An Artificial Intelligence approach to predict mutidimensional poverty of older people from unlabelled data

Lorenzo Olearo[1,*], Fabio D'Adda[1], Vincenzina Messina[1], Marco Cremaschi[1], Stefania Bandini[1] and Francesca Gasparini[1]

[1]*Department of Computer Science, Systems and Communications, University of Milano - Bicocca, Italy*

Abstract

Despite the rapid development in very recent years of Artificial Intelligence models to predict poverty, this problem still remains an unsolved open issue especially in a multidimensional perspective. In this work we present our proposal to face multidimensional poverty in case of a fragile population, the older adults, starting from an unlabelled dataset, collected administering a proper questionnaire to about 500 individuals. Firstly a model that allows to label the collected data into three classes of poverty is proposed. Then, XGBoost and Naive Bayes classifiers are considered to solve the classification problem. Finally, after having determined the relative importance of each feature, a novel Naive Bayes model is proposed that relies on new aggregated features that represent five poverty dimensions. These aggregated features are obtained by properly combining the variables collected through the questionnaire with cut-offs defined by a domain expert.

Keywords
Multidimensional Poverty, Poverty Prediction, Older Adults, Feature Ranking, Naive Bayes, XGBoost

1. Introduction

Poverty is one of the most significant social problems according to the Organization for Economic Cooperation and Development (OECD). Poverty does not only affect developing countries but can also refer specifically to certain categories of fragile people. There is a strong relationship between poverty and human rights, as also emphasised by the Council of Europe [1]. An extreme imbalance in people's wealth leads to extreme inequality in the enjoyment of human rights. Therefore, overcoming poverty also implies reducing this inequality. Being able to predict poverty therefore becomes crucial as one of the first steps in defeating it. The application of artificial intelligence to the problem of poverty prediction is very recent, with significant early work published in 2016 [2], and a rapid increase in interest in 2021 [3]. This increase in applications of artificial intelligence to tackle poverty and, in general, to contribute to the

achievement of sustainable development goals, especially in the last two years, is also linked to the cumulative effect of the COVID-19 pandemic, the Russia-Ukraine war and climate change [4], [5]. The first approaches considered poverty and the corresponding indicators mainly referring to monetary aspects [6, 7, 8]. The recent trend, on the other hand, considers poverty as the consequence of several concomitant effects and thus requires the definition of multidimensional indicators [9],[10]. Several definitions and measurements can be defined for multidimensional poverty, related to different ways in collecting and analyzing data. Moreover, considering the various fragile people requires identifying adequate measures, to grasp the peculiarities of their conditions.

One of the major problem in predicting poverty applying AI is the lack of high quality labelled data. Different proxies can be assumed as ground truth for Machine Learning (ML) training such as the Proxy Means Test (PMT) labels. However these proxies are not easily verifiable [11]. Among different ML techniques for poverty classification, decision tree [12], random forest [11], and ensemble approaches [13] are the most used. A pioneering work [2] proposed to analyze high-resolution satellite images to predicting poverty considering a CNN model pre-trained on ImageNet. In poverty prediction the role of the different poverty dimensions is crucial. AI models have been applied also for feature selection [14] and feature ranking [15] to create models that rely only on the most important variables [12].

The AMPEL Project (Artificial intelligence facing Multidimensional Poverty in ELderly) faces the problem of predicting poverty of a particular fragile population: the older adults. Especially in this case, income-based indicators are poor proxies of material conditions [16, 17] whereas non-monetary ones improve our understanding of who is poor. The final aim of the project is to define a three level poverty risk classifier useful to identify subjects that need a prompt support, especially in case of emergencies. As a secondary outcome of no less importance, the project aims at identifying which indicators in the case of the elderly population may be the most meaningful to describe multidimensional poverty. To this end, a proper questionnaire that correlates several aspects regarding economic status, needs, health conditions, loneliness, social interactions, among others, have been designed. Thanks to the help of Auser volunteers, the questionnaires have been delivered to around 500 elderly people in Lombardy, collecting a valuable amount of data.

In this paper, we report our classification approach to identify a three-level risk of poverty, considering the following main issues:

1. Pre-process the collected data in order to obtain a clean dataset and select robust features to feed ML algorithms;
2. Apply an automatic procedure to label the collected data, taking into account the multi-facet aspects of poverty, that initially relies on the intervention of domain experts;
3. Applying a proper model to understand how different features can contribute to identifying poverty;
4. Defining multiple classifiers based on new features vectors that combine the original features considering their relevance.

The paper is organised as follows. In Section 2 the data and its sources are presented. In Section 3, the proposed framework of analysis is described. In Section 6 a brief description of the technologies used for the implementation is provided, emphasizing the data visualization

dashboard that has been implemented. Eventually, we conclude this paper and discuss the future direction in Section 7.

