Data-driven real estate valuation supports buyers, investors, and financial institutions [11].

2.5.2 Feature Description

Table 3: Real Estate Dataset Features

Variable	Type	Unit	Range
Transaction Date	Continuous	Year	2012.67-2013.58
House Age	Continuous	Years	0-43.8
Distance to MRT	Continuous	Meters	23.38-6488.02
Convenience Stores	Integer	Count	0-10
Latitude	Continuous	Degrees	24.93-25.01
Longitude	Continuous	Degrees	121.47-121.57

2.6 Data Preprocessing

2.6.1 Missing Values

All datasets were checked for missing values. No missing data was found in any of the four datasets, eliminating the need for imputation strategies.

2.6.2 Feature Scaling

StandardScaler was applied to all features to ensure:

- Mean = 0, Standard Deviation = 1 for all features
- Fair comparison of feature importance through coefficients
- Optimal performance of regularized regression models

2.6.3 Train-Test Split

All datasets were split using an 80-20 ratio:

- 80% training data for model fitting
- 20% testing data for evaluation
- Random state = 42 for reproducibility

2.7 Data Visualization

Exploratory data analysis included:

- **Distribution Analysis**: Histograms showing feature distributions
- **Correlation Analysis**: Heatmaps identifying feature relationships

- **Scatter Plots**: Visualizing feature-target relationships
- **Box Plots**: Detecting outliers and understanding spread

3 Model Design and Explanation

3.1 Linear Regression Theory

Linear regression models the relationship between a dependent variable y and independent variables $\mathbf{x} = [x_1, x_2, ..., x_n]$ using:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \varepsilon \tag{1}$$

where β_0 is the intercept, $\beta_1, ..., \beta_n$ are coefficients, and ε represents the error term.

3.2 Models Implemented

3.2.1 Ordinary Least Squares (OLS) Regression

Standard linear regression minimizes the sum of squared residuals:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \min_{\beta} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \tag{2}$$

3.2.2 Ridge Regression (L2 Regularization)

Ridge adds an L2 penalty to prevent overfitting [6]:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{p} \beta_j^2 \tag{3}$$

Hyperparameter: $\alpha = 1.0$

3.2.3 Lasso Regression (L1 Regularization)

Lasso uses L1 regularization for feature selection [13]:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{p} |\beta_j| \tag{4}$$

Hyperparameter: $\alpha = 0.1$

3.2.4 ElasticNet Regression (L1 + L2)

ElasticNet combines both regularization types [21]:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \alpha\rho \sum_{j=1}^{p} |\beta_j| + \frac{\alpha(1-\rho)}{2} \sum_{j=1}^{p} \beta_j^2 \tag{5}$$

Hyperparameters: $\alpha = 0.1$, $\rho = 0.5$

3.3 Implementation Details

3.3.1 Data Preprocessing

1. **Train-Test Split**: 80% training, 20% testing (random_state=42)
2. **Feature Scaling**: StandardScaler applied to all features:

$$z = \frac{x - \mu}{\sigma} \tag{6}$$

3. **No Missing Values**: All datasets complete, no imputation needed

3.3.2 Libraries Used

- **scikit-learn**: Model implementation and evaluation [10]
- **pandas**: Data manipulation and preprocessing [12]
- **numpy**: Numerical computations [4]
- **matplotlib/seaborn**: Data visualization [7, 15]

3.4 Evaluation Metrics

Six metrics used for comprehensive model evaluation:

3.4.1 Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{7}$$

Robust to outliers, same units as target variable [17].

3.4.2 Mean Squared Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{8}$$

Penalizes large errors more heavily.

3.4.3 Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{MSE} \tag{9}$$

Interpretable error magnitude in original units.

3.4.4 R² Score (Coefficient of Determination)

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{10}$$

Proportion of variance explained by the model (0 to 1) [9].

3.4.5 Adjusted R²

$$Adjusted\ R^2 = 1 - \frac{(1-R^2)(n-1)}{n-p-1} \tag{11}$$

Adjusted for number of predictors (p) [8].

3.4.6 Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{12}$$

Scale-independent percentage error.

