Claude Shyaka

ID#: 801326243

ECGR-5105

**Homework 1: Linear Regression with Gradient Descent Algorithm**

**GitHub Repo:**

In this exercise, a linear recession model with gradient descent algorithm was developed and used to estimate the price of houses. The US Housing dataset was used to train the model. 80% of the data was used for training and 20% was used for validation.

A screenshot of a computer

Description automatically generated with medium confidence

Table 1: Preview from of the dataset.

1. Without using Normalization/standardization
   1. In this section, housing prices were predicted based on the following input variables: area, bedrooms, bathrooms, stories, parking. After training and evaluation with a **learning rate of 1x10-9** and **1000 iterations**, the following result were obtained:
      1. Final theta vector: **[0.525, 837.930, 1.835, 0.941, 1.438, 0.413]**
      2. Final training loss: **1.589x1012**
      3. Final validation loss: **2.034x1012**

Figure 1 shows the training and validation loss curve for 1000 iterations.

* 1. Next, the following input variables were used: area, bedrooms, bathrooms, stories, mainroad, guestroom, basement, hotwaterheating, airconditioning, parking, prefarea. Results obtained are as follows:
     1. Final theta vector:
     2. Final training loss:
     3. Final validation loss:

Figure 2 shows the training and validation loss curve for \_\_\_ iterations.

1. Using Normalization/Standardization
   1. For input variables: area, bedrooms, bathrooms, stories, parking. The following results were obtained.
      1. Using the MinMaxScaler to normalize the training and validation data and setting the learning rate to \_\_ for \_\_ iterations.
         * Final theta vector:
         * Final training loss:
         * Final validation loss:

Figure 3 shows the training and validation loss curve for \_\_\_ iterations.

* + 1. Using the StandardScaler to standardize the training and validation data and setting the learning rate to \_\_ for \_\_ iterations.
       - Final theta vector:
       - Final training loss:
       - Final validation loss:

Figure 4 shows the training and validation loss curve for \_\_\_ iterations.

* 1. For input variables: area, bedrooms, bathrooms, stories, mainroad, guestroom, basement, hotwaterheating, airconditioning, parking, prefarea. The following results were obtained.
     1. Using the MinMaxScaler to normalize the training and validation data and setting the learning rate to \_\_ for \_\_ iterations.
        + Final theta vector:
        + Final training loss:
        + Final validation loss:

Figure 5 shows the training and validation loss curve for \_\_\_ iterations.y

* + 1. Using the StandardScaler to standardize the training and validation data and setting the learning rate to \_\_ for \_\_ iterations.
       - Final theta vector:
       - Final training loss:
       - Final validation loss:

Figure 6 shows the training and validation loss curve for \_\_\_ iterations.

1. Using Normalization/Standardizations and adding Parameter Penalization for improve model performance on validation set.
   1. For input variables: area, bedrooms, bathrooms, stories, parking. The following results were obtained.
      1. Using the MinMaxScaler to normalize the training and validation data and setting the learning rate to \_\_ for \_\_ iterations.
         * Final theta vector:
         * Final training loss:
         * Final validation loss:

Figure 7 shows the training and validation loss curve for \_\_\_ iterations.

* + 1. Using the StandardScaler to standardize the training and validation data and setting the learning rate to \_\_ for \_\_ iterations.
       - Final theta vector:
       - Final training loss:
       - Final validation loss:

Figure 8 shows the training and validation loss curve for \_\_\_ iterations.

* 1. For input variables: area, bedrooms, bathrooms, stories, mainroad, guestroom, basement, hotwaterheating, airconditioning, parking, prefarea. The following results were obtained.
     1. Using the MinMaxScaler to normalize the training and validation data and setting the learning rate to \_\_ for \_\_ iterations.
        + Final theta vector:
        + Final training loss:
        + Final validation loss:

Figure 9 shows the training and validation loss curve for \_\_\_ iterations.y

* + 1. Using the StandardScaler to standardize the training and validation data and setting the learning rate to \_\_ for \_\_ iterations.
       - Final theta vector:
       - Final training loss:
       - Final validation loss:

Figure 10 shows the training and validation loss curve for \_\_\_ iterations.