

U2 - Implementing a Predictor from scratch

Axel Esdras De Flores Islas
Robotics Engineering
Universidad Politécnica de Yucatán
Km. 4.5. Carretera Mérida — Tetiz
Tablaje Catastral 7193. CP 97357
Ucú, Yucatán. México
Email: student@upy.edu.mx

Victor Ortiz / Machine Learning
Universidad Politécnica de Yucatán
Km. 4.5. Carretera Mérida — Tetiz
Tablaje Catastral 7193. CP 97357
Ucú, Yucatán. México
Email: professor@upy.edu.mx

Abstract

This report presents a data analysis and prediction study applied to a dataset of socioeconomic indicators. The goal of this project is to assess and predict the incidence of deficiencies within a specific population using linear regression techniques.

In the coding phase, data processing was carried out, including data cleaning, transformation, and encoding of categorical variables into a suitable format for modeling. Missing value management was also addressed through an imputation strategy.

The dataset was divided into training and testing sets, with 80 of the data used for model training and the remaining 20 for evaluating model performance. A linear regression model was implemented and trained with the training data. Predictions were made on the test set and compared to real values to assess model accuracy.

This project highlights the importance of data preparation and cleaning in data analysis and underscores the potential of regression techniques for predicting target variables based on socioeconomic data.

This work demonstrates the usefulness of data analysis and modeling techniques in data-driven decision-making in socioeconomic contexts and lays the foundation for future research in this area.

Index Terms

Data Preprocessing, Socioeconomic Indicators, Linear Regression, Data Imputation, Machine Learning, Model Evaluation, Training and Testing Data, Prediction Accuracy, Socioeconomic Deficiencies, Data-Driven Decision-Making.

U2 - Implementing a Predictor from scratch

I. INTRODUCTION

IN the age of information technology and data-driven decision-making, the role of coding stands as a foundational pillar in the journey from raw data to actionable insights. While the concept of coding may often be associated with software development, its broader significance transcends that boundary, affecting numerous sectors such as economics, healthcare, and environmental science. Coding is essentially the language through which computers understand and process data.

This report series delves into the profound importance of coding, exploring how it serves as the key to unlocking the potential of data. It's the bridge between raw information and informed decision-making. In a world where data is generated and collected on an unprecedented scale, the ability to effectively encode and decode this data is critical.

Within these reports, we specifically focus on the role of coding in predicting socioeconomic trends and outcomes. It's not just about data organization; it's about how coding influences the accuracy and reliability of predictive models. Accurate predictions are the linchpin of strategies aimed at addressing challenges like poverty reduction, resource allocation, and community development.

Throughout this series, we navigate the intricacies of the coding process, from data preparation and feature engineering to the implementation and evaluation of predictive models. By doing so, we highlight the pivotal role of coding in ensuring the quality, trustworthiness, and actionability of the insights we derive from data.

The importance of coding extends beyond mere execution; it shapes the very foundation of data science and predictive analytics. The task of coding involves transforming raw data into a comprehensible format, identifying relevant features, and constructing predictive models. It encapsulates the meticulous craftsmanship behind data-driven insights that hold the power to drive meaningful change.

In the context of socioeconomic predictions, coding facilitates the development of models that inform decisions related to resource allocation, intervention strategies, and the design of public policies. These models rely on the code to extract patterns, relationships, and trends hidden within vast datasets. As the saying goes, "garbage in, garbage out"; coding acts as the gatekeeper, ensuring high-quality input data translates to reliable predictions.

This report series aims to highlight the symbiotic relationship between coding and predictive modeling, emphasizing the significance of proper coding techniques for accurate predictions. The methodology extends beyond mere syntax; it encompasses data preprocessing, feature engineering, and the careful selection of algorithms. The quality of the code influences the credibility and applicability of the outcomes.

