Final Project Write-Up

Hongyi Dai

DS210

Dec. 11th

**Introduction**

It's a house price forecasting application implemented in Rust using a mathematical model that analyzes associations among characteristics of houses, such as a number of bedrooms, bathrooms, square feet, and prices against which the model is made to predict. The model trains a dataset, builds a simple linear regression model, and investigates some prediction scores using various statistical criteria. This program is as minimalist as possible, favoring Rust's performance and its guarantee of safety.

**How the Program Works**

1. **Data Loading and Preprocessing**:

* The program reads a CSV file, NY-House-Dataset.csv, containing house data.
* Each entry in the dataset corresponds to a house. The columns are features (number of bedrooms, number of bathrooms, etc.) of the target price.
* Open the CSV file and parse each line in it. The program provides graceful handling of missing or invalid data, including only valid rows.
* Features extracted include: bedrooms, bathrooms, square footage, latitude and longitude. These are then normalised (scaled) to be in similar ranges, important for the maths in the program.
* Target values, i.e. house prices - these also undergo processing. No transformation is performed for interpretable prices of houses.

**Splitting the Dataset**:

* + The data is divided into two parts:
    - **Training Data**: Used to build the model, comprising 80% of the dataset.
    - **Testing Data**: Used to evaluate the model, comprising 20% of the dataset.
  + This separation ensures the program can assess how well the model performs on unseen data, a critical step for evaluating its effectiveness.

1. **Model Training**:

* The program trains a mathematical model using the training data. The model is a linear regression equation
* The weights get tuned in a process of gradient descent. This is basically an optimization technique where, with the help of iterative improvements in the weights, it minimizes the difference between actual and predicted prices.
* Training runs for a fixed number of iterations, or epochs, at the end of which it improves the weights incrementally.

**Evaluation**:

* + After training, the program evaluates the model using the testing data. Several statistical metrics are calculated:
* **Root Mean Squared Error (RMSE):** It measures the standard deviation of prediction errors. The smaller the RMSE, the better the performance.
* **Mean Absolute Error (MAE):** It is the average of the absolute differences between the forecasted and actual prices.
* **Mean Squared Error (MSE):** This will be defined as an average of a squared difference between predicted and actual prices.
* **R-squared (R²):** Explains the proportion of variance in the data explained by the model. It takes negative values for bad performance.
* **Mean Absolute Percentage Error (MAPE):** This expresses the error as a proportion of actual values and therefore provides a relative measure of performance.

1. **Visualization**:
   * The program generates a scatter plot comparing actual vs. predicted prices, giving a visual representation of how closely predictions align with real values.
   * This plot is saved as a PNG file (prediction\_plot.png) in the working directory.

**How to Use the Program**

1. **Setup**:
   * Install Rust and ensure the cargo package manager is available.
   * Place the dataset (NY-House-Dataset.csv) in the program's directory.
2. **Outputs**:
   * The program outputs the following metrics to the console:
     + Root Mean Squared Error (RMSE)
     + Mean Absolute Error (MAE)
     + Mean Squared Error (MSE)
     + R-squared (R²)
     + Mean Absolute Percentage Error (MAPE)
   * It also generates a scatter plot (prediction\_plot.png)

**What the Results Look Like**

1. **Statistical Metrics**:
   * Root Mean Squared Error (RMSE): 13.3969
   * Mean Absolute Error (MAE): 13.1395
   * Mean Squared Error (MSE): 179.4775
   * R-squared (R²): -15.8872
   * Mean Absolute Percentage Error (MAPE): 98.01%
2. **Sample Console Output**:
3. **Visualization**:

* In an ideal case, points lie in a scatter plot on either the axis of actual house price versus a corresponding plot axis with house prices predicted, ideally all perfectly along the diagonal.

**Key Notes**

• Negative R²: It means the model is not good enough to explain the variance in data. One may try using other features, enhance data preprocessing, or test other methods for better performance.  
• It is a fundamental program to learn and experiment, which helps understand how simple models can be used to predict real-life phenomena.