Homework 2: Predict House Pricing Using Machine Learning Algorithms

Due: October 15th, 2025, 5 pm EDT

Submission Requirements:

- **Report:** Step-by-step documentation explaining your process (installation, data download, libraries used, etc.).
- **Code:** Provide your machine learning code (in Jupyter notebook or text form) with well-commented explanations for each part.

Action Items:

Choose one of the following approaches — either replicate an existing machine learning pipeline or create your own solution:

Option A: Replicate an Existing ML Tutorial

Use established code from an online source applying **traditional machine learning** (e.g., Random Forest, Linear Regression) to predict house prices. A great example is this Kaggle notebook using Random Forest and XGBoost for house price prediction:

- **"House Price Prediction | Random Forest & XGBoost – Kaggle"** – Machine learning code applied to the House Prices – Advanced Regression Techniques dataset (Kaggle).

Use this as a basis:

1. Create your input pipeline
2. Load the dataset
3. Build a training pipeline
4. Build an evaluation pipeline
5. Train the model(s)
6. Build a testing pipeline
7. **(Optional)** Repeat steps adjusting hyperparameters such as number of trees, max depth.
8. **(Optional)** Try at least two different machine learning algorithms (e.g., Random Forest + Linear Regression or XGBoost) and compare results.

Option B: Build Your Own ML Solution

Implement from scratch using algorithms such as:

- Linear Regression
- Decision Tree or Random Forest
- SVM Regression
- Gradient Boosting (e.g., XGBoost)
- Others (e.g., Lasso, Ridge, k-NN, ensemble approaches)

Structure your work:

1. Input pipeline
2. Dataset loading and preprocessing (encoding, scaling, handling missing values)
3. Training pipeline
4. Evaluation pipeline
5. Model training
6. Testing pipeline
7. Hyperparameter tuning
8. Compare at least two algorithms

Notes:

- **Do NOT use deep learning** to solve this problem
- You can report the outcome of one or all classifiers used in your experiments
- The classification accuracy % is not going to be graded. HOWEVER - A reasonable accuracy is expected, typically above 85% minimum.