2. The AMPEL Dataset

To collect the dataset, a questionnaire properly defined by domain experts was administered to older people by trained interviewers to investigate various aspects of their living conditions. These questions cover topics such as monetary aspects, health status, environment, social networks, and quality of life, resulting in a total of 125 variables (features) for each individual. The questionnaires were administered to 496 people, (336 females) with an average age of 76.4 (standard deviation = 8.9).

Table 1
Municipalities where the data have been collected to create the AMPEL dataset.

AUSER Senior Center	Number of elderly
LECCO	71
LODI	62
CANTU'	51
CINISELLO	61
CERNUSCO SUL NAVIGLIO	31
PIOLTELLO	59
PREALPI-MILANO	33
COLOGNO	62
GALLARATE	66
TOTAL	**496**

The collected variables can be grouped into two categories: i) Categorical variables: variables which assume a fixed set of values (*e.g.*, private health insurance coverage: yes or no); ii) Numeric variables: variables which assume numeric values, both discrete and continue (*e.g.*, the number of individual that live in a house). It should be noted, however, that in the collected dataset numerical features are considerably less than categorical ones, a topic that will later be addressed in Section 3.1.

With the supervision of a domain expert, each of these features has been assigned to one of the five poverty dimensions here listed:

- **Maintenance Capacity**: the financial situation of an individual;
- **Consumption Deprivation**: organises some information related to the affordance capacity of the individual;
- **Health Status**: collects health status features related to the individual;
- **Housing Facilities**: contains all the information that describes the conditions of the dwelling and the neighbourhood in which the individual lives;
- **Social and Context Deprivation**: reports information about social relations.

3. The proposed framework of analysis

The prediction of multidimensional poverty using innovative AI-based solutions requires fixing methodological issues in order to face the complexity of the task, especially in this case in which
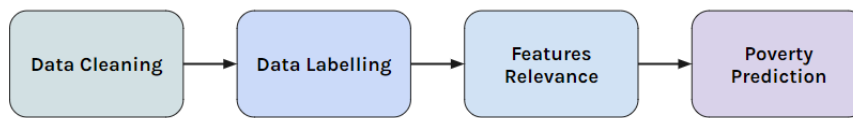
Figure 1: The proposed framework

the dataset which is composed by a limited sample size (496 individuals) and has missing values due to the lack of information from participants. In this section, a methodology addressing this critical data gap, and the distribution of poverty among individuals within the dataset are illustrated through a pipeline of four distinct phases, as illustrated in Figure 1:

1. **Data Cleaning**: This step aims to identify and correct errors or inconsistencies in the dataset to improve its quality, accuracy, and reliability. It is a crucial step for the next phases because the quality of the data directly impacts the validity and effectiveness of the employed machine learning model. Moreover, it requires feature engineering to correct for type heterogeneity within the dataset.
2. **Data Labelling**: The dataset does not explicitly tell the distribution of poverty across the population, so this fundamental step implements an approach that allows to classify individuals in different poverty levels.
3. **Feature Relevance**: The algorithm relies on multidimensional poverty in order to consider consider the different level of deprivation of an individual. The purpose of this phase is to find which features have the greatest impact on definition of poverty.
4. **Poverty Prediction**: The last step aims to build a machine learning model by using information such as, labelled data and feature relevance, to classify new, previously unseen data instances. The goal is to define a three level poverty predictor.

3.1. Data Cleaning

This dataset requires essential preliminary steps of data pre-processing to clean and normalise the data entries. Firstly, generalities are removed, along with other irrelevant features and others with too many missing or noisy values. When possible, however, missing values are filled with relevant ones. For example, if an individual does not have a certain condition, they did not answer the relevant question in the form, all such entries are thus filled with a categorical value representing the absence of that specific condition.

The dataset is composed of data collected from a survey. Some of its features are answers to open-ended questions and as such, they present considerable name heterogeneity: answers that are semantically equivalent but syntactically different. To address this heterogeneity, when possible, the various semantically different values are mapped into a common one, otherwise, thanks to the reduced dimensions of the dataset, these values are manually corrected one at the time.

Another considerable issue of this dataset is represented by features that are related and dependent, features such as the fact an individual is retired from work can be clearly related to

features describing the job which an individual is engaged or features describing if an individual is unemployed or not. To address these redundant information, the dataset is simplified by reducing its dimensionality, these groups of redundant columns are then collapsed into a single meaningful one.

The dataset is composed for the majority of categorical features but some numerical ones are present, such the age of a given person or the number of cigarettes an individual smokes over the course of a week.

