Problem Set 2: Predicting Poverty

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

1 Introduction

This problem set was 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 objective is to predict poverty at the household level. Data, however, are provided at the household and individual levels. You can use individual-level information to build additional variables to improve your prediction.

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.

An essential dimension for policymakers is they can *rapidly and cheaply* measure poverty. When building your model, aim for a model that uses the minimum number of variables.

There are three sets expected outputs:

1. A `.pdf` document.
2. A set of slides for presentation.
3. Submissions with your team's predictions in Kaggle. To join the competition use the following link.

General Instructions

The main objective is to construct a predictive model of household poverty. Note that a household is classified as

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

where I is an indicator function that takes one if the family income is below a certain poverty line.

This suggests two ways to go about predicting poverty. First, approach it as a classification problem: predict zeros (no poor), and ones (poor). Second, as an income prediction problem. With the predicted income, you can use the poverty line and get the classification. You will explore only the classification route in this problem set.

The document must contain the following sections:

1. **Introduction**. The introduction briefly states the problem and if there are any antecedents. It briefly describes the data and its suitability to address the problem set question. It contains a preview of the results and main takeaways.

2. **Data.**[1]. When writing this section up, you must:

   2.1 Describe the adequacy of the data to solve the predictive question, the sample construction process, including how the data was cleaned, combined, and how new variables were created.

   2.2 Include a descriptive analysis of the data. At a minimum, you should include a descriptive statistics table with its interpretation. However, I expect a deep analysis that helps the reader understand the data, its variation, and the justification for your data choices. Use your professional knowledge to add value to this section. Do not present it as a "dry" list of ingredients.

3. **Models and Results**

   This section presents the specifications and models used for the predictive tasks. Each team must submit at least **eight (8) predictions** using models trained with at least **five (5) distinct classification algorithms** chosen from the following:

   Linear Regression, Logistic Regression (Logit), Elastic Net, CARTs, Random Forest, Boosting, and Naive Bayes.

   The analysis should include:

[1]This section is located here so the reader can understand your work, but probably it should be the last section you write. Why? Because you are going to make data choices in the estimated models. And all variables included in these models should be described here.

3.1 *Model Selection and Training:* A detailed discussion of the model that achieved the team's highest public score in Kaggle. This discussion should include the training process, selection of hyperparameters, and any additional methodological considerations. Additionally, the analysis must *justify whether a class imbalance strategy was used or not*, explaining any sub-sampling or weighting strategies implemented to address potential class imbalances in the data.

3.2 *Hyperparameter Tuning:* A dedicated section describing how hyperparameters were selected for each chosen model. This must include the range of values tested, the search strategy (e.g., grid search or other methods), the rationale for choosing those ranges, and a summary of cross-validation results or performance metrics that guided the final choices.

3.3 *Comparative Analysis:* A comparison between the **best-performing model** and at least **five of the other five best-performing team submissions in Kaggle**. This comparison should highlight differences in performance and provide an explanation of why certain models outperformed others — whether due to issues related to specification, training parameters, feature selection, or any other relevant aspect.

3.4 *Feature Importance:* A discussion of the relative importance of the variables in the **best-performing model**, explaining why certain features played a key role in improving predictions. This explanation should be supported by empirical evidence, such as feature importance scores or any other method used to assess variable contribution.

This section should be written as a structured discussion, providing both technical details and an analytical perspective on the modeling choices and their implications for predictive performance.

4. **Conclusion** . In this section, you state the main takeaways of your work.

2 Additional Guidelines

- Predictions have to be submitted on Kaggle. Check the competition website for more information.
- Turn a `.pdf` document in Bloque Neón. The document should not be longer than 12 (twelve) pages and include, at most, 10 (ten) exhibits (tables and/or figures). Bibliography and exhibits don't count towards the page limit. You are welcome to add an appendix, but the main document must be self-contained. Specifically, a reader should be able to follow the analysis in the paper and be convinced it is correct and coherent from the main text alone, without consulting the appendix.
- The document must include a link to your GitHub Repository.

  - The repository must follow the template.
  - The README should help the reader navigate your repository. A good README helps your project stand out from other projects and is the first file a person sees when they come across your repository. Therefore, this file should be detailed enough to focus on your project and how it does it, but not so long that it loses the reader's attention. For example, Project Awesome has a curated list of interesting READMEs.
  - Include brief instructions to fully replicate the work.
  - The main repository branch should show at least five (5) substantial contributions from each team member.
  - The code has to be:
    * Fully reproducible.
    * Readable and include comments. In coding, like in writing, a good coding style is critical. I encourage you to follow the tidyverse style guide.

- Tables, figures, and writing must be as neat as possible. Label all the variables included. If you have something in your figures or tables, I expect they are addressed in the text. Tables must follow the AER format.

- **Slides for in-class presentation:** In addition to the `.pdf` document, each team must prepare **three** sets of slides to present in class. These must be uploaded to the activity 'Slides: PS2' in Brightspace.

  - **File name format:** `nombre_equipo_##` (use leading zero for teams numbered below 10). Example for team 1:
    * `data_equipo_01` (Data)
    * `othermodels_equipo_01` (Other Models)
    * `best_equipo_01` (Best Model)
  - **Respect these file names exactly**.
  - **Purpose of each slide deck:**
    * `data_equipo_##`: Show how the data was built and cleaned, highlight key variables and descriptive statistics, and explain how these choices supported the best model.
    * `othermodels_equipo_##`: Compare alternative models, summarize performance, and explain why some underperformed. This should position the best model in contrast to the others.
    * `best_equipo_##`: Provide a deep dive into the best model (training, tuning, feature importance, diagnostics) and conclude with takeaways. This is the climax of the story.
  - Maximum **15 minutes** per presentation.

- **Mandatory First Slide:** Regardless of which section you present, you must start with a *Best Model Overview* slide, including:
  - Specification of the best model.
  - Kaggle score (public leaderboard).
- Focus on highlighting the most important aspects of your work (key results, interpretations, conclusions).
- Tables and figures must be self-contained and properly formatted, with a title, labeled axes, and a legend. Tables must not be screenshots from R, Python, or any other software. They do not have to be the same as those in the `.pdf` document, and it is encouraged to reformat them for presentation purposes.
- Avoid excessive text or code in the slides.
- **Golden Rule:** Every part of your presentation should serve the story of the best model.