Predictor from Scratch

Diego Armando May Pech
Computational robotics
Universidad Politécnica de Yucatán
Km. 4.5. Carretera Mérida — Tetiz
Tablaje Catastral 7193. CP 97357
Ucú, Yucatán. México
Email: 2009092@upy.edu.mx

Victor Alejandro Ortiz Santiago
Universidad Politécnica de Yucatán
Km. 4.5. Carretera Mérida — Tetiz
Tablaje Catastral 7193. CP 97357
Ucú, Yucatán. México
Email:victor.ortiz@upy.edu.mx

Abstract

This project aimed to develop regression models for predicting social backwardness in municipalities, utilizing poverty indicators as input features. The study leveraged Google Coolab to construct two distinct models: one built from scratch and the other using the popular scikit-learn library. The primary objective was to evaluate the effectiveness of these models in generating accurate predictions. The results indicated that both models struggled to provide precise predictions, with deviations from actual values. However, the model constructed with the scikit-learn library displayed a more favorable average prediction performance in comparison to the one created from scratch. This outcome underscores the importance of leveraging established machine learning libraries, suggesting that they offer more robust tools for addressing complex regression problems. Despite the limitations in predictive accuracy, this project contributes to the ongoing discourse on social backwardness and poverty indicators, emphasizing the potential for data-driven approaches to shed light on these critical issues. Future work in this domain should focus on refining feature engineering and exploring additional modeling techniques to further enhance predictive outcomes.

October 22, 2023 © Universidad Politécnica de Yucatán

Predictor from Scratch

I. Introduction

The purpose of this project was to create a model from the ground up to gain a better understanding of how algorithms operate within each model. We utilized a data-set of poverty indicators due to its extensive range of features, allowing each study participant to address a different problem using one of these features as the target variable (Y) to predict. In my case, I selected a column that pertains to individuals experiencing social backwardness in each mentioned municipality. Hence, the problem at hand falls under linear regression, as we aim to predict a numerical value based on various input features.

In simple terms, the model seeks to establish a linear relationship between the input features (independent variables) and the target variable (dependent variable). It does this by fitting a straight line to the data points, aiming to minimize the difference between the predicted values and the actual observed values. This line represents the best approximation of how the features collectively influence the target variable. By using this line, we can make predictions about the target variable based on new sets of input features. In essence, linear regression attempts to find the "best-fit" line that summarizes the relationship between the features and the variable we want to predict, allowing us to make informed estimations.

II. Development of the activity

1) selecting the target : The target to be chosen was one that shows the number of people with social backwardness in each municipality. This decision was personal and was the basis for deciding the type of problem to be addressed and the way to solve it, in this case a regression problem. linear

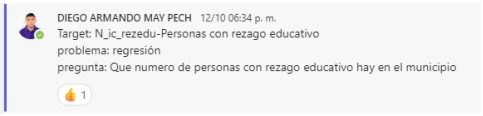

Fig. 1. Target selection.

2) Data preparation:

- Data Loading: The dataset was loaded from a local file using the pandas library. This action allowed for an initial inspection of the dataset to ensure its proper import.
- Feature Selection: Specific columns were identified for exclusion from the analysis. These columns include information not relevant to the study, such as 'ent', 'nom-ent', 'mun', 'nom-mun', 'mun-key' and 'N-ic-rezedu'. These columns are not useful because they only give information about the municipality and my main focus is on the characteristics that influence school backwardness and where people live at the moment is not useful for my model.
- Encoding and Handling Missing Values: Label encoding was applied to categorical features ('gdo-rezsoc00', 'gdo-rezsoc05', 'gdo-rezsoc10') to convert them into numerical representations. Additionally, a simple imputation strategy was used to handle missing values by replacing them with the mean of the available data.
- Feature Scaling: To normalize features and ensure they had a mean of 0 and a standard deviation of 1, standard scaling was applied using the 'StandardScaler' library
- Data Splitting: The dataset was divided into training and testing sets in an 80-20 proportion using the 'train-test-split' function.

3) Predictor Training:

1) Gradient Descent Algorithm: A custom gradient descent algorithm was implemented to train the predictor. This algorithm iteratively adjusted the weights and bias to minimize the mean squared error (MSE) in both the training and testing datasets.