Considering the small number of observations in this dataset, namely 496, the numerical columns are simplified reducing their information by *binarizing* their values. For example, instead of representing the number of people that live with an individual, a possibly challenging numerical value to handle in an otherwise categorical dataset, this information is transformed representing whether this individual lives alone or not. This type of feature engeneering is applied with the same logic over all the numerical features remaining in the dataset thus transforming it into a fully categorical dataset.

The steps illustrated above are repeated on the whole dataset and allow reducing the number of features from 125 to 103.

3.2. Data Labelling

Since the dataset does not contain information about the distribution of the poverty, it is necessary to define a process that is able to introduce this information, providing labelled data. This issue has been solved applying the framework described in a previous work [18] and firstly proposed by Liberati et al. [19]. This approach seeks to assess the likelihood of poverty for each person by employing a set of vector weights to understand the various aspects of multidimensional poverty. Instead of assigning a single weight to each variable, this technique allows for the generation of a representation of all potential weight combinations. These weights enable the classification of individuals in the dataset into three poverty levels: high (identified with red color), medium (yellow color), or low (green color). The implementation is the same as reported in [18], just properly adjusting the initial deprivation cut-offs, according to the new dataset. In order to understand how the vector space has been built, it's necessary to summarize the steps of the labelling process, which are also illustrated in Figure 2:

1. **Load Dataset**: This is a preliminary step which is necessary to load the dataset and the relative cut-offs. The purpose is also to prepare the data for the next steps of the pipeline.

2. **Deprivation Matrix**: The first phase relies on mapping the initial dataset on a *deprivation matrix* by applying the binary cut-offs properly defined by a domain expert for this data. The resulting matrix is a binary matrix that indicate, if an individual is deprived or not, with respect to each feature.

3. **Deprivation Score Matrix**: The *deprivation matrix* enables to calculate a poverty score for each individual by weighting each feature with a predefined score. Instead of defining a single vector of weights, Liberati et al. [19] proposes to randomly generate a set of m weight vectors from a uniform distribution, allowing us to explore better the feasible weight space. In this work we use $m = 10000$. The final matrix contains m deprivation scores, for each individual, based on changes in weight vectors. For this reason, each

vector can be considered as an **embedding representation of the poverty level of each individual**. Eventually, the number of random vectors is an hyperparameter to be discovered by stressing the model.

4. **Poverty Indicator Matrix**: Starting from the *deprivation score matrix*, the *poverty indicator matrix* is defined. This matrix represents the poverty ranks of each individual based on their deprivation scores within the considered population. For each vector of weights, the rank of each individual is computed by adding the number of individuals whose deprivation score is higher than the deprivation score of the considered individual, plus one. In simpler terms, the score is one plus the count of individuals who are multidimensionally poorer than the current individual. Therefore, the higher score the lower the poverty level.

5. **Probability Matrix**: The last phase builds a *probability matrix*, which represents the probabilities of individuals being in poverty. Each entry in the matrix indicates the likelihood of each individual of occupying a rank from 1 to N in the final poverty matrix.

$$
\begin{bmatrix}
2 & 2 & 2 & 8 \\
2 & 2 & 2 & 6 \\
1 & 2 & 3 & 5 \\
2 & 1 & 2 & 5 \\
1 & 2 & 1 & 1
\end{bmatrix}
\longrightarrow
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1
\end{bmatrix}
\times
\begin{bmatrix}
0.10 & 0.36 & 0.03 & 0.88 \\
0.53 & 0.67 & 0.85 & 0.19 \\
0.71 & 0.31 & 0.08 & 0.99 \\
0.27 & 0.22 & 0.75 & 0.27
\end{bmatrix}^\top
=
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0.88 & 0.19 & 0.99 & 0.27 \\
1.01 & 1.57 & 1.78 & 1.29 \\
1.24 & 0.86 & 1.30 & 0.49 \\
0.98 & 0.72 & 1.70 & 0.54
\end{bmatrix}
\longrightarrow
\begin{bmatrix}
5 & 5 & 5 & 5 \\
4 & 4 & 4 & 4 \\
2 & 1 & 1 & 1 \\
1 & 2 & 3 & 3 \\
3 & 3 & 2 & 2
\end{bmatrix}
$$

Figure 2: Data labelling: the first matrix represents for each row a subject, for each column a feature. The second matrix is the Deprivation Matrix, obtained applying the cut-offs defined by the domain expert. This matrix is multiplied with a matrix of weights, to obtain the Deprivation Score Matrix. In our procedure we extract 10000 different weight vectors from a uniform distribution. The last matrix is the Poverty Indicator Matrix, that reports the rank of poverty of each individual within the considered population.

From the Poverty Indicator Matrix the individuals can be classified into three levels of poverty, high (visually coded with red color), medium (visually coded with yellow color) or low (visually coded with green color).

