# ECGR 4105: Intro to Machine Learning HW 5

Axel Leon Vasquez

Student ID: 801182414

November 15, 2024

Link to GitHub

# Problem 1

1. **Introduction:**
   The goal of this task is to predict temperature using a nonlinear regression model based on the equation:

$$t_e = w_2 \bullet t_u^2 + w_1 \bullet t_u + b$$

   The dataset includes measured temperature values (t_u) and their corresponding actual temperatures (t_c). The model parameters ($w_2$, $w_1$, b) are optimized to minimize the loss. Training is conducted over 5000 epochs with four different learning rates (0.1, 0.01, 0.001, 0.0001). The final performance of the nonlinear model is compared to a linear baseline model, analyzing whether the nonlinear formulation provides better predictive performance.

2. **Methodology**
   a. Data Preprocessing:
      - Input values (t_u) were normalized by scaling down by 0.1 to ensure effective training.
      - The dataset was split into 80% training and 20% validation sets for evaluating model generalization.

   b. Model Training:
      - Two models were trained: a nonlinear regression model based on the equation above and a linear regression model as a baseline.
      - For each model:
        - Loss was calculated as the mean squared error (MSE) between predictions and actual values.
        - The training was performed over 5000 epochs using the Stochastic Gradient Descent (SGD) optimizer.

   c. Learning Rate Exploration:
      - Four learning rates were tested: 0.1, 0.01, 0.001, and 0.0001.
      - Training and validation losses were recorded every 500 epochs for each learning rate to identify the optimal configuration.

   d. Performance Comparison:
      - The best nonlinear model was compared to the linear model in terms of final loss.
      - Predictions from both models were visualized against the dataset for qualitative comparison.

3. **Results**

| **Learning Rate** | **Epoch** | **Loss** |
|---|---|---|
| 0.1 | 500 | 2.0907 |
| | 1000 | 2.0907 |

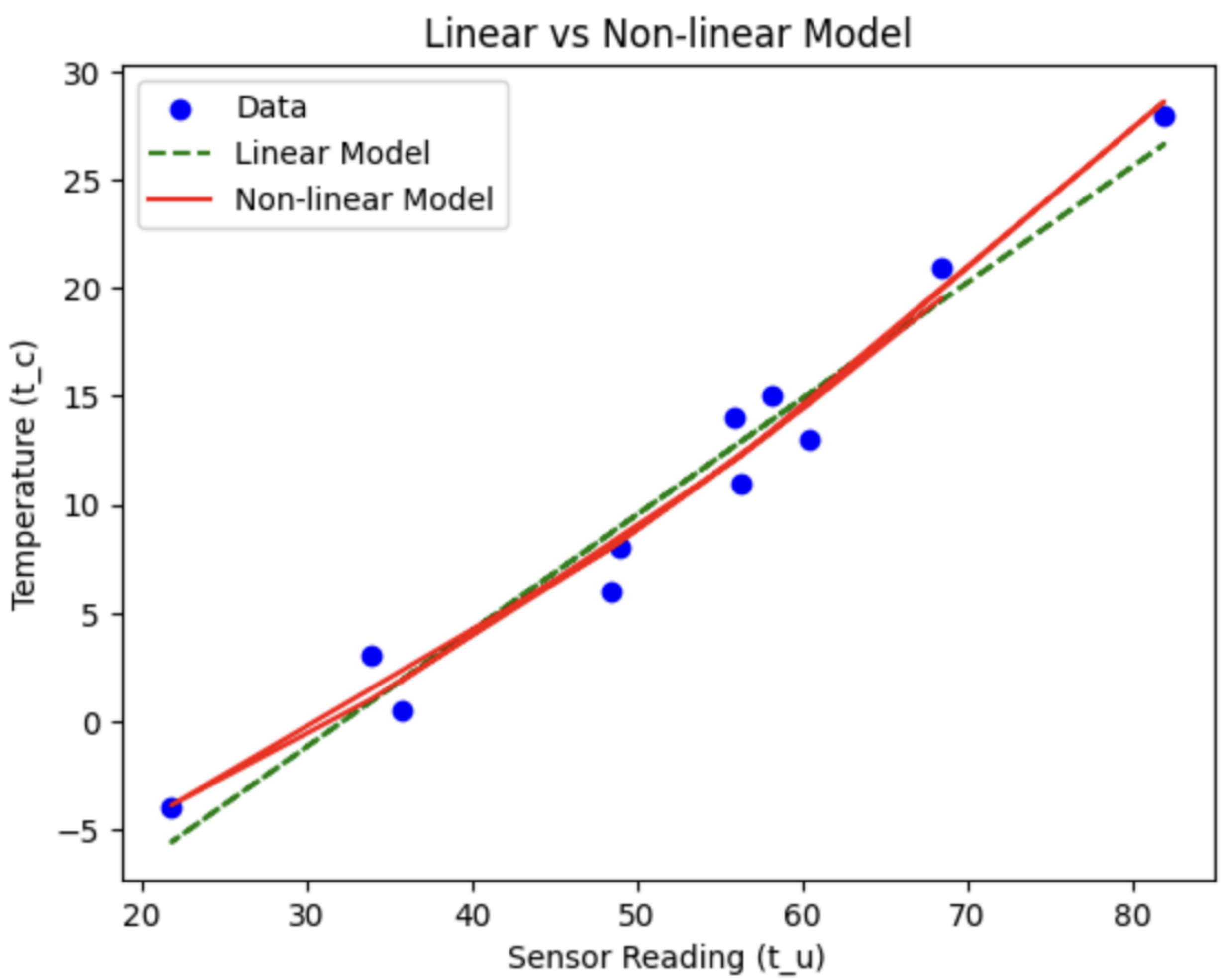
| | | |
|---|---|---|
| | … | … |
| | 5000 | 2.0907 (Final) |