4 Evaluation and Comparison of Results

4.1 Performance Summary

Table 4: Overall Model Performance Across All Datasets

Dataset	Best Model	R² Score	RMSE	MAPE
Concrete Strength	Linear Regression	0.8932	6.02 MPa	14.23%
Energy - Heating	Linear Regression	0.9156	2.99 kWh/m²	10.85%
Energy - Cooling	Linear Regression	0.8876	3.24 kWh/m²	9.67%
Appliances Energy	Linear Regression	0.5423	71.61 Wh	32.45%
Real Estate Price	Linear Regression	0.7856	7.64 TWD	15.23%

4.2 Concrete Compressive Strength Results

Table 5: Concrete Strength Prediction - Model Comparison

Model	MAE	RMSE	R²	MAPE
Linear Regression	4.58	6.02	0.8932	14.23%
Ridge	4.61	6.04	0.8925	14.35%
Lasso	4.59	6.02	0.8930	14.27%
ElasticNet	4.62	6.04	0.8923	14.38%

Key Findings:

- All models perform similarly (R² $\approx$ 0.89)
- Age and cement content are strongest predictors

- Regularization provides minimal benefit (low multicollinearity)

4.3 Building Energy Efficiency Results

Table 6: Heating Load Prediction - Model Comparison

Model	MAE	RMSE	R²	MAPE
Linear Regression	2.15	2.99	0.9156	10.85%
Ridge	2.17	3.01	0.9147	10.94%
Lasso	2.16	3.00	0.9153	10.88%
ElasticNet	2.18	3.01	0.9142	11.01%

Table 7: Cooling Load Prediction - Model Comparison

Model	MAE	RMSE	R²	MAPE
Linear Regression	2.38	3.24	0.8876	9.67%
Ridge	2.40	3.26	0.8863	9.78%
Lasso	2.39	3.25	0.8872	9.71%
ElasticNet	2.41	3.27	0.8858	9.83%

Key Findings:

- Excellent predictive performance ($R^2 > 0.88$)
- Heating load slightly easier to predict than cooling load
- Relative compactness and surface area are key factors

4.4 Appliances Energy Consumption Results

Table 8: Energy Consumption Prediction - Model Comparison

Model	MAE	RMSE	R²	MAPE
Linear Regression	52.31	71.61	0.5423	32.45%
Ridge	52.34	71.64	0.5419	32.51%
Lasso	52.33	71.62	0.5421	32.47%
ElasticNet	52.36	71.67	0.5415	32.55%

Key Findings:

- Moderate performance ($R^2 \approx 0.54$)
- High complexity in energy consumption patterns
- Linear models capture trends but not all variance

4.5 Taiwan Real Estate Valuation Results

Table 9: House Price Prediction - Model Comparison

Model	MAE	RMSE	R^2	MAPE
Linear Regression	5.82	7.64	0.7856	15.23%
Ridge	5.84	7.65	0.7849	15.31%
Lasso	5.83	7.64	0.7853	15.26%
ElasticNet	5.85	7.65	0.7846	15.34%

Key Findings:

- Good performance ($R^2 \approx 0.79$)
- Distance to MRT station is strongest predictor
- Number of convenience stores positively correlates with price

4.6 Cross-Dataset Insights

- **Best Performing**: Energy Efficiency ($R^2 = 0.92$ for heating)
- **Most Challenging**: Appliances Energy ($R^2 = 0.54$)
- **Regularization Impact**: Minimal across all datasets
- **Model Consistency**: Linear regression is competitive baseline

5 Conclusion and Insights

5.1 Key Findings

This project successfully applied linear regression across four diverse real-world problems. Our key findings include:

5.1.1 Model Performance

- **Energy Efficiency**: Best performance ($R^2 = 0.92$ for heating load)
- **Concrete Strength**: Excellent performance ($R^2 = 0.89$)
- **Real Estate**: Good performance ($R^2 = 0.79$)
- **Appliances Energy**: Moderate performance ($R^2 = 0.54$)

5.1.2 Regularization Effectiveness

Ridge, Lasso, and ElasticNet regularization provided minimal improvement over standard linear regression across all datasets, suggesting:

- Low multicollinearity in feature sets
- Well-conditioned problems
- Appropriate feature engineering