Once individuals are labelled, also the associated vectors in the *Deprivation Score Matrix* can be referred to the three poverty levels.

A dimensionality reduction technique such as T-distributed Stochastic Neighbor Embedding TSNE [20], which is commonly used in machine learning and data analysis for visualizing high-dimensional data in a lower-dimensional space, can be used. This representation allows us to visualize how individuals are distributed in the representation space varying the number of random weight vectors used, and to assess whether it is possible to visually distinguish poverty groups. This analysis allows to understand how the number of random vectors changes the representation space of individuals, as reported in Figure 3, considering 5, 10, 50, 100, 1000 and 10000 weight vectors respectively.

When the number of vectors increases, the representation space changes, and the poverty groups become more separated. This behavior better explains the benefits offered considering a set of randomized vectors:
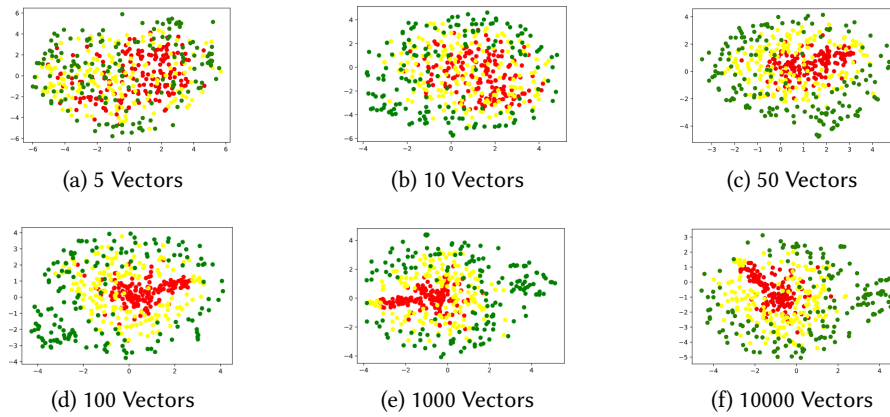
(a) 5 Vectors   (b) 10 Vectors   (c) 50 Vectors

(d) 100 Vectors   (e) 1000 Vectors   (f) 10000 Vectors

Figure 3: TSNE representation of vector spaces by varying the number of weight vectors

- **Exploration of Solution Space**: Randomized vectors allow to explore a wider range of solutions within the feasible weight space. So, having a unique vector of weights, the solution would be locked into a single one. In contrast, randomized vectors provide a more comprehensive view of the solution space, which can be beneficial in cases where the optimal solution is not known in advance.
- **Robustness**: Randomized vectors can make the optimization of the algorithm more robust to variations in the data. Randomisation can help mitigate this problem by making the solution less sensitive to specific data characteristics.
- **Efficient Exploration**: Randomization allows for efficient exploration of the solution space, because it's possible to explore a wide range of weight vectors quickly, which is particularly useful when searching for good solutions in large and complex optimization problems.

Increasing the number of weight vectors considered, the **convergence** of the model increseas, as reported also in Table 2.

# Vectors	Red	Yellow	Green
2	174	163	159
4	167	168	161
5	162	169	165
10	167	162	167
50	164	165	167
100	163	166	167
1000	162	167	167
10000	162	167	167
100000	162	167	167

Table 2
The number of individuals within each group is reported varying the number of weight vectors. Increasing this number, the groups remain stable.

4. Feature Relevance

One of the key aspects of this project lies in the understanding of which and how much features from different domains contribute to an individual's poverty, however, with such a complex categorical dataset this task becomes challenging.

The strength of the correlation between the features is firstly analyzed thought the Cramer's V [21], a measure of association between two nominal variables based on Pearson's χ^2 test. From the resulting correlation matrix, it seems that there is a significant correlation only among features related to the health status, while no other significant correlation among other variables has been found.

To try to achieve considerable results, a diametrically opposite approach was attempted using machine learning models. In particular the information gain of each feature in a tree based model, in this case XGBoost [22], is investigated.

XGBoost is used to fit boosted trees for the 3-classes classification task. It allows for the exploitation of the tree-based structure of the classifier to better understand the features in our dataset, furthermore, it allows us to operate with categorical data.

Once fitted, an XGBoost model returns the importance of each feature in the dataset the model was trained with. This importance can be defined following different metrics, in this case particular interest lies in the **gain score** of each feature which is defined as *the average gain across all splits the feature is used in*. The gain of a certain feature represents its relative contribution in the classification for each of the trees in the model, it depicts the amount of information gained about the target class after adding the relative split in the model.

The information gain of a tree-based model is a powerful metric but it can be misleading, especially in our case in which the dataset has a large number of features (103) while the observations are unfortunately low (496), the model is at high risk of overfitting as verified in our experiments, which are available at the following link: AMPEL Code.