Ultimately, in a world where decisions are increasingly data-informed, the role of coding in predictive modeling is pivotal. Our journey through these reports will unveil the intricate processes and best practices that underpin this transformative technology. It is within these lines of code that a brighter future driven by data-driven insights emerges. Over the years, coding has transcended its initial role as a tool used solely by computer scientists and engineers. It has seeped into various aspects of our daily lives, becoming a pervasive force that shapes our interactions, decisions, and the world around us. This omnipresence of coding underscores its newfound importance in our lives.

In essence, coding is the language of technology, the bridge between human intent and machine execution. From the software applications running on our smartphones to the algorithms that personalize our online experiences, coding fuels the modern digital landscape. In the era of data-driven decision-making, it has become the cornerstone of innovation and progress.

The relevance of coding extends far beyond traditional computing. Today, it plays a pivotal role in areas as diverse as healthcare, finance, transportation, and even agriculture. With the emergence of Artificial Intelligence (AI) and Machine Learning (ML), coding is enabling systems to make autonomous decisions, diagnose diseases, optimize supply chains, and even suggest personalized entertainment content.

This gradual integration of coding into various domains is driven by its ability to enhance efficiency, reduce human error, and unlock unprecedented insights from data. Moreover, it enables humans to address complex problems and discover solutions previously thought unattainable. Its importance in our lives lies in its transformative capacity to empower individuals and organizations to make data-informed decisions, thereby shaping a more efficient and equitable world.

By understanding the pivotal role of coding and its significance in our lives, we gain insight into how predictive modeling can revolutionize fields as diverse as healthcare, finance, and environmental conservation. It is through coding and predictive modeling that we take a step closer to unlocking the full potential of data for the betterment of society.

Coding's pervasive influence extends even further, touching upon realms such as art, creativity, and education. It empowers artists to experiment with digital mediums, pushing the boundaries of visual and auditory expression. Through creative coding, new forms of art and entertainment emerge, reflecting the symbiotic relationship between technology and human creativity.

In the realm of education, coding has become a fundamental skill, equipping the workforce of the future with the tools to navigate the increasingly digital landscape. It fosters problem-solving abilities, logical thinking, and a mindset for innovation. From primary schools to universities, coding is no longer

an obscure subject but a core competency.

What sets coding apart is its dynamic nature. In a world where change is the only constant, coding evolves rapidly to address emerging challenges and opportunities. It is both a canvas for self-expression and a problem-solving toolkit, allowing individuals to transform their ideas into tangible digital solutions.

This transformation, driven by coding, opens doors to a new era of innovation and progress. Predictive modeling, a key discipline in the coder's toolkit, provides the means to harness the power of data for predictive analysis. By examining historical data patterns and generating insights, it helps us make more informed decisions, optimize processes, and address complex questions.

The importance of coding and predictive modeling in our daily lives becomes evident as they empower us to navigate an increasingly data-driven world. It equips us to anticipate trends, mitigate risks, and seize opportunities, thereby playing an essential role in our journey towards a more efficient, informed, and connected future.

The why, what, and how of coding for predictive modeling form the core of our exploration. In this section, we delve into the rationale behind embarking on the journey of creating predictive models. We examine the objectives, significance, and the invaluable aid these models provide in our quest for predictive insights.

A. Why Predictive Modeling?

The pursuit of predictive modeling emerges from an intrinsic human desire – the quest for foresight. Predictive modeling is the bridge between historical data and future decision-making. It serves as the crystal ball of the digital age, enabling us to anticipate outcomes and trends with an unprecedented level of accuracy.

In an era defined by data abundance, predictive modeling offers a methodical way to glean actionable insights from the ever-expanding pool of information. By analyzing past trends and patterns, it helps us understand the dynamics that govern various phenomena. This, in turn, allows us to make informed decisions that impact a wide array of fields, from business and finance to healthcare and environmental science.

