Poverty Level Prediction for Households

1st Chiranth Jawahar
Computer Science Engineering
PES University
Bangalore, India
chiranthjawahar@gmail.com

2nd Namrata R
Computer Science Engineering
PES University
Bangalore, India
namrata.ajjampur@gmail.com

3rd Diya Chandra
Computer Science and Engineering
PES University
Bangalore, India
diyasateesh96@gmail.com

***Abstract*—This project explores data consisting of details of various socio-economic factors that effect households in the Latin American nation of Costa Rica. We look to predict the poverty standard of each household and classify them into 4 distinct categories. These predictions are done with the aid of feature selection and normalization of the data to fit various machine learning models that attempt to give accurate results. The data is then explored to provide details regarding the factors that strongly effect each category to help recommend methods for improvement and appropriate allocation of social welfare benefits.**

***Index Terms*—classification, poverty, random forest, xgboosting**

I. Introduction

Poverty levels have been hard to pinpoint in developing nations due to the scarce availability of data. This problem also occurs due to the lack of any substantial model which accurately represents these categories thereby causing unscientific norms to be used for present day classification leading to a certain amount of misclassification of social welfare benefits. This classification of poverty levels helps governments and organizations allocate social welfare benefits in an efficient and effective manner thereby expediting the process of socio-economic upliftment in the region causing an adverse positive effect in these nations. The classification that takes place also allows clearer understanding of the major pain points of each poverty class by observing the most notable features, this allows for more care to be taken in the handling of those specific factors thereby helping in alleviating individuals of that status.

II. Previous Work on Similar Problems

A. Random Forest Implementation

Random forests are an ensemble of decision trees where they are trained using a sample of the data and the class that receives the majority of votes is obtained. Random forests run considerably fast and are known to have high accuracy when compared to other classification methodologies. It is taken into consideration that generally random forests do not overfit as for a large number of trees, the generalized error value converges to a limited value under a strong law of large numbers [1].
The method used is in previous work[2] is as follows:

- First a random sample of observations are taken and the subsequent bootstrap samples for other trees are taken.
- A subset of variables much smaller than the total number of variables are taken and using the Gini score the best split is found.
- An out of bag (OOB) prediction is obtained from the majority vote across tree samples whose observations were not taken in the bootstrap sample.

This method relied heavily on the income of individuals as its primary differentiator and hence removed all individuals who did not have any income mentioned. To evaluate the importance of a feature, [1, 3] a technique was proposed by which, for all trees in the forest the average of an impurity decrease measure for all nodes in which the feature is concerned is taken into consideration. The feature which has the largest decrease of this measure is considered to be the most important. They achieve this using the Mean Decrease Gina (MDG).
Cases in which any sort of missing values are present are also removed from the sample. The results obtained show factors that strongly effect whether an individual is poor or not and helps classify them into 2 base categories.There is also a noticeable limitation where a lot of features were completely removed due to the presence of a large number of missing values.
a few issues that can be noticed is that due to multiple cases and features being removed an accurate representation of the population is not truly seen and this may cause the result to report incorrect conclusions. Another case where this methodology has issues is with regard to its static barrier of stating that individuals are only classified into 2 groups which makes it harder to notice and differentiate between extreme cases in the same class which in actuality may need to be treated differently.

B. XGBoosting

XGBoosting is used for classification and detection of epilepsy[4] in a binary format (healthy; patient has epilepsy).This is done by finding the combination of features that show the best predictive power in the binary classification. Xgboosting is an implementation of gradient boosting decision trees. the method for implementation is as follows:

- feature selection is done using filter and wrapper methods. Filter methods allow fast computation and provide feature ranking which facilitates removal of unnecessary features. Wrapper approaches uses a classification algo-

rithm for an evaluation of a subset of features by training and testing with cross validation on the subsets.
- A learning rate of 0.01 for better generalization is taken with xgboost[5].The number of boosting trees that are used as estimators is set to 1200. A subsample value of 0.7 over the default value of 1 is taken to prevent overfitting. The model's complexity is also decreased by making maximum depth 3 over the default depth of 6.