To deal with this issue a 5-Fold cross validation process is implied. The entire dataset is initially divided into two subsets: 80% of it will be used for training while the remaining 20% is retained for testing. In the training subset of the dataset, a 5-Fold cross validation of the XGBoost model is carried on, all of the 5 models fitted in the process are saved, along with their gain scores. To avoid overfitting, these scores are then averaged.

Moreover, from the gain scores of the features within each group, it is also possible to compute the relevance of each of the five dimensions, as reported in Figure 4, that shows a fairly balanced distribution of importance.

Computing the relevance of the features and their groups is a fundamental task, it allows for a subset of questions to be identified and focused upon in case a new questionnaire is to be submitted to another sample of individuals.

5. Towards the prediction of multidimensional poverty

Having properly pre-processed the dataset and assigned the poverty labels, the problem of multidimensional poverty prediction can be faced. Three distinct classification models are here investigated: XGBoost, Categorical Naive Bayes and a novel hybrid approach that combines XG-
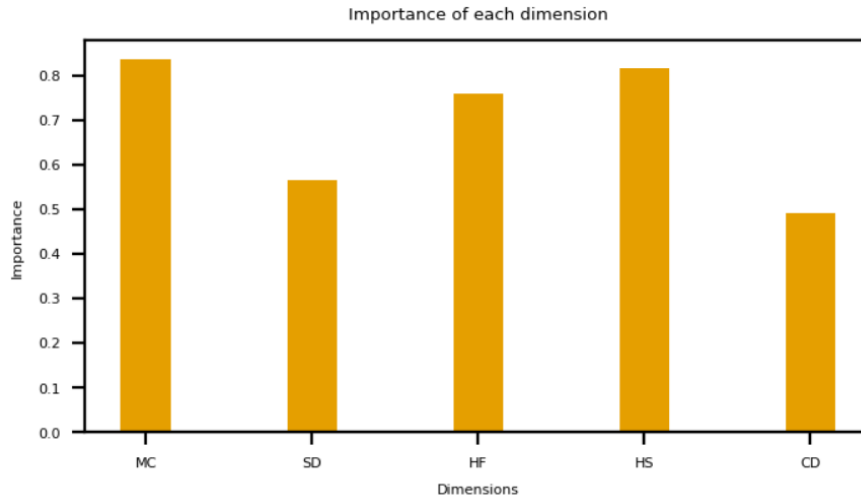
Figure 4: Category Relevance computed using the gain scores obtained from XGBoost

Boost feature gain scores with a Naive Bayes classifier, considering the five poverty dimensions defined in Section 2.

In all of the following analysis, the same 80% of the dataset previously used for the feature relevance is used to train the presented models while the remaining 20% is retained for the testing, all of the models are thus trained on the same data and evaluated on the same test set.

In this section the proposed models are evaluated considering three different metrics: *accuracy*, *recall* and *F1-score*.

5.1. The XGBoost classification model

Following our previous work [18] and having used it to understand the impact of each feature, the first model analyzed is XGBoost in its categorical classification form, this model provides the baseline for subsequent predictive models. This model is chosen as it offers significant interpretability through the visualization of the aforementioned feature importance, moreover, both training and inference on the model are remarkably fast. With these considerations, once fitted, our XGBoost model achieves the 79% of overall accuracy across the test set.

Class	Accuracy	Recall	F1-Score	Support
Green	0.73	0.90	0.81	21
Yellow	0.72	0.70	0.71	37
Red	0.89	0.81	0.85	42

Table 3
Performance of the XGBoost classifier

5.2. The Categorical Naive Bayes classification model

With the aim of having an explainable model, a Bayesian approach is analyzed, more precisely a Naive Bayesian Classifier. No conditional dependence is assumed between the various features of the dataset. This is a reasonable assumption as the questions asked to the individuals cover different subjects one from the other, albeit of the same topic in some cases. Furthermore, finding any conditional dependence between such a large number of features is non-trivial and requires rich domain knowledge.

Once again, the model is trained on the entire train set, this time achieving slightly better results than the XGBoost model with a global accuracy of 84% as shown in Table 4.

Class	Accuracy	Recall	F1-Score	Support
Green	0.89	0.76	0.81	21
Yellow	0.72	0.89	0.80	37
Red	0.94	0.81	0.87	42

Table 4
Performance of the Categorical Naive Bayesian classifier

Increasing the number of features tends to add more information that the model can learn. Ideally this can lead to better discrimination between classes. However, this can also have its downsides: as the number of features grows so does the risk of overfitting the train data: the model might learn apparently unrelated noise while loosing the ability to later generalize on the test set.

Naive Bayes classifier are known for their simplicity and effectiveness in many classification tasks [23], however, they tend to be less informative as the number of different features increases. In order to create a model as explainable as possible, it can be interesting to study how the performance of the model changes while increasing the number of features it is trained with.