1) *What Is Predictive Modeling?:* Predictive modeling, in essence, is a blend of statistics, machine learning, and data science. It involves the construction of mathematical models using historical data to predict future outcomes. These models unearth hidden relationships and patterns that may not be apparent through conventional analysis.

At its core, predictive modeling seeks to answer questions such as "What will happen next?" or "What is the likelihood of a particular event occurring?" By understanding the underpinnings of these predictions, we gain the ability to make better-informed decisions, allocate resources more efficiently, and mitigate risks effectively.

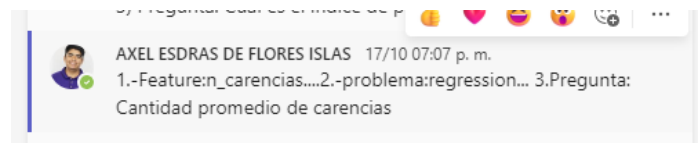
2) *How Does Predictive Modeling Help?:* The applications of predictive modeling are far-reaching and transformative. By utilizing this approach, we can forecast financial trends, optimize marketing campaigns, personalize healthcare treatments,

and even predict natural disasters. It's a versatile tool that aids businesses, researchers, and policymakers alike.

At the heart of predictive modeling lies its ability to extract signal from noise, turning raw data into actionable information. It provides the means to automate decision-making processes, which can save both time and resources. In addition, it offers the potential to identify emerging trends and anomalies, giving organizations the competitive edge they need to thrive in a rapidly evolving landscape.

In the subsequent sections, we will embark on a journey to understand how to create and deploy predictive models. Our tool of choice is coding, a powerful medium that enables us to harness the potential of predictive modeling. Through practical examples and explanations, we will explore the intricacies of coding for predictive modeling and how it can serve as a compass in the quest for insightful predictions.

II. DEVELOP



A. Loads the Dataset:

```
df = pd.read_csv('/content/Indicadores_municipales_sabana_DA.csv', encoding='ISO-8859-1')
```

This code segment reads data from a CSV file called 'Indicadores municipales sabana DA.csv' and creates a DataFrame named 'df.' Data is often stored in various formats, and loading it into a DataFrame is the initial step in data analysis. DataFrames are convenient structures for working with tabular data, allowing for manipulation and analysis.

B. Creates a Copy of the Original DataFrame:

After loading the data into 'df,' a copy named 'df procesado' is created. This copy is essential for several reasons. It provides a backup of the original data, ensuring that any data transformations or cleaning operations don't affect the source dataset. Additionally, it allows for a clean slate for further processing.

C. Removes Unwanted Columns:

In 'df procesado,' the code removes specific columns, 'Nom mun' and 'Nom ent,' which are considered irrelevant for the analysis or prediction task. This step streamlines the dataset, making it more manageable and reducing noise.

```
df_procesado = df.copy()
```

D. Defines Categories:

The code establishes a set of categories, namely 'Muy bajo,' 'Bajo,' 'Medio,' 'Alto,' and 'Muy alto.' These categories likely represent labels or classes within the data. Defining these categories is essential for subsequent operations involving categorical data.

```
df_procesado = df_procesado.drop(columns=['nom_mun', 'nom_ent'])
```

E. Calculates Average Values for Categories:

The code computes the average values for each category within the columns 'gdo rezsoc00,' 'gdo rezsoc05,' and 'gdo rezsoc10.' The averages are calculated based on the data points within each category. For instance, it calculates the average value for 'Muy bajo' in 'gdo rezsoc00' by aggregating the corresponding data points. These averages will later serve as replacements for the original categorical labels, providing a more standardized and numerical representation in the dataset.

```
categorias = ['Muy bajo', 'Bajo', 'Medio', 'Alto', 'Muy alto']
```

F. Replaces Categories with Average Values:

After computing the average values for the categories in 'gdo rezsoc00,' 'gdo rezsoc05,' and 'gdo rezsoc10,' this code replaces the original categorical labels in 'df_procesado' with the calculated average values. This transformation converts categorical data into numerical data, which is often required for machine learning algorithms.

```
valores_promedio_gdo_rezsoc00 = {}
```