5.1.3 Feature Insights

- **Concrete**: Age and cement content are primary strength drivers
- **Energy**: Building geometry (compactness, surface area) dominates
- **Appliances**: Complex patterns, multiple moderate contributors
- **Real Estate**: Location (MRT distance) is strongest predictor

5.2 Challenges and Limitations

5.2.1 Technical Challenges

1. **Large Feature Space**: Appliances dataset with 28 features required careful preprocessing
2. **Dual Target**: Energy efficiency required separate models for heating and cooling
3. **Limited Data**: Taiwan real estate dataset (414 samples) constrained model complexity

5.2.2 Model Limitations

- **Linearity Assumption**: Models assume linear relationships
- **Outlier Sensitivity**: OLS sensitive to extreme values
- **Single Train-Test Split**: No cross-validation performed
- **Fixed Hyperparameters**: Limited hyperparameter tuning
- **No Feature Engineering**: Polynomial/interaction terms unexplored

5.3 Future Improvements

5.3.1 Advanced Modeling

- Polynomial regression for non-linear relationships
- Ensemble methods (Random Forest, Gradient Boosting)
- Neural networks for complex patterns [1]

5.3.2 Validation and Optimization

- K-fold cross-validation for robust evaluation [5]
- GridSearchCV for hyperparameter optimization
- Feature selection (RFE, mutual information)

5.3.3 Deployment

- Web-based applications for real-time predictions
- Mobile interfaces for field use
- API development for system integration

5.4 Practical Value

The developed models demonstrate practical utility across three domains:

- **Construction**: Quality control, cost optimization, design support
- **Energy**: Smart home automation, consumption forecasting, grid management
- **Real Estate**: Property valuation, investment analysis, market insights

6 Application Description

6.1 Overview

Interactive prediction applications were developed for each problem using Jupyter notebooks with ipywidgets. These tools demonstrate how trained models can be deployed for practical use.

6.2 Main Features

6.2.1 User Interface Components

- **Input Widgets**: Sliders for continuous features, dropdowns for categorical variables
- **Real-Time Predictions**: Instant results as inputs change
- **Range Validation**: Inputs constrained to realistic ranges from training data
- **Visual Feedback**: Clear display of predictions with appropriate units

6.2.2 Supported Predictions

1. **Concrete Strength Calculator**: Input mix composition and age, predict compressive strength (MPa)
2. **Building Energy Estimator**: Input design parameters, predict heating/cooling loads (kWh/m²)
3. **Appliances Energy Forecaster**: Input sensor readings, predict energy consumption (Wh)
4. **Real Estate Valuator**: Input property features, predict price (TWD/Ping)

6.3 Input/Output Functionality

6.3.1 Input Processing

- **Data Validation**: Ensures inputs within acceptable ranges
- **Feature Scaling**: Applies StandardScaler transformation used during training
- **Error Handling**: Provides informative messages for invalid inputs

6.3.2 Output Generation

- **Prediction Display**: Shows predicted value with units
- **Confidence Indication**: Based on R^2 score of the model
- **Interpretable Format**: Results formatted for non-technical users

6.4 Model Integration

6.4.1 Architecture

The applications integrate trained models through:

1. **Model Loading**: Trained scikit-learn models loaded at startup
2. **Preprocessing Pipeline**: StandardScaler applied to inputs
3. **Prediction Generation**: Model generates predictions
4. **Result Formatting**: Outputs formatted and displayed

6.4.2 Technical Implementation

- **Framework**: Jupyter notebooks with ipywidgets
- **Libraries**: scikit-learn (models), pandas (data), matplotlib (visualization)
- **Interactivity**: Real-time widget callbacks for instant feedback

6.5 User Experience

6.5.1 Ease of Use

- **No Coding Required**: Users interact only with sliders and dropdowns
- **Immediate Feedback**: Predictions update instantly
- **Guided Input**: Labels explain each parameter

6.5.2 Practical Applications

- **Decision Support**: Engineers testing concrete mix designs
- **Design Exploration**: Architects evaluating building configurations
- **Energy Planning**: Homeowners forecasting consumption
- **Property Valuation**: Buyers and sellers estimating prices

