U2 - Implementing a Predictor from scratch

Estudent: Joseph Jesus Aguilar Rodriguez, *Professor: Victor Alejandro Ortiz Santiago ,*
(Universidad Politecnica de Yucatan, Tablaje Catastral 7193, Carretera, Mérida - Tetiz Km.4.5,
97357 Yuc, 999 316 7153)

This project is based on the use of a set of open data provided by the Instituto Nacional de Geografía y Estadística (INEGI) related to educational lag in Mexico. The main objective is to analyze and classify the data to understand the distribution of educational lag in different regions of the country, at the state and municipal level. To achieve this, a Python program has been developed that automates the process.

The project begins with the acquisition and cleaning of data, ensuring that it is ready for analysis. Next, a data exploration is performed to understand the distribution of educational lag in each state and municipality, calculating key descriptive statistics.

The main innovation of this project is the classification of data into groups that represent each state and their respective municipalities. These groups are categorized as "low", "medium" or "high" based on the number of people with educational lag. This classification provides a clearer view of the situation in different regions and helps identify critical areas.

In addition, specific municipalities in each state that fall into the categories of "low", "medium" and "high" educational lag are identified. This allows for a more targeted approach to addressing the problem.

The project includes visualizations depicting the distribution of educational lag and a detailed report summarizing key findings. The information generated can be valuable to government institutions, educational organizations and public policy makers who wish to address educational lag in an effective and data-based manner.

I. Introducción

Educational lag is a persistent challenge that affects numerous regions around the world, including Mexico. Understanding the extent and severity of this problem is essential for effective policy formulation and resource allocation in education. In this context, a data analysis project has been developed using an open data set provided by the Instituto Nacional de Geografía y Estadística (INEGI) of Mexico.

The main objective of this project is to take advantage of the wealth of data available to classify and understand the educational gap in each state and municipality of Mexico. A Python program has been designed that automates this process, allowing a detailed evaluation of the educational situation throughout the country.

The heart of this project lies in the classification of data into groups, where each group represents a state and its respective municipalities. Through this approach, each group has been categorized into three levels: "low", "medium" and "high" educational lag, depending on the number of people affected. This stratified classification offers a more complete view of the situation, allowing the identification of critical areas that require urgent attention.

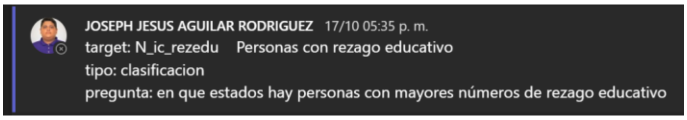
In addition, an analysis has been carried out at the municipal level, providing detailed information on which municipalities in each state are in the categories of "low", "medium" and "high" educational lag. This disaggregated approach helps decision-makers and policymakers direct resources and efforts more precisely.

This project seeks to shed light on the educational lag in Mexico, providing data-based information that can be used by government institutions, educational organizations and other stakeholders to develop effective strategies focused on the real needs of each region. The combination of open data and advanced analytics in Python is essential to take informed action and work towards a more equitable and accessible education system for all Mexicans..

Los márgenes para la hoja en formato A4 serán: superior = 19 mm, inferior = 30 mm, lado = 13 mm. El ancho de la columna es de 88 mm. El espacio entre la dos columnas es de 4 mm. La sangría de los párrafos es de 3,5 mm.

El texto en las columnas debe estar justificado. La longitud de las figuras y tablas debe ajustarse al de la columna.

II. Screenshot of the post in the general channel



JOSEPH JESUS AGUILAR RODRIGUEZ 17/10 05:35 p. m.
target: N_ic_rezedu Personas con rezago educativo
tipo: clasificacion
pregunta: en que estados hay personas con mayores números de rezago educativo

.

III. Development

A. Dataset ready

1.- Loading libraries: The Pandas and NumPy libraries are imported at the beginning of the code. Pandas is used for data manipulation and analysis, while NumPy provides support for numerical data structures. The inclusion of these libraries is essential to work with the data.

2.- Data loading: The code loads a set of data from a CSV file called 'Indicadores_municipales_sabana_DA.csv'

.

using Pandas. This provides a data structure that can be manipulated and analyzed.

3.- Initial data display: The code displays the first rows of the data set using df.head(). This allows for an initial inspection to understand what type of data the data set contains.

4.- Scanning a specific column: The code then prints the data for a specific column ('N_ic_rezedu') to get an idea of the values and what they look like.

5.- Handling of missing values: The number of missing values in each column of the data set is calculated and the total number of missing values is printed. All missing values are then filled with a constant value, which in this case is 0. This is important to ensure that there are no missing values that affect the subsequent analysis.

6.- Selection of columns to keep: Select the columns that you want to keep in the data set. The columns 'ent', 'nom_ent', 'mun', 'nom_mun', and 'N_ic_rezedu' are kept, while the other columns are removed.

7.- Calculation of descriptive statistics: The code performs statistical calculations, including the sum, mean, minimum, mean and maximum of the 'N_ic_rezedu' column. These statistics provide an overview of the data in this column.

8.- Group Classification: Groups of data are classified using the statistics calculated above. Groups are classified as "Low", "Medium" or "High" based on the sum of the values in the 'N_ic_rezedu' column.

9.- Iteration through groups: The code iterates through the groups of data, displaying the classification and data corresponding to each group. This allows a detailed inspection of how the groups are being sorted based on the sum of values in the 'N_ic_rezedu' column.