The results that this model received were fairly positive with the AUC being the performance metric with a value of 0.91. A few notable issues were with present as feature selection in their case was moderately unsuccessful, making the classification with the initially employed methods inaccurate, due to this further cross validation was required. This scenario also consists of almost perfect data which required little to no preprocessing, which is not the case in most real world scenarios such as the problem being addressed.

III. Proposed Problem Statement

The need for classification of households based on their situational factors into specific levels of poverty (or absence of it) to help in allocation of social welfare benefits suitably amongst the population and to help identify the factors that strongly affect different levels for general upliftment.

IV. Problem Approach

The data of each individual in a household is acted upon along with their collective features. Poverty levels are broadly categorised into 4 tiers which are,

- Not vulnerable
- Mildly vulnerable
- Vulnerable
- Extreme poverty

The process involves starting with preprocessing of the data and feature engineering followed by exploratory data analysis to gain insights and then proceed with modelling of the data to find to the most suitable predictor for the problem at hand.

A. Data

The dataset used is from Kaggle which is an online community of data scientists and machine learners. The training dataset contains 143 columns and 9557 rows. Each column in the dataset represents the various features that most likely affect the households. Each row represents a member of a given household and the information of the various features.The target attribute has been classified in to classes indication their poverty status. Each household is uniquely identified by the attribute "idhogar" and each row is uniquely identified by an "Id". The columns indicate whether a household has access to certain amenities, provides insight to their educational background and their present living conditions. The squared values of certain features are also provided as separate columns.

B. Constraints

The data set is specific to the Latin American nation of Costa Rica which makes it difficult to translate these properties worldwide given the massive difference in features when compared to other nations with socio-economic factors of a different scale.

V. Methodology

A. Preprocessing

In Real world data can generally be incomplete(lacking attribute values,missing data, lacking certain attributes of interest, or containing only aggregate data), noisy(containing errors or outliers) or inconsistent(containing discrepancies in the values that the attributes take)

Preprocessing is the technique that involves transforming raw data into an understandable format and handling the above cases.It broadly consists of Data cleaning, Data Transformation and Data reduction.
Feature engineering is about creating new input features from existing ones, based on domain knowledge and understanding of the dataset.

- Handling missing data or data that is inconsistent: The dataset had missing values int the below fields as follows:

rez_esc	7928
v18q1	7342
v2a1	6860
meaneduc	5
SQBmeaned	5

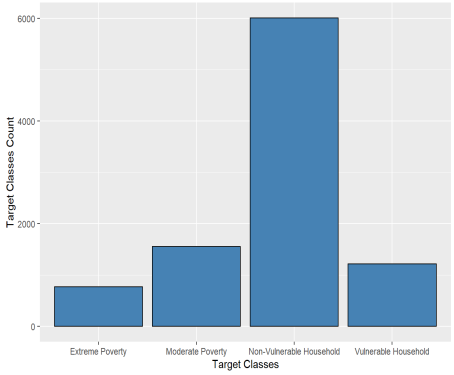
  Each of the attributes have been handled separately
- The dataset is grouped based on the household that the members belong to and each member of the household is reassigned the same "Target" class as the head of the household. In case of families with no member as the head of the household, we check for consistency of the labels of the rest of the members(This was already consistent)
- Handling attributes with non-numeric values:

  "Id" "idhogar" "dependency" "edjefe" "edjefa"

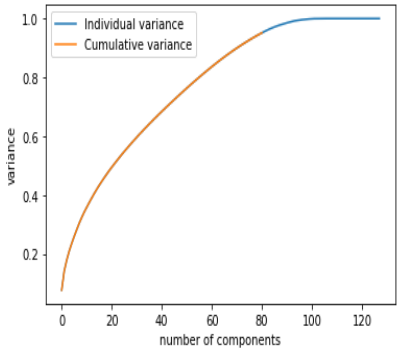
  The above attributes are of type object(python) / type factor(R) dependency(dependency rate), edfeje(years of education of male head of household) and edjefa(years of education of female head of household).They can either take values "yes" or "no" or a certain numeric value. On comparison it was observed that all "no" in dependency column corresponded to 0 in the SQBdependency column and similarly "yes" corresponded to 1. Hence the values "yes" and "no" were mapped to integer values 1 and 0 respectively and the numeric value stayed as is for the above mentioned columns. "Id" and "idhogar" remain unaltered as they uniquely identify the rows.
- Analysing and handling missing values.
  For the column "rez_esc"(Years behind in school), members who do not belong to the age group of 7-18 are