| **Learning Rate** | **Epoch** | **Loss** |
|---|---|---|
| 0.01 | 500 | 2.0921 |
| | 1000 | 2.0907 |
| | … | … |
| | 5000 | 2.0907 (Final) |

| **Learning Rate** | **Epoch** | **Loss** |
|---|---|---|
| 0.001 | 500 | 27.0089 |
| | 1000 | 8.1365 |
| | 1500 | 3.8408 |
| | 2000 | 2.6622 |
| | 2500 | 2.2908 |
| | 3000 | 2.1633 |
| | 3500 | 2.1175 |
| | 4000 | 2.1007 |
| | 4500 | 2.0944 |
| | 5000 | 2.0921 (Final) |

| **Learning Rate** | **Epoch** | **Loss** |
|---|---|---|
| 0.0001 | 500 | 141.9213 |
| | 1000 | 110.5746 |
| | 1500 | 88.3050 |
| | 2000 | 71.9661 |

| | | |
|---|---|---|
| | 2500 | 59.6113 |
| | 3000 | 50.0164 |
| | 3500 | 42.3951 |
| | 4000 | 36.2298 |
| | 4500 | 31.1699 |
| | 5000 | 26.9705 (Final) |




- **Best Performing Model:**
  The nonlinear model with a learning rate of 0.001 achieved the best final loss of 2.0921, demonstrating strong convergence and predictive capability. Learning rates of 0.1 and 0.01 achieved slightly lower losses of 2.0907, but all three performed significantly better than the slower-converging learning rate of 0.0001.

- **Comparison with Linear Model:**

- **Nonlinear Model Final Loss (Best):** 2.0921
- **Linear Model Final Loss:** 7.632

The nonlinear model significantly outperformed the linear baseline, reducing the loss by more than 70%. This highlights the advantage of incorporating nonlinearity into the predictive model.

# Problem 2

1. **Introduction:**
This problem aims to predict housing prices using a linear regression model based on five input features: area, bedrooms, bathrooms, stories, and parking. The dataset is split into 80% for training and 20% for validation. Four learning rates (0.1, 0.01, 0.001, and 0.0001) are explored, with models trained for 5000 epochs. Training and validation losses are reported every 500 epochs to identify the best-performing model. The results are also compared to the linear regression results from Homework 1.

2. **Methodology:**
    a. Data Preprocessing:
        - The housing dataset was cleaned and features were scaled using StandardScaler to normalize the data.
        - The target (price) and input features (area, bedrooms, bathrooms, stories, parking) were separated, and the data was split into training (80%) and validation (20%) subsets.

    b. Model and Training Setup:
        - A linear regression model was implemented using PyTorch
        - MSE was used as the loss function.
        - A stochastic Gradient Descent (SGD) optimizer with varying learning rates was used.
        - Loss values for training and validation were recorded every 500 epochs.