Key Components:

- Initialization: The process begins with the initialization of the weights and the bias to zero. These values serve as the starting point for the learning process.
- Training Iterations: The algorithm iteratively updates the model's parameters. Each iteration involves the following steps:
- Prediction: The algorithm calculates predictions for the target variable (in this case, the number of people with social backwardness) using the current weights and bias.
- Error Computation: The MSE, a measure of the average squared difference between the predicted values and the actual values, is computed for both the training and testing datasets. This error quantifies how far off the predictions are from the true values.
- Gradient Calculation: The gradients of the weights and the bias with respect to the MSE are calculated. These gradients represent the direction and magnitude of the steepest ascent in the error surface. The goal is to move in the opposite direction to minimize the error.
- Parameter Updates: The weights and bias are updated by subtracting a fraction of the gradients (determined by the learning rate) from their current values. This update moves the parameters in the direction that reduces the error.
- Convergence: The algorithm continues these iterations for a specified number of epochs or until convergence is achieved, meaning that the MSE reaches a minimum or a sufficiently small value.

4) Performance Evaluation:

- Error Metric: The predictor's performance was evaluated using the Mean Squared Error (MSE). This metric quantified the accuracy of predictions in comparison to actual values.

- Custom Predictions: The prediction process for a randomly selected municipality was illustrated, providing the predicted number of individuals with social backwardness and comparing it to the actual value.

5) Training the Predictor Using scikit-learn:

1) Model Selection and Training:

- A linear regression model from scikit-learn was selected and instantiated as modelo-regresion.
- Training the Model:
- The model was trained on the training data using the fit method, where it learned the relationships between the input features and the target variable.
- Making Predictions:
- Predictions for the test dataset were generated using the predict method, and these predicted values were stored in the variable y-pred.
- Evaluation:
- Model performance was assessed using metrics such as Mean Squared Error (MSE) and R-squared (R^2) calculated from the predictions and actual target values.

III. Results

A. Model Comparison

Both models, the one built from scratch and the one implemented using the scikit-learn library, exhibited significant discrepancies between the predicted values and the actual values. However, the scikit-learn-based model demonstrated better performance in terms of prediction accuracy when compared to the model built from scratch.

Reasons for the Better Performance of the scikit-learn Model:

- Optimized Implementation: Scikit-learn regression models are highly optimized and efficient in terms of performance. They are designed to effectively handle regression tasks and provide a comprehensive toolkit that includes data preprocessing, feature selection, and model evaluation.
- Effective Learning Strategies: Scikit-learn models employ effective and proven learning strategies, including regularization, optimization, and overfitting control.
- Better Data Preprocessing: Data preprocessing, such as label encoding and handling missing values, is executed more efficiently within the scikit-learn model. This can improve the quality of the input data for the regression model.
- Use of Optimized Gradient Descent Routines: Scikit-learn incorporates highly optimized gradient descent routines, enabling faster and more precise model training.
- Improved Hyperparameter Selection: Scikit-learn allows for hyperparameter tuning and cross-validation, which can result in better model configuration and, consequently, better performance.

IV. Conclusion

In this project, two approaches were undertaken to address a regression problem related to poverty indicators. The first approach involved building a regression model from scratch, and the second utilized the scikit-learn library. The primary objective was to predict the number of people with social backwardness in municipalities using available indicators. Upon conducting these experiments, it became evident that the scikit-learn-based model outperformed the one built from scratch. The scikit-learn model demonstrated superior efficiency, optimized data preprocessing, and effective learning strategies. It achieved a better alignment between predicted values and actual outcomes, making it a more reliable predictor for this particular regression task.

- Challenges in Data Handling for the Custom Model

One notable challenge encountered in this project was the complexity of working with the dataset during the manual model construction. The data often lacked the requisite features or was in a form that necessitated extensive transformations or truncations. This data preprocessing aspect presented difficulties in terms of ensuring data quality and relevance.

- Application in Robotics

This modeling and prediction process is not only relevant for data analysis, but also has applications in robotics. The ability to build and train accurate models can enable autonomous robots to make informed decisions in dynamic and unknown environments. For example, robots can use regression models to predict changes in environmental conditions and adjust their behavior accordingly. This is especially valuable in autonomous robotics applications, such as navigation, decision making, and interacting with changing environments.