In Figure 5, we present the performance evaluation of the model employing a 5-fold strategy, showcasing the model performance as we increment the number of features. These features are arranged based on their relevance in descending order, determined by computing the average feature gain scores across the 5 cross-validated XGBoost models discussed in Section 4.

The trend of the accuracy seems irregular varying the number of features considered, probably due to the small number of observations in the dataset. However, it is still possible to appreciate the incremental tendency, that seems to stabilizes around the top 60 ones.

5.3. The Naive Bayes multidimensional poverty classifier

This model is introduced with the aim of having a predictive model as explainable as possible, capable of operating with a reduced number of features but in a multidimensional environment. These dimensions correspond to the five groups defined in Section 2: *Social Deprivation, Maintenance Capacity, Consumption Deprivation, Housing Facilities and Health Status.*

For each dimension, a new feature is computed by linearly combining the XGBoost information gain of each feature with its relative cut off defined by the domain expert. In order to improve the generalization power of the model, the gain used for this linear combination is the one averaged from the 5 extracted cross-validated models in Section 4.
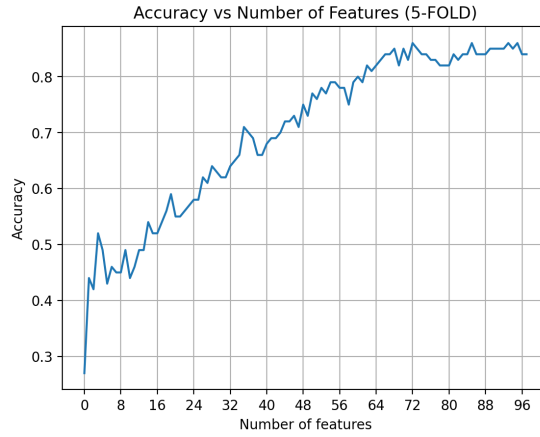
Figure 5: Accuracy of the Categorical Naive Bayes classifier with respect to the number of features selected from a list, ordered with respect to the gain scores obtained applying the XGBoost model.

The procedure to compute the five new features is reported below. Starting from the binary deprivation matrix defined in Section 3.2, each row describes the features where an individual is considered deprivated. These rows are here referred as *deprivation vectors*. Each column, instead, represents the distribution of individuals that exceeds the cut-offs defined by the domain expert.

Considering now the gain scores obtained for each feature vector, these are grouped by respect of their previously defined poverty dimension and linearly combined with their relative *deprivation vector* as shown in Equation 1, where a is the importance vector for the group of features of the considered poverty dimension, each a_i *with* $i \in \{1, \cdots, n\}$ is the gain score of the i-th feature, while b instead is the deprivation vector where each b_j *with* $j \in \{1, \cdots, n\}$ represents the feature j-th of that group.

$$a^\top b = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix} = deprivation\ score \tag{1}$$

These computations result in five numerical values referred as *deprivation scores* representing the "magnitude" of the deprivation an individual is subjected for each of the five dimensions.

Each individual at this point is encoded into 5 values that will now be used to determine their poverty class. To this end, a Naive Bayes classifier is thus applied having a parent node for each of the 5 classes of features as shown in Figure 6.

The classifier built on top of this pipeline shows good performance as reported in Table 5 reaching an overall accuracy of 89% on the test set which it is important to mention has never been used in the entire pipeline except to assess this metric. This result shows how well the model is able to generalize by using the average of the gain extracted from the cross-validated XGBoost models introduced in Section 4.
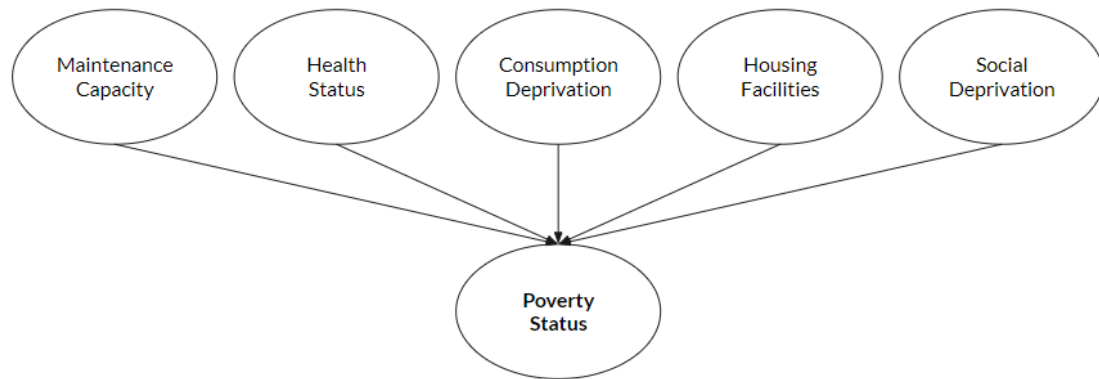
Figure 6: Structure of the Naive Bayes classifier, where each of the 5 parent nodes represents the score in that poverty dimension for each individual