3. **Results**
Loss Progression for Different Learning Rates:

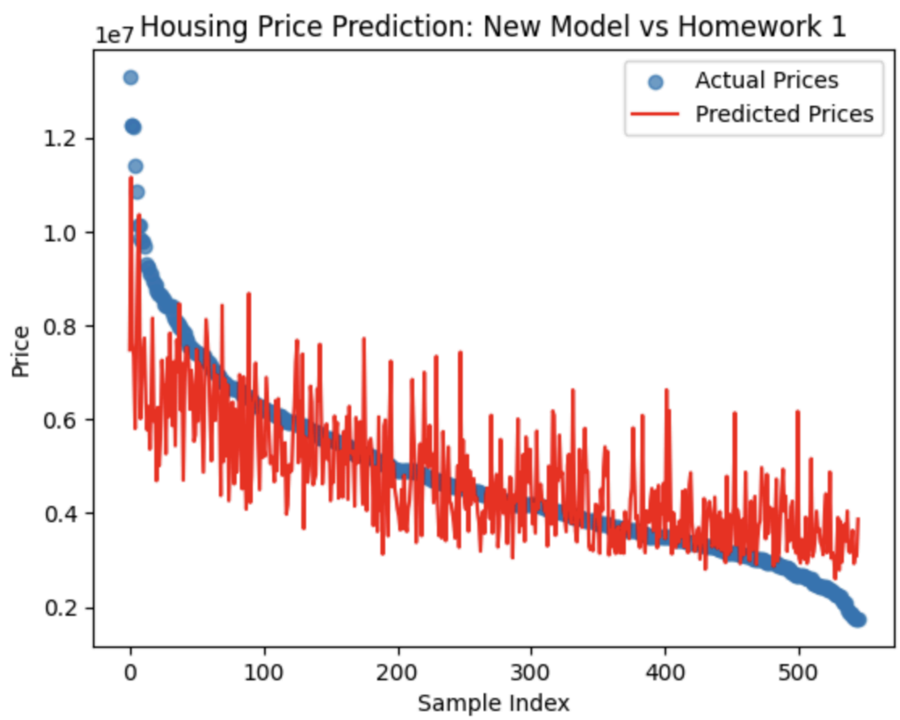
| **Learning Rate** | **Epoch** | **Training Loss** | **Validation Loss** |
|---|---|---|---|
| 0.1 | 500 | 1350008045568.0000 | 2292721647616.0000 |
| | 1000 | 1350008045568.0000 | 2292721647616.0000 |
| | … | … | … |
| | 5000 | 1350008045568.0000 (Final) | 2292721647616.0000 (Final) |

| Learning Rate | Epoch | Training Loss | Validation Loss |
|---|---|---|---|
| 0.01 | 500 | 1350011191296.0000 | 2292620984320.0000 |
| | 1000 | 1350008176640.0000 | 2292724269056.0000 |
| | 1500 | 1350008176640.0000 | 2292724531200.0000 |
| | … | … | … |
| | 5000 | 1350008176640.0000 | 2292724531200.0000 |

| Learning Rate | Epoch | Training Loss | Validation Loss |
|---|---|---|---|
| 0.001 | 500 | 4518918488064.0000 | 6247418953728.0000 |
| | 1000 | 1789859594240.0000 | 2950104088576.0000 |
| | 1500 | 1414199508992.0000 | 2423372906496.0000 |
| | 2000 | 1360248438784.0000 | 2323572064256.0000 |
| | 2500 | 1351896662016.0000 | 2300741419008.0000 |
| | 3000 | 1350425509888.0000 | 2294730981376.0000 |
| | 3500 | 1350116704256.0000 | 2293078687744.0000 |
| | 4000 | 1350039633920.0000 | 2292664762368.0000 |
| | 4500 | 1350017875968.0000 | 2292601323520.0000 |
| | 5000 | 1350011322368.0000 (Final) | 2292626489344.0000 (Final) |