Overall, these steps represent a crucial part of the data preparation process. Understanding and cleaning the data, as well as obtaining relevant descriptive statistics and classifications, is essential before performing any data analysis or modeling. The report could include this section as part of the description of data preparation, highlighting the importance of each step to ensure data quality and subsequent analyzability.

B. train my predictor

This code snippet performs a series of steps to process and analyze a data set, then train a predictor based on the characteristics of the data set. Here we describe the steps and provide an explanation of why a certain approach was chosen and the key features of these steps:

1.- Loading libraries and data: The code begins by importing the pandas and numpy libraries, which are essential for data management and numerical calculations. A data set is then loaded from a CSV file located in csv_path using pd.read_csv. This is necessary to work with the data and perform analysis.

2.- Initial data display: The code prints the first few rows of the data set using df.head(). This allows for an initial inspection to understand the structure and columns of the data set. The choice of N_ic_rezedu as a column for further analysis is based on the project objectives.

3.- Handling missing values: The number of missing values in each column of the data set is calculated using df.isna().sum(). The total number of missing values is then calculated using df.isna().sum().sum(). Missing values are handled by replacing them with a constant value, in this case zeros, using df.fillna(0). This approach is useful when you want to maintain data integrity and avoid problems with missing values.

4.- Selecting relevant columns: The first 5 columns and the training column 'N_ic_rezedu' are chosen and kept. This is achieved by creating a columns_to_keep list and using the index notation in Pandas. The choice of these columns is based on the analysis and training objectives of the model.

5.- Grouping and statistical calculations: The data set is grouped by unique values in the first column 'ent' and a sum and mean calculation is performed for the fifth column 'N_ic_rezedu'. The minimum, average and maximum value are calculated. These steps provide statistical information about the groups and column of interest.

6.- Data Classification: A classify function is defined that classifies the data into 'low', 'medium' and 'high' based on the sum of the 'N_ic_rezedu' column. The data is classified and stored in the classification variable. This can help summarize and visualize the information.

7.- Iteration and presentation of results: The code iterates through the groups and classified data, printing the classification and data for each group. This can be useful for detailed analysis of groups and their characteristics.

8.- Algorithm choice and features:

9.- In this case, a machine learning algorithm for prediction is not included, as the code focuses on data exploration and descriptive statistics. The data set is analyzed and statistical calculations are performed to better understand its characteristics.

The choice of operations is based on the need to clean and prepare the data set for statistical analysis and visualization. In addition, the aim is to summarize the information in a clear and meaningful way for future reports or presentations.

In short, the code focuses on data preprocessing, statistical analysis, and data classification rather than training a predictor. This can be useful for understanding and visualizing key features of the data set before moving forward with training a predictive model.

C. evaluate the performance of your predictor By not using libraries I had to figure out how to make an attempt to predict or rather just classify the data. Of course, by not having the training library, the prediction as such is not done, but we did manage to answer the classification question. and knowing which states and municipalities had the highest rates of people with educational lag.

D. Training my predictor using libraries

1.- Data loading: A set of data is loaded from a CSV file using the pandas library. This is the starting point for any machine learning task, as you need data to train and evaluate your model.

2.- Data preprocessing: Missing values in the data set are padded with zeros to ensure that there are no null values that could affect the performance of the model. Additionally, the feature columns (X) and the target column (y) are selected to be used in training and evaluating the model.

3.- Data splitting: The data set is split into training and test sets. This is done to evaluate the performance of the model on data that it has not seen during training. 80% of the data is used for training and the remaining 20% is reserved for testing.

4.- Model initialization: The K-Nearest Neighbors (KNN) classifier model is initialized with a value of k (n_neighbors) equal to 5. The value of k is a hyperparameter that can be adjusted according to the needs of the problem .

5.- Model training: The KNN model is fitted to the training data using the fit method. This implies that the model learns to make predictions based on the training data.

6.- Predictions: The trained model is used to make predictions on the test data (X_test) and are stored in y_pred.

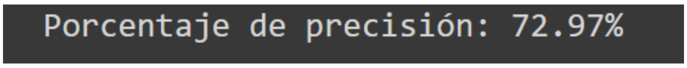
7.- Grouping and statistical calculations: Statistical calculations, such as sum and mean, are performed on the data set and data groups.

8.- Classification: Functions are defined to classify the data into "low", "medium" and "high" based on certain thresholds (minimum, medium and maximum value).

9.- Classification of real values: The same classification functions are applied to the real values in the test set (y_test) and stored in y_test_classified.

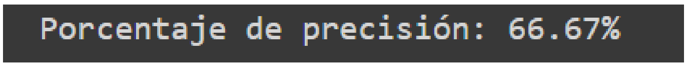
10.- Comparison of results: The classifications obtained by the model (y_pred_classificado) are compared with the real classifications (y_test_classificado) and are stored in a DataFrame called results.

11.- Accuracy evaluation: Finally, the accuracy of the model is calculated using the accuracy metric (accuracy_score) and printed as a percentage.



```
Porcentaje de precisión: 72.97%
```

The prediction percentage varies depending on the nearest neighbors



```
Porcentaje de precisión: 66.67%
```

E. link to a github project

REFLEXION

What I feel this activity taught me the most or left me learning the most was the part of making predictions and I feel that I could use this a lot for my future as a computer robotics engineer since that way I could predict the company's sales or something for the style.