Class	Accuracy	Recall	F1-Score	Support
Green	0.95	0.95	0.95	21
Yellow	0.81	0.92	0.86	37
Red	0.95	0.83	0.89	42

Table 5
Performance of the Naive Bayes multidimensional poverty classifier

6. The AMPEL Project Dashboard

The implementation of the project involves multiple independent software modules that co-operate by exchanging data resulting from the analyses described above. The figure 7 shows the different pieces of software implemented and deployed by using *Docker containers*. A brief description of each backend component is reported below:

- **MongoDB**: *MongoDB* has been used to save AMPEL data and it is an open-source NoSQL database management system. It falls under the category of document-oriented databases and is designed to handle large volumes of unstructured or semi-structured data.
- **API**: The deployable code runs in the *AMPEL API* container because they enable the integration of diverse applications and technologies.
- **Jupyter Notebook**: Most of the analyses have been implemented in a *Jupyter Notebook* because it allows to create and share documents containing live code, equations and visualizations.

The last service reported in figure 7 is the **Dashboard** (reported on figure 8). It plays a crucial role because it allows the visual analysis of the results. carried out in the previous sections. The **Dashboard**, is a visual representation of key information, data, metrics and performance indicators organized and displayed in a centralized and easily accessible manner. In this case, the dashboard has been designed to provide a quick and concise overview of important information, allowing experts to monitor, analyze, and make informed decisions based on real-time and historical data. Among the charts available in the dashboard it's possible to find:
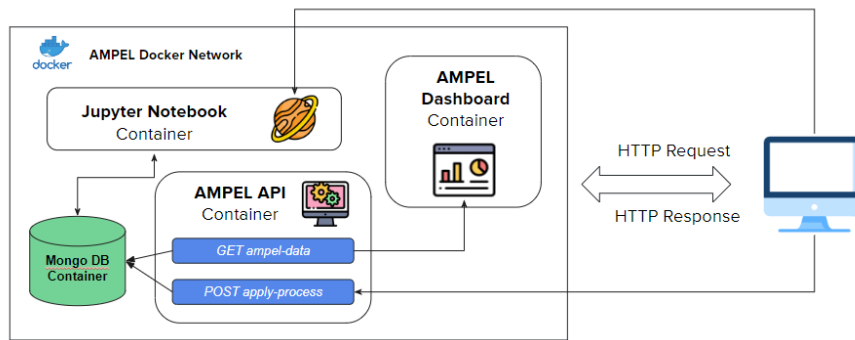
Figure 7: AMPEL Software Architecture

- **Poverty on degree**: Poverty is analyzed by categorizing individuals according to their educational attainment.
- **Poverty on Gender**: Poverty is analyzed by categorizing individuals according to their gender.
- **Map Poverty**: Poverty is analyzed by highlighting on a geographical map the areas with the highest concentrations of poverty.

7. Conclusion and future works

The proposed AI approaches to predict multidimensional levels of poverty have shown good performance. The feature relevance analysis allows to define poverty scores, related only to five poverty dimensions, starting from the original features properly linearly combined. This approach allows to train a model on feature vectors of a lower dimension, that preserve the valuable information of the original set of features. However it is worth noting that the labelling procedure could have introduced a bias in the final results. As well as the definition of the five dimensions could be revised and extended. To test our proposal and make it more generalizable, we plan to adopt other labelled datasets available in the literature, to test both the labelling procedure and the proposed Naive Bayes multidimensional classifier.

Acknowledgments
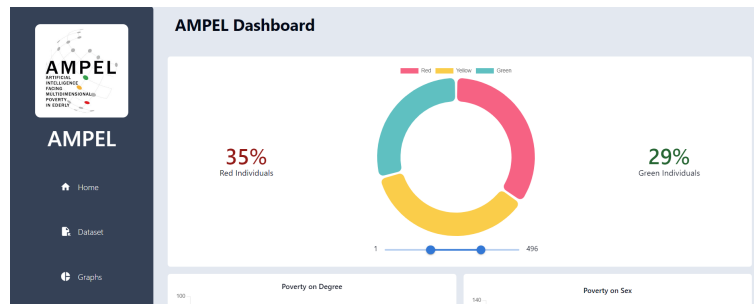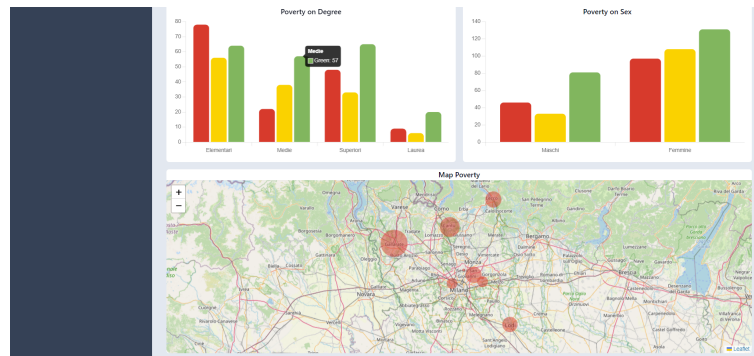
(a) Dashboard Panel

