Integrated Project: Predicting Poverty in Colombia

Machine Learning

2025

"Wars of nations are fought to change maps.
But wars of poverty are fought to map change"

— M. Ali

UNLP

.1 Introduction

This group project is inspired by a recent competition hosted by the world bank: Pover-T Tests: Predicting Poverty. The idea is to predict poverty in Colombia. As the competition states, *"measuring poverty is hard, time consuming, and expensive. By building better models, we can run surveys with fewer, more targeted questions that rapidly and cheaply measure the effectiveness of new policies and interventions. The more accurate our models, the more accurately we can target interventions and iterate on policies, maximizing the impact and cost-effectiveness of these strategies."*

The main objective of the project is to construct a predictive model of household poverty in Colombia. In this country a household is classified as

$$Poor = I(Ingpcug < Pl) \tag{1}$$

where I is an indicator function that takes one if $Ingpcug$ is per-capita income of the spending unit and Pl is the poverty line used in this course. This is a course-long project and to reach this objective you'll deliver two parts:

1. **Part 1 (Regression)**: Predict income (`Ingpcug`) → Continuous regression problem
2. **Part 2 (Classification)**: Predict poverty directly (Poor = 0/1) → Binary classification problem

Each part builds upon the other. Part 1 is your foundation. The variables you create, the insights you gain, and the data preparation you do will directly support your Part2 work. Invest time in understanding your data deeply in Part 1 it will pay dividends in Part 2.

Teams for the project are fixed for the duration of the course and cannot exceed 3 members.

The weights and due dates are specified in the table below:

Deliverable	Weight	Due Date
Part 1	**25%**	Document (November 16)
Part 2	**75%**	Kaggle (December 2) Oral Presentation + Code (December 4)

.2 The Data

The data comes from DANE and the mission for the "Empalme de las Series de Empleo, Pobreza y Desigualdad - MESE". The data contains four sets divided into training and testing at the household and individual levels. You can use the variable `id` to merge households with individuals. You will note that some variables are missing in the testing data sets; this is designed to make things a bit more challenging. More information about the data is available at the competition website.[1]

[1] To join the competition you need a link that will be shared in class

Part I
Predicting Income

I.1 Overview

- **Weight**: 25% of final grade
- **Due date**: November 16
- **Deliverable**: Single document (maximum 8 pages)

I.2 Objective

Build a predictive model of **per capita income of the spending unit (Ingpcug)** using regression models, from simple to regularized approaches.

This preliminary analysis will help you:

- Familiarize yourself with the data before tackling the classification problem
- Build fundamental skills in regression models
- Create derived variables that you can reuse in Problem Set 2
- Understand the relationship between income and poverty

I.2.1 Specific Learning Objectives

- Merge household and individual-level datasets
- Build aggregated variables (feature engineering)
- Implement regression models with increasing complexity
- Perform hyperparameter selection through cross-validation
- Evaluate performance using appropriate metrics (RMSE, MAE, R^2)

I.3 Deliverable

I.3.1 Single Document (Maximum 8 pages)

This is an **exploratory document** that serves as foundation for part 2. While no code submission is required for Part 1, maintain your scripts as you'll build on this work in Part 2. Please follow this structure

I.3.1.1 Introduction

- Brief problem statement: Why predict income?
- Relationship between income and poverty measurement
- Data overview
- Preview of main findings

I.3.1.2 Data

Must include:

Data construction process

- How you merged household and individual datasets
- Aggregated variables created from individual-level data:
  - Demographic variables: composition by age, sex, dependency ratio
  - Human capital: average education, years of schooling
  - Labor market: employment rates, formality, sectors
  - Other relevant variables
- Data cleaning: treatment of missing values, outliers, inconsistencies
- Transformations applied (log, standardization, etc.)

Descriptive analysis

- Descriptive statistics table of key variables
- Distribution of target variable (Ingpcug)
- Important relationships (correlations, visualizations)

Add professional value. Justify each data decision based on the predictive goal.

I.3.1.3 Regression Models

Train your model using the following algorithms:

1. Simple/Multiple Linear Regression
2. Ridge Regression
3. Lasso Regression
4. Elastic Net

This section must include:

Training and Hyperparameter Selection

- Validation strategy (train/test split, k-fold CV, etc.)
- For Ridge/Lasso/Elastic Net:
  - Range of alpha/lambda values tested
  - Search strategy (grid search, random search)
  - Justification for chosen ranges
- Main evaluation metric and why you chose it
- Cross-validation summary table for all models

Best Model Analysis

- Which model performed best
- Final selected hyperparameters
- Basic diagnostics:
  - Residuals vs fitted values
  - Predicted vs actual values

Model Comparison

- Comparative performance table
- Discussion: Why do some models outperform others?
- Trade-offs between model complexity and performance

Variable Importance

- What variables are the most important predictors of income?
- Do these results make economic sense?

