Linear Regression models comparison

Topiltzin Hernández Mares

March 3, 2022

Abstract

Linear regression models can be easily implemented given its simplicity to program and test. But if an improvement in the performance and predictions is required, then a number of improvements are needed, such as hyperparameter optimization and data normalization.

1 Model Description

In the last year, the **LinearRegressionGD** model was implemented as a class project. In this model, the cost function optimization was done with **Gradient Descent (GD)**, and it remains the same. The **GD** was implemented with help of matrix multiplications using the library **numpy** in order to improve the performance of the model. Other than small optimizations on the multiplication of some numbers, the model stays the same as the original implementation.

2 Data Processing

In this section of the model is were most of the improvements are located. In the following subsections the changes are going to be described in detail, with a reason of why were needed.

2.1 Original Model

In the original model, the data processing process was really simple, this were the steps:

1. The data is extracted from the insurance.csv file.
2. All the non-smoker registers are removed from the dataframe.

3. The dataframe is divided into train and test data.

4. Both dataframes (train and test) are "normalized" with a simple multiplication to escalate the values.

5. Finally, when a prediction is needed, the scaling process is reverted with the predicted values.

With the raw data (unscaled), the model, even with a small learning rate, diverged easily with a few hundred iterations. This is why the scaling was needed. In the original implementation, the scaling is simple because of lack of time in the development of the model. In this second iterations, the main objective was to improve the scaling of the data to improve the training process.

2.2 New Model

After learning more about data scaling and normalization, the new data processing steps are the following:

1. The data is extracted from the insurance.csv file.

2. All the non-smoker registers are removed from the dataframe.

3. The *min-max* normalization method is applied to the cleaned dataframe.

4. The dataframe is divided into train and test data.

5. Finally, when a prediction is needed, the normalization process is reverted with the predicted values.

In the following sections, the improvements on the model is going to be explained in detail, but it can be said that the normalization method implemented in this new iteration improved the quality of the predictions.

3 Evaluation

Since this is a linear regression model, there are two main metrics that are easily comparable: *Mean Squared Error (MSE)* and *Coefficient of Determination (R2)*. These two metrics will be measured from the train and test processes from both models.

3.1 Method

For measuring both metrics for both models, it was decided to implement a script in order to simplify the process. The script *measure.py* was written, along with helper functions to reduce code duplication in the different training scripts for each model.

In this script, each model is trained 100 times, measuring each *MSE* and *R2* in each iteration. In every training, the number of learning iterations for the model was increased by 200. This was done in order to know the learning behavior of both models.

4 Results

After executing the experiment, the metrics were obtained correctly and the comparison between the two models was possible.

As can be seen in Figure 1, the new model has a faster learning behavior, since the *MSE* and *R2* reach their minimum much faster than the old model. It is also important noting that new the *R2* is much better than the one from the old model, this means the predictions are more accurate in the new model.

5 Conclusions

As can be seen in the Results section, the new model makes better predictions, this is because the data processing in this model is better than a simple scaling. When using data normalization, we allow our model to fit better to the training data and make more accurate predictions.

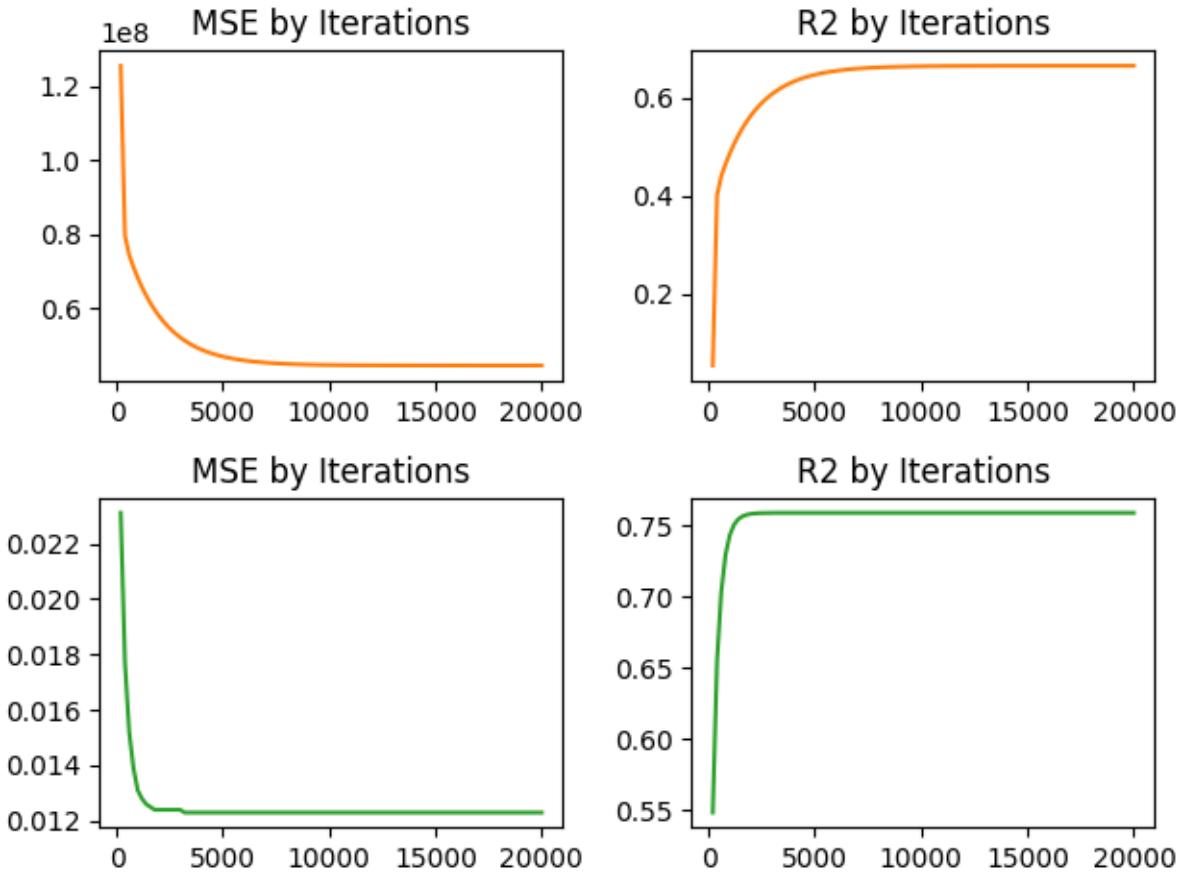

Figure 1: *MSE* and *R2* from both models in train process. Old model (orange) and new model (green).