(b) Dashboard Charts

Figure 8: Figure *a* displays the panel utilized by the domain expert for examining poverty within the population, whereas Figure *b* presents the charted results obtained from the investigation.

References

[1] Council of europe, https://www.coe.int/en/web/portal/home, ???? Accessed: 2023-09-16.

[2] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, S. Ermon, Combining satellite imagery and machine learning to predict poverty, Science 353 (2016) 790–794.

[3] A. Usmanova, A. Aziz, D. Rakhmonov, W. Osamy, Utilities of artificial intelligence in poverty prediction: a review, Sustainability 14 (2022) 14238.

[4] S. K. Satapathy, S. Saravanan, S. Mishra, S. N. Mohanty, A comparative analysis of multidimensional covid-19 poverty determinants: An observational machine learning approach, New Generation Computing 41 (2023) 155–184.

[5] W. Bank, Poverty and shared prosperity 2020: Reversals of fortune, The World Bank, 2020.

[6] A. B. Atkinson, Income distribution in oecd countries, Evidence from Luxemburg income study (1995).

[7] M. Biewen, Income inequality in germany during the 1980s and 1990s, Review of Income and Wealth 46 (2000) 1–19.

[8] M. F. Förster, M. Mira D'Ercole, Income distribution and poverty in oecd countries in the second half of the 1990s, Income Distribution and Poverty in OECD Countries in the Second Half of the 1990s (February 18, 2005) (2005).

[9] P. Townsend, Poverty in the United Kingdom: a survey of household resources and

standards of living, Univ of California Press, 1979.

[10] M. Terraneo, A longitudinal study of deprivation in european countries, International Journal of Sociology and Social Policy (2016).

[11] J. H. Mohamud, O. N. Gerek, Poverty level characterization via feature selection and machine learning, in: 2019 27th Signal Processing and Communications Applications Conference (SIU), IEEE, 2019, pp. 1–4.

[12] N. S. Sani, M. A. Rahman, A. A. Bakar, S. Sahran, H. M. Sarim, Machine learning approach for bottom 40 percent households (b40) poverty classification, Int. J. Adv. Sci. Eng. Inf. Technol 8 (2018) 1698.

[13] A. Abu, R. Hamdan, N. Sani, Ensemble learning for multidimensional poverty classification, Sains Malaysiana 49 (2020) 447–459.

[14] T. P. Sohnesen, N. Stender, Is random forest a superior methodology for predicting poverty? an empirical assessment, Poverty & Public Policy 9 (2017) 118–133.

[15] D. Arribas-Bel, J. E. Patino, J. C. Duque, Remote sensing-based measurement of living environment deprivation: Improving classical approaches with machine learning, PLoS one 12 (2017) e0176684.

[16] M. Adena, M. Myck, Poverty and transitions in health in later life, Social science & medicine 116 (2014) 202–210.

[17] B. Nolan, C. T. Whelan, Measuring poverty using income and deprivation indicators: alternative approaches, Journal of European Social Policy 6 (1996) 225–240.

[18] F. D'Adda, M. Cremaschi, E. Messina, M. Terraneo, S. Bandini, F. Gasparini, A three level prediction of multidimensional poverty in elderly, in: Proceedings of the Italia Intelligenza Artificiale - Thematic Workshops co-located with the 3rd CINI National Lab AIIS Conference on Artificial Intelligence (Ital IA 2023), volume 3486, CEUR, 2022, pp. 544–549.

[19] P. Liberati, G. Resce, F. Tosi, The probability of multidimensional poverty: A new approach and an empirical application to eu-silc data, Review of Income and Wealth (2022).

[20] L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of machine learning research 9 (2008).

[21] H. Cramer, Mathematical methods of statistics, princeton univ, Press, Princeton, NJ 27 (1946).

[22] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[23] I. Rish, et al., An empirical study of the naive bayes classifier, in: IJCAI 2001 workshop on empirical methods in artificial intelligence, volume 3, 2001, pp. 41–46.