Università degli studi di
Milano-Bicocca

Advanced Machine Learning

Final Project

# Household Poverty Classifier

*Authors:*
Mattia Artifoni - 807466 - m.artifoni@campus.unimib.it
Federico Bottoni - 806944 - f.bottoni1@campus.unimib.it
Riccardo Capra - 808227 - r.capra2@campus.unimib.it

24 January 2020

**Abstract**

Poverty is a complex issue in every country, mostly in Latin America where, according to data, many people live in dangerous houses, have no toilets and have no level of education. From a dataset provided by the Inter-American Development Bank a predictive model has been trained to classify people automatically in four wealth-levels. The classifier is able to predict its target with 97% of accuracy and, after a fairness study, data prove that the model doesn't discriminate on the people's gender.

# 1 Introduction

Many social programs have a hard time making sure the right people are given enough aid. In Latin America, one popular method uses an algorithm to verify income qualification. It's called the Proxy Means Test (or PMT). With PMT, agencies use a model that considers family's observable household attributes, like the material of their walls and ceiling or the assets found in the home, to classify them and predict their level of need. [1]
Costa Rica and many other countries face this same problem of inaccurately assessing social need, so a predictive model is necessary.

# 2 Datasets

The data are presented in an only table where each row represents a person living in Costa Rica. The 142 columns are features that describe the 9557 entities of this dataset, their family, their job and their economic situation. In the features-set is possible to find general informations about the person such as the gender, the monthly rent payment, the number of members of the hosehold and the region where the person lives. More specific infomrations like the type of rubbish disposal, the conditions of the walls and the roof and the usage of the main source of energy can also be found.
The 143rd column represents the target which introduces a number between 1 and 4 that refers to the welfare level of the person described in the row. The levels' meaning and the frequencies are described in the following table.

| Code | Description | Abs. frq. | Rel. freq. |
|---|---|---|---|
| 1 | extreme poverty | 755 | 7.89% |
| 2 | moderate poverty | 1597 | 16.71% |
| 3 | vulnerable households | 1209 | 12.65% |
| 4 | non-vulnerable households | 5996 | 62.74% |

The dataset is unbalanced on the four classes because more then half of the records describe the non-vulnerable households who are considered the most wealthy category between these.

## 2.1 Data manipulation

The first step needed to obtain a clean training-proof dataset is the deletion of the redundant or useless columns for the presented task. Here's a list of these with the respective description:

- 'Id': Univocal identifier for each person
- 'r4h3': Total males in the household
- 'r4m3': Total females in the household
- 'r4t1': Persons younger than 12 years of age
- 'r4t2': Persons 12 years of age and older
- 'r4t3': Total persons in the household
- 'rez_esc': Years behind in school
- 'elimbasu5': If rubbish disposal mainly by throwing in river, creek or sea
- 'idhogar': Household level identifier
- 'hogar_total': Number of total individuals in the household
- 'mobilephone': If in posses of a mobile phone

Second, some columns that presented empty cells or incomplete/difficult to read values were treated respectively:

- 'v2a1': Monthly rent payment (the empty cells were filled with the absolute mean calculated from each other cell value fo the same column)
- 'v18q1': Number of tablets household owns (the empty cells were filled with 0 i.e. the mode of the column)
- cells containing yes/no values (cenversion to 1/0 numeric values was applied)

Next the rows of 'meaneduc' (average years of education for adults (18+)) and 'SQBmeaned' (square of the mean years of education of adults ($\geq$18) in the household) that presented empty cells have been removed. Finally 0 has been prepended to the float values that were expressed in dotted notation (e.g. .09) and periodic numbers has been rounded to 2 decimal values.

# 3 The Methodological Approach

Since the task aims to the classification of the welfare level of the people living in Costa Rica, the approach consists in training a classifier which, given the person's features, is able to collocate him (or her) in a existing welfare level. Initially the data have been preprocessed in order to be handled by the machine learning library, removing the useless features and the incomplete columns and rows. After they have been used as input of the deep neural network to train it and evaluate the metrics.
The choice of the deep neural network technique, despite being a black-box, is connected to the elasticity of the model that can be computationally expansive during the training and the evaluation of metrics step, but quick and well-performing in the inferential (or predictive) step.