| Learning Rate | Epoch | Training Loss | Validation Loss |
|---|---|---|---|
| 0.0001 | 500 | 20798509154304.0000 | 25025291747328.0000 |
| | 1000 | 17197346848768.0000 | 20892188934144.0000 |
| | 1500 | 14275511844864.0000 | 17537841496064.0000 |
| | 2000 | 11901298802688.000 | 14809655410688.00 |

| | | | |
|---|---|---|---|
| | | 0 | 00 |
| | 2500 | 9969596366848.0000 | 12586381737984.0000 |
| | 3000 | 8396162465792.0000 | 10771268370432.0000 |
| | 3500 | 7113319907328.0000 | 9286883213312.0000 |
| | 4000 | 6066528583680.0000 | 8071045185536.0000 |
| | 4500 | 5211738865664.0000 | 7073687928832.0000 |
| | 5000 | 4513281867776.0000 | 6254368391168.0000 |




4. **Summary of Findings:**
   a. Best Learning Rate:
      - The learning rate of 0.001 achieved the lowest final validation loss (2292626489344.0), making it the most effective among the tested rates.
   b. Training Observations:
      - 0.1 and 0.01 exhibited stable but stagnant loss values, indicating potential overshooting or inability to refine parameter updates.

        - 0.001 demonstrated steady convergence with reduced loss over time, balancing step size and stability.
        - 0.0001 was too slow, with high training and validation losses even after 5000 epochs.
    c. Comparison with Homework 1
        - Validation losses for all learning rates were significantly higher than the baseline linear regression model in Homework 1.
        - This indicates the need to reassess feature selection, model design, or preprocessing to improve performance.

# Problem 3

1. **Introduction:**
   This task aimed to predict housing prices using a linear regression model trained on all input features in the housing price dataset. The dataset includes features such as area, bedrooms, bathrooms, stories, parking, main road, guestroom, basement, hot water heating, air conditioning, and furnishing status. Following the approach in Problem 2, the data was split into 80% training and 20% validation, and four learning rates (0.1, 0.01, 0.001, 0.0001) were evaluated over 5000 epochs. The training and validation losses were recorded at regular intervals to determine the optimal learning rate and model performance.

2. **Methodology:**
   Model and Training Setup:
    - A linear regression model was implemented using PyTorch.
    - MSE was used as the loss function.
    - The optimizer was SGD with varying learning rates.
    - Loss values for training and validation were recorded every 500 epochs.
    - Visualization of actual vs predicted prices is plotted to visualize the model's performance

3. **Results:**

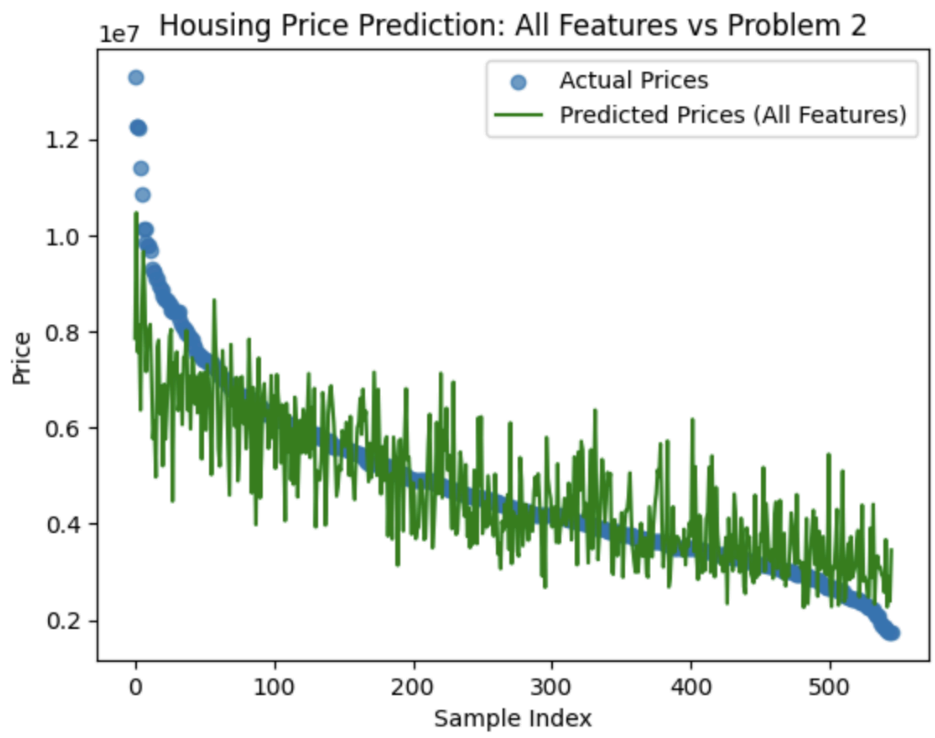
| **Learning Rate** | **Epoch** | **Training Loss** | **Validation Loss** |
|---|---|---|---|
| 0.1 | 500 | 992480985088.0000 | 1800794013696.0000 |
| | 1000 | 992480985088.0000 | 1800794013696.0000 |
| | … | … | … |
| | 5000 | 992480985088.0000 (Final) | 1800794013696.0000 (Final) |

