UNIVERSIDAD POLITECNICA DE YUCATAN.

Subject: Machine Learning.

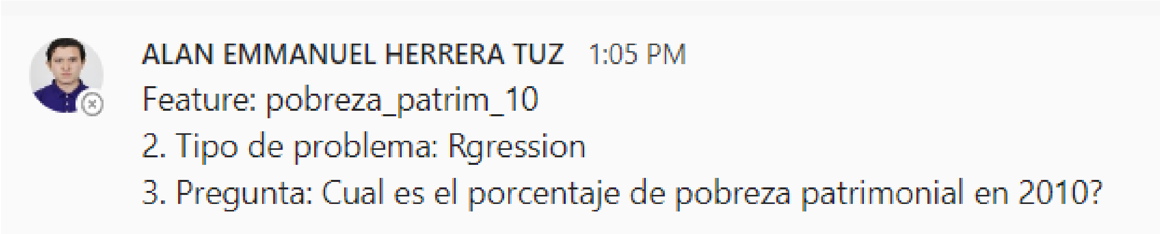
Assigment: Predictor from Sracth.

Professor: Victor Alejandro Ortiz

Student: Alan Emmanuel Herrera Tuz.

Deadline: 10/23/2023.

a) Problem proposal:



ALAN EMMANUEL HERRERA TUZ 1:05 PM

Feature: pobreza_patrim_10

2. Tipo de problema: Rgression

3. Pregunta: Cual es el porcentaje de pobreza patrimonial en 2010?

b) Getting dataset ready.

To clean the dataset, the first thing we did was to import the pandas library in order to eliminate unnecessary columns, for this process we first eliminated the columns that contained information about the name of the municipalities and their state keys, since they did not represent relevant data for the predictive model, this was achieved by using the drop function.

As a next step, in order to apply a downward gradient multilinear regression model, we had to contain only numerical values in our dataset, which was difficult since the columns 'gdo_rezsoc00', 'gdo_rezsoc05', 'gdo_rezsoc10' contained string values, each containing the following labels: very low, low, medium, high, very high.

To achieve the replacement of these values we chose to create 5 rows for each existing column, which would contain boolean values, i.e. for 'gdo_rezsoc00' we added 'gdo_rezsoc00_very_low', 'gdo_rezsoc00_low, 'gdo_rezsoc00_medium, 'gdo_rezsoc00_high, 'gdo_rezsoc00_very_high, this was achieved by using the get.dummies function.

c) Training the predictor.

1. Gradient Descent Algorithm:

The chosen method for training the predictive model is the gradient descent algorithm. This algorithm is well-suited for adjusting the model's parameters (weights w and bias b) in order to minimize the prediction error. It iteratively updates these parameters based on the gradient (derivative) of the loss function, effectively moving towards the optimal values.

2. Multilinear Regression Model:

Given that there are 145 features in the dataset, it was decided to use a multilinear regression model. This model type is ideal for situations where multiple features contribute to the prediction of a target variable. It considers a linear combination of all features, each weighted by a corresponding coefficient, to make predictions.

3. Predictive Model Function fw,b(x) = wx + b:

The chosen predictive model is represented by the function $f(w, b, x) = wx + b$. In this equation, w represents the weights associated with each feature, x denotes the feature values, and b is the bias term. The model essentially computes a weighted sum of the features, which is then adjusted by the bias.

4. Target Variable Selection: 'poverty_patrim_10':

The target variable chosen for prediction is 'poverty_patrim_10'. This indicates that the model's purpose is to estimate values of this particular variable based on the input features. The algorithm will learn how each feature contributes to the variation in 'poverty_patrim_10'.

5. Data Split for Training:

The dataset is divided into x_train and y_train. The first 1964 samples (80% of the data) are used for training. This allows the model to learn the underlying relationships between the features and the target variable. The remaining samples are typically used for testing the model's performance.

6. Why Gradient Descent:

Gradient descent is chosen due to its effectiveness in minimizing the prediction error. It allows the model to iteratively adjust its parameters to find the optimal values that minimize the loss function.

In this case, as there are a large number of features (145), gradient descent efficiently handles the optimization process.

d) Evaluating the performance of the predictor.

To evaluate the performance of the predictor, a quadratic cost error function was used, which is normally used to evaluate cost prediction models, but since it is still dealing with numerical models, it works perfectly with multilinear and multigradient descending regression models, subsequently to create the quadratic error function, which is the sum of the difference of the prediction minus the real value squared multiplied by the total number of values between 2, the dataset was divided into 20% (x_test, y_test) to test it with the quadratic error function.

- The quadratic cost error function is commonly used in regression problems, as it is well-suited for evaluating models that aim to predict continuous numerical values, which is the case for your predictive model.
- It quantifies how well the model's predictions align with the actual target values.

Finally using the mentioned functions with a learning rate alpha of 0.000001 and using 10000 iterations in order to get the proper values for matrix w and b value, we got a really big square error, that’s probably because we have a lot on the dataset and also a lot of the features that we are using are not really related with the value that we aim to predict.

e) Predictor using libraries and differences:

Explanation:

Data Loading and Cleaning: The code begins by loading the dataset using Pandas and importing the required libraries.

Target Variable and Feature Selection: Both the target variable (y) and the features (X) are defined. This instance shows that Y has only the 'pobreza_patrim_10' column, but X contains all of the columns other than that one.

Standardization: Scikit-learn's StandardScaler is used to standardize the characteristics. By guaranteeing that all characteristics have comparable scales, standardization can enhance the model's convergence during training.

Train-Test Split: Using array slicing, the dataset is divided into training and testing sets. The samples from 1965 that come after are utilized for testing, and the initial ones are used for training.

Initializing and Training the Model: The fit() function is used to build an instance of the Linear Regression model, which is then trained on the standardized training data.

Prediction: The predict() method is used by the model to generate predictions on the test set.

Calculating the Mean Squared Error (MSE): The MSE is a metric used to assess how well a model performs on a test set.

Printing MSE: The console receives a print of the MSE value.

Why the results are so different?

- By offering pre-built functions for activities like model training, prediction, and assessment, the use of machine learning libraries streamlines implementation.
- To make sure that features are scaled similarly, standardization is used. This holds significance for models that depend on the relative scales of features, such as Linear Regression.
- Weight initialization, optimization, and other aspects are handled internally by the scikit-learn Linear Regression model, which can result in more reliable and effective training.
- Scikit-learn is a flexible toolkit that may be used for a variety of regression and classification problems because it offers a large selection of machine learning methods.
- Because they take advantage of years of community-driven development and optimization, well-established libraries frequently provide more effective and efficient code.

f) Git-hub link of the project:

https://github.com/AlanHerreraTz/Machine-Learning_Codes.git

g) Main challenge and robotics applications:

My biggest challenges during the development of the project was to understand the mathematics behind linear regression predictive models, in this case I applied a descending gradient as a function of self-adjustment of weights and the value of b, for this I had to investigate various sources with the purpose of having the necessary knowledge to carry out the project in addition.

To solve problems such as cleaning the dataset, I had to research the appropriate functions to carry out this process.

Robotics applications:

- A basic optimization method used in machine learning model training is the gradient descent algorithm. It can be used to train models for robotics tasks such as object detection and path planning.
- Multilinear Regression Model: This kind of model is useful in robotics applications where the goal is to forecast a numerical result based on several input features. For instance, forecasting a robotic arm's location by analyzing many sensor readings.
- Function of Predictive Model $fw,b(x) = wx + b$: This illustrates a linear model that can be used in situations (like some control systems) where a linear relationship between input attributes and output is suitable.