## 3.1 Working hypothesis

The dataset describes a set of people with many interesting features that help the training of the model under different aspects. An assumed hypothesis is that data of the members of the same household (distinct rows in the dataset) affect the model's predictions in the same way as the members of different families, so the feature that describes it was ignored.

## 3.2 Software architecture

The project was developed as a simple Python application, exploiting some preprocessing and evaluating methods of the machine learning library Scikit-learn while the classifier was built in Keras.
The application has multiple entry points in order to solve different tasks:

- "preprocess_raw.py" parses the dataset exporting two new tables: one with the not relevant features removed and one like the first one but also with the binary fields which refer to the same feature grouped in order to test a logical grouping hypothesis
- "nnscore.py" given the input dataset path, trains the final deep neural network and plot the performances
- "sample_hps.py" given the input dataset path, runs the sampling of the specified hyperparameters in the */src/auto_ml.py* file. The sampling is made through the surrogate models Gaussian Process and Random Forest from the pyGPGO library [2]
- "fairness.py" given the input dataset path, evaluate the faireness of the trained model cheking the bias of some selected features

## 3.3 Data preprocessing

Data labels have been initially converted into categorical (with keras' method *to_categorical*) and the features standardized (with sklearn's module *StandardScaler*). Next a data upsampling step have been performed, since the 4 different target classes presented substatially different cardinalities. The method used is the imblearn *SMOTE*. The upsampling was performed such that the cardinality of all the classes but the most numerous became the 80% of this last one.
Finally the dataset has been shuffled and splitted in training-set 90% and test-set 10% and passed to the neural network.

## 3.4 Neural Network

The network is built as following:

- Input layer with 142 neurons and "relu" as activation function

- Dense layer with 254 neurons and "relu" as activation function
- Dropout layer to turn off 20% of neurons
- Dense layer with 160 neurons and "relu" as activation function
- Dropout layer to turn off 20% of neurons
- Dense layer with 216 neurons and "relu" as activation function
- Dense output layer with 4 neurons and "softmax" as activation function

The main structure is the result of a lot of tests changing the number of layers, the number of neurons and activation functions. The loss function is a "categorical_crossentropy" chosen because it provided the best performance as a categorical loss function. The optimizer passed to the compilation method is "Adam" which parameters have been extracted from the surrogate models. The network was fitted with data using batch_size = 32.

## 3.5 Auto-ML

An auto-ml module has been developed in order to sample some specific hyperparameters. The techniques used are the Gaussian Process and the Random Forest as surrogate models from the pyGPGO library, evaluating the described neural network through the 10-fold cross validation. At the end of the execution of both the two surrogates, the best performance is printed to the user's console. The selected hyperparameters chosen to be sampled are the learning rate, the two beta-values of the Adam optimizer and the three numbers of neurons that compose the hidden layers. Ranges were defined as following:

| | |
|---|---|
| learning_rate | [ 0.001, 0.01 ] |
| beta1 | [ 0.8, 0.999 ] |
| beta2 | [ 0.8, 0.999 ] |
| n1 | [ 8, 256 ] |
| n2 | [ 8, 256 ] |
| n3 | [ 8, 256 ] |

After 20 iterations for each surrogates, the best result is:

$$\text{HPs} = \begin{vmatrix} learning\_rate \\ beta1 \\ beta2 \\ n1 \\ n2 \\ n3 \end{vmatrix} = \begin{vmatrix} 0.0020909227100753 \\ 0.9199471584216276 \\ 0.9788631577850126 \\ 254 \\ 160 \\ 216 \end{vmatrix}$$

In this iteration is possible to find the first three parameters which describe the behavior of the optimizator and the last three that define the number of neurons for each hidden layer.
From this hypothesis the neural network has been created and trained, the tests performed and the best sampled performance returned.

# 4 Results and Evaluation

The evaluations of the network were calculated as precision, recall and f1-score on the test-set (10% of the dataset which is not used to train the dataset) and also with 1000 iterations of *leave-1-out-crossvalidation* as exhaustive metric where the network was trained using all the dataset but one record which alone makes up the test-set.