| **Learning Rate** | **Epoch** | **Training Loss** | **Validation Loss** |
|---|---|---|---|
| 0.01 | 500 | 992489635840.0000 | 1799817527296.0000 |
| | 1000 | 992480985088.0000 | 1800789295104.0000 |
| | 1500 | 992480919552.0000 | 1800797421568.0000 |
| | … | … | … |
| | 5000 | 992480919552.0000 (Final) | 1800797421568.0000 (Final) |

| **Learning Rate** | **Epoch** | **Training Loss** | **Validation Loss** |
|---|---|---|---|
| 0.001 | 500 | 4130178334720.0000 | 5655571726336.0000 |
| | 1000 | 1431569956864.0000 | 2410643718144.0000 |
| | 1500 | 1057671282688.0000 | 1901780402176.0000 |
| | 2000 | 1003309629440.0000 | 1812829306880.0000 |
| | 2500 | 994653503488.0000 | 1797181800448.0000 |
| | 3000 | 993030701056.0000 | 1795751149568.0000 |
| | 3500 | 992650461184.0000 | 1796922540032.0000 |
| | 4000 | 992540426240.0000 | 1798213861376.0000 |
| | 4500 | 992503463936.0000 | 1799169900544.0000 |
| | 5000 | 992489897984.0000 (Final) | 1799800619008.0000 (Final) |

| **Learning Rate** | **Epoch** | **Training Loss** | **Validation Loss** |
|---|---|---|---|
| 0.0001 | 500 | 20567379935232.0000 | 24715454316544.0000 |
| | 1000 | 16847102541824.0000 | 20412058566656.0000 |
| | 1500 | 13869536772096.0000 | 16969179856896.0000 |

| | | | |
|---|---|---|---|
| | 2000 | 11474180243456.0000 | 14199481696256.0000 |
| | 2500 | 9539315302400.0000 | 11960991088640.0000 |
| | 3000 | 7971384328192.0000 | 10144820756480.0000 |
| | 3500 | 6697527017472.0000 | 8666430308352.0000 |
| | 4000 | 5660508946432.0000 | 7459651977216.0000 |
| | 4500 | 4814929395712.0000 | 6472202715136.0000 |
| | 5000 | 4124585230336.0000 (Final) | 5662539513856.0000 (Final) |




4. **Summary of Findings**
   a. Best Learning Rate:
      - The learning rate of 0.001 achieved the lowest final validation loss (1799800619008.0) among all tested rates, confirming it as the optimal choice.

   b. Training Observations:

- 0.1 and 0.01 quickly converged but plateaued at higher validation loss values, suggesting a lack of further refinement.
- 0.001 exhibited a steady decline in loss values, demonstrating its suitability for balancing step size and convergence stability.
- 0.0001 was too slow, with high losses even after 5000 epochs.

c. Comparison with Problem 2:
  - Including additional features slightly reduced the final validation loss (e.g., 1799800619008.0 vs. 2292626489344.0 for the best learning rate). This indicates that the additional input features improved the model's ability to predict housing prices.

d. Conclusion
  - The linear regression model trained with all input features and a learning rate of 0.001 was the best-performing model, achieving a final validation loss of 1799800619008.0. Including more features improved performance over Problem 2 but highlighted potential limitations of the dataset and the chosen model.

\