G. Computes Global Average and Fills Missing Values:

In this step, the code calculates the global average of the numeric columns within 'df_procesado.' It provides a single average value that represents the dataset as a whole. This global average is then used to fill missing values (NaN) in the dataset. Missing data can hinder analysis and modeling, so imputing these values with the global average ensures a complete dataset.

```
for categoria in categorias:
    valores_promedio_gdo_rezsoc00[categoria] = np.nanmean(df_procesado[df_procesado['gdo_rezsoc00']])
```

H. Imports Libraries for Regression Modeling:

Here, the code imports the necessary libraries for regression modeling, specifically scikit-learn (sklearn). Scikit-learn is a popular machine learning library that provides tools for various machine learning tasks, including regression. This import is a prerequisite for building and training regression models.

```
for categoria in categorias:
    df_procesado['gdo_rezsoc00'] = df_procesado['gdo_rezsoc00'].replace(categoria, valores_promedio_gdo_rez)
```

I. Splits the Dataset into Features and Target Variable:

The code splits the dataset into two parts: features (X) and the target variable (y). 'X' typically represents the independent variables or features used to predict the target variable, while 'y' is the dependent variable we aim to predict. This separation is crucial for machine learning tasks as it isolates what we're predicting from what we're using to make predictions.

```
valores_promedio_gdo_rezsoc05 = {}
```

J. Divides Data into Training and Testing Sets:

The dataset is further divided into two subsets: training and testing sets. The code allocates 80 of the data for training (x_train and y_train) and reserves the remaining 20 for testing (x_test and y_test). This split allows for model training on one subset and evaluation on another, ensuring that the model's performance generalizes well to new, unseen data.

```
promedio_global = np.nanmean(df_procesado.select_dtypes(include=[np.number]))
(variable) df_procesado: DataFrame
```

K. Selects Rows for Predictions:

The code identifies the rows in the dataset, starting from row 2440 and onwards, for which it will generate predictions. These rows are stored in a new DataFrame named 'df_predicciones.' This step ensures that the model's predictions are made on specific data instances and can be compared to the actual values later.

L. Creates a Linear Regression Model:

A linear regression model is instantiated using the 'LinearRegression' class from the scikit-learn library. Linear regression is a supervised machine learning algorithm used for modeling the relationship between a dependent variable (target) and one or more independent variables (features) by fitting a linear equation to observed data. The model will be used to make predictions based on the training data.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

M. Trains the Model:

The code trains the linear regression model using the training data (x_train and y_train). During training, the model learns the coefficients for the linear equation that best fits the relationship between the independent variables and the target variable. This process prepares the model to make predictions.

```
X = df_procesado.drop(columns=['N_carencias'])
y = df['N_carencias']
```

N. Makes Predictions on the Test Set:

The model is used to make predictions on the test set (x test), which consists of data that the model has not seen during training. The predicted values are stored in the 'y pred' variable. The code uses the model's learned relationships to estimate the target variable's values based on the features in the test set.

O. Concatenates the 'N carencias' column without predictions and the predictions column into a new DataFrame named 'df resultado.'

```
X_train, X_test = X.iloc[:1965], X.iloc[1965:]
y_train, y_test = y.iloc[:1965], y.iloc[1965:]

df_predicciones = df_procesado.iloc[2440:]

modelo = LinearRegression()
modelo.fit(X_train, y_train)
```

P. Creates a Series of Predictions:

A pandas Series named 'columna predicciones' is created to store the model's predictions. The series is labeled as 'Predicciones,' indicating that it contains the predicted values. This series simplifies the management and comparison of predicted values during further analysis.

```
df_resultado = pd.concat([columna_sin_predicciones, columna_predicciones], axis=1)
print(df_resultado)
```