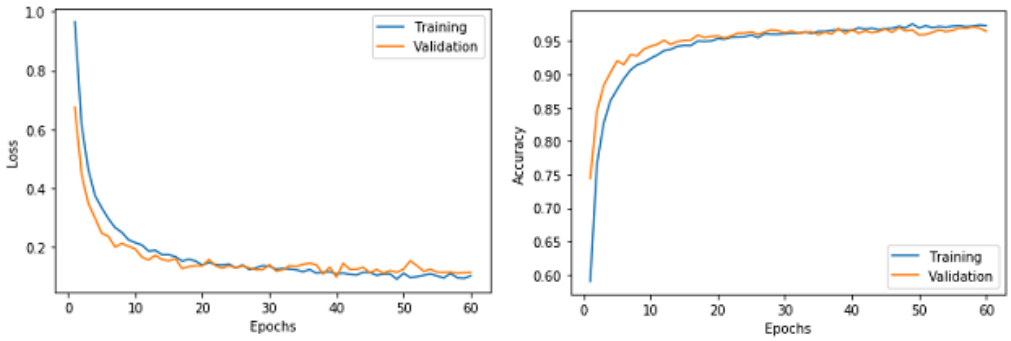


Figure 1: Loss and accuracy plots of the base model

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 0.99 | 0.99 | 493 |
| 1 | 0.96 | 0.96 | 0.96 | 489 |
| 2 | 0.98 | 0.93 | 0.95 | 489 |
| 3 | 0.95 | 0.98 | 0.96 | 566 |
| accuracy | | | 0.97 | 2037 |
| macro avg | 0.97 | 0.96 | 0.97 | 2037 |
| weighted avg | 0.97 | 0.97 | 0.97 | 2037 |
| leave-1-out-cv | | | 0.945 | 1000 |

## 4.1 Experiments

From this baseline some tests have been made to check the model's limits.

### 4.1.1 Oversampling step removed

In the first one the oversampling preprocess was removed and the network has been trained another time with the same hyperparameters in order to recognize the importance of balancing the dataset:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.77 | 0.84 | 0.80 | 67 |
| 1 | 0.81 | 0.76 | 0.78 | 158 |
| 2 | 0.81 | 0.93 | 0.86 | 107 |
| 3 | 0.96 | 0.94 | 0.95 | 623 |
| accuracy | | | 0.87 | 955 |
| macro avg | 0.84 | 0.87 | 0.85 | 955 |
| weighted avg | 0.90 | 0.90 | 0.90 | 955 |
| leave-1-out-cv | | | 0.864 | 1000 |

### 4.1.2 Grouping of data

Another performed test was the following. The columns that refer to a same feature of the person were grouped according to the logic, for example:

- instlevel1, =1 no level of education
- instlevel2, =1 incomplete primary
- instlevel3, =1 complete primary

- instlevel4, =1 incomplete academic secondary level
- instlevel5, =1 complete academic secondary level
- instlevel6, =1 incomplete technical secondary level
- instlevel7, =1 complete technical secondary level
- instlevel8, =1 undergraduate and higher education
- instlevel9, =1 postgraduate higher education

These columns were grouped in an only feature "instlevel" which values are integers between 0 and 8 according to the education level. When the rebuilding of the feature from the column's semantic was possible, these groups were created; in other cases the columns have been left categorical. The final number of features was 93 and the performances of the network were:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 0.98 | 0.98 | 486 |
| 1 | 0.95 | 0.94 | 0.95 | 469 |
| 2 | 0.98 | 0.95 | 0.97 | 472 |
| 3 | 0.94 | 0.97 | 0.95 | 610 |
| accuracy | | | 0.96 | 2037 |
| macro avg | 0.96 | 0.96 | 0.96 | 2037 |
| weighted avg | 0.96 | 0.96 | 0.96 | 2037 |
| leave-1-out-cv | | | 0.925 | 1000 |

# 5 Discussion

The unbalance of the dataset frequency probably depends directly on how people were requested to provide their important data. Maybe poor people were not interested in this or simply were unaware of the existence of this project because ignorant or just far from the community. Despite it's hard to represent people and household features describing their socio-economic context, the results have been higher then expected. One of the possible explanations lies behind on the dataset' structure, built by an official institution upon impartial and deterministic metrics.
Given that the quantity of data is not so huge, it can be supposed that the

model can learn almost all the correlations that allow to determine the wealth level of people. The model can be realistic if these correlations don't change in time and if it couldn't learn any other.
The first experiment was reported in order to show how much the oversampling preprocessing could be effective, in fact this important technique has brought almost 10% of accuracy to the base model. The second one was made in order to reduce the amount of data, to shrink the network size and to decrease the execution time just by removing 49 columns from the total of 143 (34%). This preprocessing is lossless so the model's performances are quite similar.