I.3.1.4 Conclusions

- Summary of main findings
- Lessons learned about data and models
- How this work prepares you for part 2

I.4 Evaluation Rubric

Total: 25% of final grade = 100 points

Component	**Points**	**Breakdown**
Data Preparation	30	Correct merging (5), Well-justified variables (12), Cleaning and data treatment (8), Descriptive analysis (5)
Models & Methods	50	5 models correctly implemented (20), Hyperparameter selection documented (15), Systematic comparison (10), Results interpretation (5)
Document Quality	20	Clarity and structure (10), Professional format (5), Narrative coherence (5)

Note: An excelent document is one that demonstrates critical thinking, not just mechanical model execution.

Part II
Predicting Poverty

II.1 Overview

- **Weight**: 75% of final grade
- **Due date**: Predictions are due on Kaggle on December 2, Oral Presentations and Code are due on December 4.
- **Components**: (1) Kaggle submissions, (2) Oral presentation, (3) Reproducible code

II.2 Objective

Build a predictive model of **household-level poverty** using classification algorithms. Now that you understand the data and have built variables predictive of income, you'll directly address the binary prediction:

$$Poor = I(Inc < Pl) \tag{2}$$

Public policy context: Policymakers need to measure poverty quickly and cheaply. Your goal is to build a model that uses the minimum number of variables while maintaining high accuracy.

II.3 Deliverables

The part 2 deliverables consists of **three components**:

1. **Kaggle submissions** with your team's predictions
2. **Oral presentation** of results
3. **Reproducible code** of the winning model

II.3.1 Component 1: Kaggle Submissions

- **Minimum**: 12 submissions using at least 5 different algorithms
- **Required algorithms** (choose at least 5):
  - Linear Regression
  - Logistic Regression (Logit)
  - Elastic Net
  - Classification and Regression Trees (CART)
  - Random Forest
  - Boosting (AdaBoost, XGBoost, LightGBM)
  - Neural Networks

Note: You can and should experiment to maximize your score. Winning model will receive a 10 percent bonus on the project grade.

II.3.2 Component 2: Oral Presentation

- **Format**: Last day of class
- **Structure**: Single slide deck

II.3.2.1 Recommended Content

Mandatory First Slide: Best Model Overview

- Specification of the best model.
- Kaggle score (public and private leaderboard)

Data Story

- How you built and cleaned the data
- Key variables created and why
- Most important descriptive statistics
- Data decisions that impacted the best model
- Challenges encountered and solutions

Model Exploration

- Comparison of alternative models
- Comparative performance table
- Why some models underperformed
- Issues: specification, overfitting, underfitting
- How learnings from these models informed your best model

Best Model Deep Dive

- Detailed training process
- Hyperparameter selection (grid search, ranges, CV results)
- Examine class balance in your training data and document your approach (whether you address it or explain why not)
- Feature importance

Conclusion

- Key insights
- Policy recommendations

II.3.2.2 General Rules for Presentations

Do:

✓ Self-contained slides (title, labeled axes, legends)

✓ Focus on insights, not minute technical details

✓ Use high-quality visualizations

✓ Tell a coherent story

✓ Respect time limit

Don't:

× Unformatted screenshots from R/Python

× Excessive text or code on slides

× Illegible tables

× Generic slides that don't add value

Golden rule: Every element of your presentation should serve the story of your best model.

II.3.3 Component 3: Reproducible Code

You must send a compressed file that contains code to clean and merge the data and returns the best model. **Don't include the data**, I will download it and put it in the folder indicated by the README. I should be able to get the same score as you did on Kaggle.

File :

- **The code must be**:
    - Fully reproducible scripts with a master file that clean and merge the data and returns the best model.
    - README explaining how to reproduce the code, R/Python version, package versions, estimated run time.
    - Readable and include comments. In coding, like in writing, good coding style is critical. I encourage you to follow the tidyverse style guide

II.4 Evaluation Rubric

Total: 75% of final grade = 100 points.All team members should be prepared to answer questions

Component	Points	Breakdown
Data Preparation	10	Adequate description (4), Sample construction (2), Descriptive analysis with interpretation (4)
Models & Methods	35	12 submissions/5+ algorithms (10), Best model training process (10), Hyperparameter tuning (8), Comparative analysis (7)
Feature Importance	10	Feature importance analysis (5), Empirical evidence (3), Economic/policy interpretation (2)
Kaggle Performance	15	At least 12 submissions (5), Competitive score on leaderboard (10)
Oral Presentation	15	Clarity and structure (7), Time management (3), Visual quality (5)
Code	15	Full reproducibility (10), Excellent README (5)