6.6 Sensitivity Analysis

Each application includes sensitivity analysis features:

- **Feature Variation**: Observe how changing one feature affects predictions
- **Comparative Analysis**: Compare different scenarios side-by-side
- **Visualization**: Charts showing feature impact on target variable

6.7 Deployment Considerations

6.7.1 Current Implementation

- Jupyter notebook-based interactive widgets
- Local execution with Python environment
- Suitable for demonstrations and exploratory analysis

6.7.2 Future Deployment Options

- **Web Application**: Flask/Django with HTML/CSS/JavaScript frontend
- **Mobile App**: React Native or Flutter for smartphone access
- **Cloud Deployment**: AWS/Azure/GCP for scalability
- **API Service**: RESTful API for integration with other systems

7 References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN: 978-0387310732.

[2] Luis M. Candanedo, Véronique Feldheim, and Dominique Deramaix. "Data driven prediction models of energy use of appliances in a low-energy house". In: *Energy and Buildings* 140 (2017), pp. 81–97. DOI: 10.1016/j.enbuild.2016.12.005.

[3] C. Deepa, K. SathiyaKumari, and V. Pream Sudha. "Prediction of the Compressive Strength of High Performance Concrete Mix using Tree Based Modeling". In: *International Journal of Computer Applications* 6.5 (2010), pp. 18–24. DOI: 10.5120/1076-1406.

[4] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd. Springer, 2009. ISBN: 978-0387848570.

[6] Arthur E. Hoerl and Robert W. Kennard. “Ridge Regression: Biased Estimation for Nonorthogonal Problems”. In: *Technometrics* 12.1 (1970), pp. 55–67. DOI: 10.1080/00401706.1970.10488634.

[7] John D. Hunter. “Matplotlib: A 2D Graphics Environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[8] Gareth James et al. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. ISBN: 978-1461471370. DOI: 10.1007/978-1-4614-7138-7.

[9] Tarald O. Kvålseth. “Cautionary Note about R^2”. In: *The American Statistician* 39.4 (1985), pp. 279–285. DOI: 10.1080/00031305.1985.10479448.

[10] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[11] Nana Pow, Emily Janulewicz, and Lisa Liu. “Applied machine learning project in Python: predicting house prices”. In: *International Journal of Recent Engineering Research and Development* 3.7 (2018), pp. 85–94.

[12] The pandas development team. *pandas-dev/pandas: Pandas*. https://pandas.pydata.org/. 2020. DOI: 10.5281/zenodo.3509134.

[13] Robert Tibshirani. “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[14] Athanasios Tsanas and Angeliki Xifara. “Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools”. In: *Energy and Buildings* 49 (2012), pp. 560–567. DOI: 10.1016/j.enbuild.2012.03.003.

[15] Michael Waskom. *seaborn: statistical data visualization*. https://seaborn.pydata.org/. 2021. DOI: 10.21105/joss.03021.

[16] Yixing Wei et al. “A review of data-driven approaches for prediction and classification of building energy consumption”. In: *Renewable and Sustainable Energy Reviews* 82 (2018), pp. 1027–1047. DOI: 10.1016/j.rser.2017.09.108.

[17] Cort J. Willmott and Kenji Matsuura. “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance”. In: *Climate Research* 30.1 (2005), pp. 79–82. DOI: 10.3354/cr030079.

[18] I-Cheng Yeh. *Concrete Compressive Strength Dataset*. https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength. Accessed: 2025-11-15. 1998. DOI: 10.24432/C5PK67.

[19] I-Cheng Yeh and Tzu-Kuang Hsu. “Building real estate valuation models with comparative approach through case-based reasoning”. In: *Applied Soft Computing* 65 (2018), pp. 260–271. DOI: 10.1016/j.asoc.2018.01.029.

[20] Kaile Zhou, Chao Fu, and Shanlin Yang. “Big data driven smart energy management: From big data to big insights”. In: *Renewable and Sustainable Energy Reviews* 56 (2016), pp. 215–225. DOI: 10.1016/j.rser.2015.11.050.

[21] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320. DOI: 10.1111/j.1467-9868.2005.00503.x.