## 5.1 Data fairness

An objective of this project was to try to verify if the provided data could contain biasses related to some fields. If a dataset has unbalanced informations about some entities that belong to a minority it colud lead to a prejudice during the training phase that would make the process unfair favoring those entities over others.
The dataset contained gender information so it has been checked if a bias may occour in this field. Out of 9552 records, males and females are present with the following frequencies.

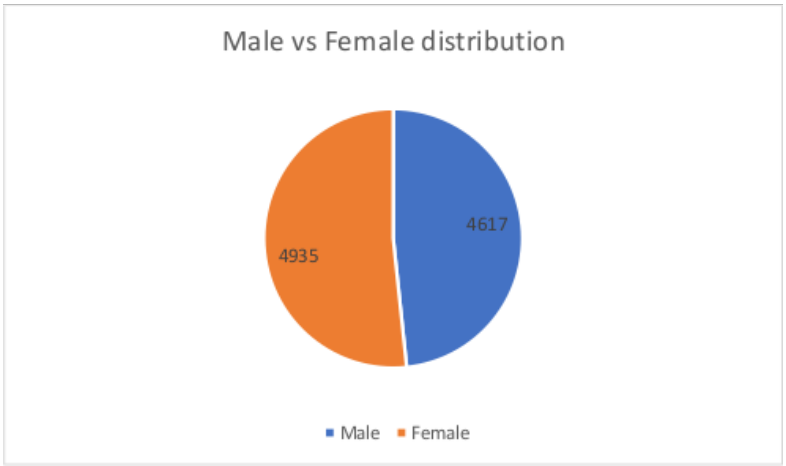


Figure 2: Distribution of males and females in the dataset

Despite the distribution seems balanced, these data weren't enough to ensure the fairness of the dataset in fact it could be possible that for some reasons the distribution of wealth among these two classes was unbalanced, leading to further prejudice. Another analysis made to check the hypothetical bias was the study of the distribution of wealth between the two classes. About males it has been found that 64.2% were "non-vulnerable households", 12.5% were "vulnerable households", 16.3% were "moderately poor" and 7% were "extremely poor".
About females it has been found that 61.1% were "non-vulnerable households", 12.8% were "vulnerable households", 17.3% were "moderately poor" and 5.9% were "extremely poor".

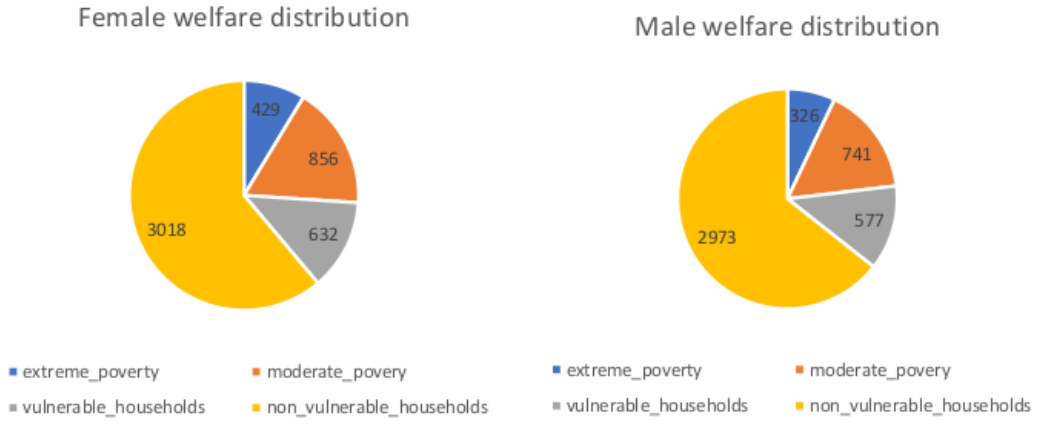


Figure 3: Distribution of welfare among males and females in the dataset

It was so observed that the distribution of wealth between the two classes looked very similar. The cosine similarity between the two distributions was 0.9990 so the dataset was balanced.