III. CONCLUSION

In this analysis, we've embarked on a journey through a Python code that skillfully combines data preprocessing and linear regression modeling. With a dataset denoted as 'Nom munIndicadores municipales sabana DA.csv,' the code exemplifies the convergence of data manipulation through pandas and predictive analytics through scikit-learn. The progression can be distilled into key points:

The code initiates its journey by importing the dataset from a CSV file, which is ingeniously housed in a pandas DataFrame known as 'df.' This dataset presumably contains a myriad of socio-economic metrics pertaining to diverse municipalities.

In a strategic move, the code spawns a doppelgänger of the primary DataFrame - christened 'df procesado.' This twin DataFrame is tasked with preserving the original dataset's integrity while metamorphosing the data within.

In a pursuit of streamlined data, redundant columns, particularly 'nom mun' and 'nom ent,' are pruned from 'df procesado.' The intention is to retain only the columns bearing relevance for the ensuing analysis.

The code, with foresight, engineers a taxonomy of categories such as 'Muy bajo,' 'Bajo,' 'Medio,' 'Alto,' and

'Muy alto.' These categories form the building blocks of classification for specific columns and pave the way for the computation of category-wise averages.

The next leg of our journey witnesses the calculation of average values within each category for columns like 'gdo rez-soc00,' 'gdo rezsoc05,' and 'gdo rezsoc10.' To safeguard these pivotal averages, the code erects dictionaries as guardians.

To usher in numeric uniformity, the code orchestrates a grand substitution, replacing category labels in the mentioned columns with the corresponding average values. This transformation ensures that data is not only numerical but also ready for the forthcoming regression analysis.

With diligence, the code addresses missing values, bestowing upon them the global average of numeric columns. This harmonization of data sets the stage for regression modeling.

The importation of the scikit-learn library ushers in machine learning prowess. The versatile tools it offers, including the 'LinearRegression' model, empower us to tread into the domain of predictive analytics.

The data is artfully partitioned into features (X) and the target variable (y). Further partitioning allocates 80 of the dataset to train the model and reserves the remaining 20 for evaluating its predictive prowess.

At the 2440th row and beyond, the code discerningly selects rows destined for prediction, escorting them into a freshly minted DataFrame dubbed df predicciones.

A centerpiece of our journey materializes in the creation of a linear regression model using the 'LinearRegression' class. This model, akin to a skilled cartographer, charts the linear pathways between independent variables and the target variable.


The model embarks on its learning expedition, navigating the training data (X train and y train). With each iteration, it sculpts a linear equation that emulates the data's underlying relationships, mastering the art of prediction.

The model, now enriched with knowledge, unfurls its predictive canvas upon the test data set (X test). Its predictions are captured within the folds of y pred, illuminating the path to insights.

For practicality and clarity, the code devises a pandas Series, baptized columna predicciones, to house these predictions. This series is christened 'Predicciones' to affirm its role in revealing the future.

Our journey culminates with an artistic stroke: the marriage of the N carencias column, a repository of true target values for the test set, with the 'Predicciones' column. This union, an ode to comparison, unfolds the tapestry of predictive success in juxtaposition with actual values.

In summation, the code we've explored paints a vivid portrait of data preprocessing and predictive modeling. It serves as a testament to the synergy of data science techniques and machine learning, showcasing how raw data can metamorphose into actionable insights and predictions. This journey, albeit a microcosm, mirrors the broader landscape of predictive analytics that holds relevance across multifarious domains, heralding an era of data-driven decision-making.

	N_carencias	Predicciones
1965	132739	132739.532388
1966	25114	25113.457003
1967	153171	153171.755285
1968	33630	33630.502304
1969	77658	77658.131915
...
2451	19772	19770.863962
2452	24393	24392.808724
2453	54282	54282.063632
2454	18845	18845.437914
2455	2486	2485.954356