Data Analysis

The file `train_hh_features.csv` includes 104,234 households and multiple features describing the form, living conditions, and consumption habits of each household. Each record corresponds to a unique household, identified by the `hhid`.

The main categories into which the features are divided are demographic characteristics, such as the gender of the head of the household (`male`), the total household size (`hsize`), and the number of children in different age groups. These variables offer information on the household structure and affect consumption needs and the degree of financial vulnerability.

Furthermore, another category includes economic variables, with the most important being consumption per person in purchasing power parity (`utl_exp_ppp17`), which functions as a key indicator of well-being. This specific variable constitutes the core of poverty estimation, as it directly reflects the households' actual standard of living. Complementarily, statistical weights (`weight`) and sampling variables play a significant role; while they do not directly affect prediction at the individual level, they are necessary for the correct interpretation and generalization of results at the population level.

Finally, the largest part of the dataset consists of binary food consumption variables, which indicate whether the household consumed specific goods, and are used as indicators of economic status and quality of life. Households with a higher level of consumption tend to present a greater variety of dietary goods.

Application of Machine Learning and Deep Learning Algorithms

Three different Machine Learning algorithms were selected, as well as one Deep Learning algorithm. Data preprocessing is similar across all 4 models. Initially, I performed data cleaning, handled missing values, and applied appropriate encoding for categorical features so that the data would be compatible with the algorithms applied to each model, and subsequently, the data was split into training and evaluation sets.

The first algorithm applied is **Random Forest**. This specific model was chosen because it operates by creating multiple different decision trees and combining their predictions using a Bagging method. The goal was to reduce the risk of overfitting and provide more stable and reliable results compared to a single tree; it also presents good performance on data with a large number of features. In the corresponding code, basic hyperparameters are tuned, such as the number of trees and maximum depth, aiming to avoid overfitting and improve the model's generalization (training was done with as many iterations as Colab allowed). Additionally, a key positive of Random Forest is that it shows feature importance indicators and is easier to interpret.

The second algorithm examined is **Gradient Boosting**, a machine learning algorithm belonging to the ensemble methods category, meaning it combines many simple models to create a better one. Specifically, it is based on decision trees, which are trained sequentially; each new tree attempts to correct the errors made by the previous ones. In this way, the model improves gradually and can learn complex, non-linear relationships in the data.

Gradient Boosting was chosen for this problem because the data is structured, contains many categorical and numerical variables, and the relationships between them are not simple. Such algorithms are particularly efficient in predicting socio-economic indicators, such as household consumption and poverty rates. In the code, the base model is trained on the training data using parameters such as learning rate, tree depth (`depth`), and the number of iterations, which determine how fast and how complexly the model learns. Subsequently, cross-validation per survey is applied to check if the model generalizes well to different surveys and not just the data seen during training. This specific approach proved particularly effective for the problem, achieving the highest score compared to the other models in the competition.

The third algorithm is **Support Vector Regression (SVR)**, chosen because it can predict numerical values when relationships between data are not linear. The data concerning households contains diverse information as well as noise, and SVR is good for such cases because it generalizes well and is not easily affected by outliers. Furthermore, through its parameters, it allows control over model accuracy and overfitting. A basic limitation is that it requires increased computational cost (in time) on large datasets; therefore, model training and parameter tuning were performed on a small random subset of the training set, while the final model was trained on the full dataset.

Finally, a **Deep Learning** algorithm was applied. For model training, a Multi-Layer Perceptron (MLP) neural network was used, where input data passes sequentially through hidden layers with non-linear activation functions, allowing the network to learn complex relationships between features and consumption. Training was performed iteratively in epochs (as many as Colab allowed) using the backpropagation algorithm and the AdamW optimizer. Initially, the network was trained only for consumption prediction, while subsequently, fine-tuning was applied where the loss function was modified to also take into account poverty rates per survey. The use of the deep learning algorithm presented some challenges; training required more time compared to other algorithms, especially when many hidden layers were used, and it is also not easy to explain in a simple way how each feature affects the final prediction.

Explanation and Analysis of Results

The results show that each model has its own advantages and difficulties.

The **SVR** model had moderate performance, with an RMSE of approximately 5.48 and R2 of approximately 0.70. This means it can predict the general trend of the data but struggles when relationships between features are more complex. It is also quite affected by the choice of its parameters.

Random Forest presented better and more stable performance, with a weighted MAPE of approximately 0.15, showing that it makes more reliable predictions. It also works well when there are many interactions between features but can be affected by noise in the data.

Gradient Boosting also gave some of the best results, as it gradually improves its errors and gives greater weight to the most important features, though it again requires care to avoid overfitting.

The **Deep Learning** model had low error during training (loss approximately 0.31), but its performance depends heavily on the quantity of data and does not perform as well on new data. Generally, all models place greater emphasis on numerical and percentage features, which play a significant role in prediction quality. The performance of each model could be improved with more data, better preprocessing, better hyperparameters for the algorithms, or even more training iterations.

Competition Submission

The best submission at the competition level was that of the Gradient Boosting algorithm, and below is the screenshot from the competition ranking.