## 5.2 Model fairness

Since the data appeared to be fairly distributed between males and females, it was studied if for other reasons the model could still prefer one gender over another, granting a better rating to it. In order to evaluate it, the model was trained, the test-set was duplicated and renamed in Y and Y': the former contained the normal records from the dataset while the latter contained the

same records of Y but with the opposite gender.
The model predicted the labels for both Y and Y' and the differences of predictions due to the change of gender were analyzed.

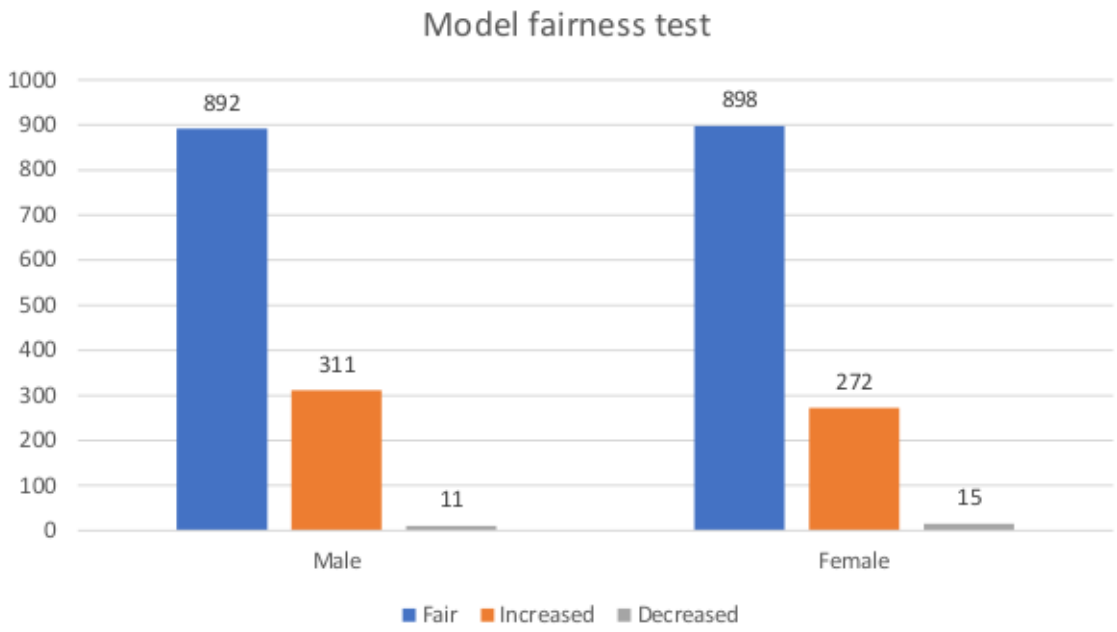


Figure 4: Changes between the predicted values of Y and Y'

It was observed that reguarding males becoming females, 73.4% hadn't suffered any changes, 25.6% increased their esteemed wealth and 1% decreased it. Reguarding females becoming males, 75.7% hadn't suffered any changes, 22.9% increased their esteemed wealth and 1.4% decreased it.
The model worked fine, on most of the predictions the result didn't change regardless of gender. Secondly the higher frequency of the increases compared to decreases is evident, so the absence of balance between the two data lead to exclude a direct connection between the change of gender and the wealth level.

# 6 Conclusions

From a subset of Costa Rica residents, it was possible to develop a fair and predictive model. People-related data are hard to predict in particular in the socio-economic context, since the cases that concern them are never-ending. Poverty is a relevant problem for many country, so a welfare-model could help a lot the workers from Inter-American Development Bank to identify who needs more aid.

# References

[1] https://www.kaggle.com/c/costa-rican-household-poverty-prediction.

[2] https://pygpgo.readthedocs.io/.