assumed to not be attending school and hence a 0 value is set to the missing values. For those within the given age group, the value is replaced by the mean of the those already present in the dataset that lie within the age group
For the column "v18q1" (number of tablets owned by a family), all the missing attributes are filled with 0 as they indicate that no tablets are owned by the family, which can be verified by comparing with "v18q" column(owns a tablet or not). An entry of 0 was made for the corresponding rows indicating that they do not own tablets.
For the column "v2a1"(Monthly rent payment), The missing values are are replaced by 0 if the head of the household owns the house, as they do not need to pay any rent.In other cases, the mean rent payment for each group of poverty level is found and missing values are replaced by the mean value found above based on the class of poverty that the household of the individual belongs to.
For the column "meaneduc"(average years of education for adults (18+)), "meaneduc" is calculated by grouping the members based on household, finding the mean of escolari(number of years of education) of all the members whose age is above 18.
Similarly the "SQBmeaned" column is also filled with the squared value of the previously discussed attribute "meaneduc"

- Attributes with duplicate values are listed : "hhsize" "hogar_total" and "agesq". These attributes are dropped from the dataset as they are redundant and do not specially contribute in classification of poverty
Once all the missing attributes are taken care of, we dropped columns which indicate the squares of the values of certain other columns. These include: SQBescolari (square of numer of years of education), SQBage (square of age), SQBhogar_total (square of total individuals in the household), SQBedjefe (square of years of education of male head of household), SQBhogar_nin(square of number of children 0 to 19 in household), SQBovercrowding(overcrowding squared), SQBdependency(squared dependency), SQBmeaned (square of the mean years of education of adults (¿=18) in the household)



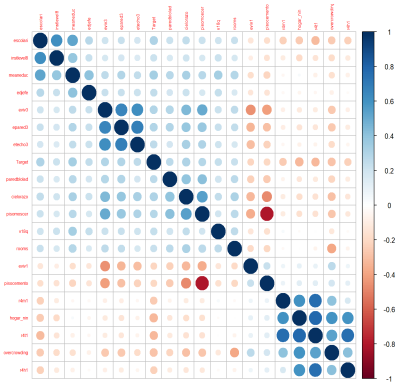
As seen from the graph, the dataset is imbalanced. Downsampling the "Non vulnerable" class would reduce the training data by a significant amount, hence upsampling is done to deal with these unequal class sizes as the data is skewed towards the majority of the population being of the lowest poverty level, not vulnerable(label 4). Classes which were smaller were increased to create equal bin sizes for equal representation.
Since the dataset consists of 130 odd columns after preprocessing, the dataset was processed using Principal Component Analysis (PCA).



From the above graph, it can be observed that 80 attributes would contribute to 90+ percent of the cumulative frequencies [6].But PCA-based feature transformations allow summarising of the information from a large number of components such as linear combinations of the original features. However the principal components are often difficult to interpret and as the empirical results in this paper indicate they usually do not improve the classification performance.

B. Exploratory Data Analysis

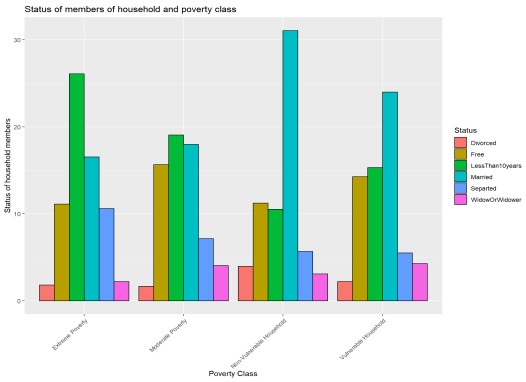
Exploratory Data Analysis (EDA) refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies and to check assumptions with the help of summary statistics and graphical representations. Thus, EDA helped us maximize our insight into the data set and extract important variables.



Initially, before the data was cleaned or preprocessed, the correlation matrix depicted highly correlated features. After a round of preprocessing and cleaning, the correlation matrix was plotted. It can be observed that the highly correlated features were removed.

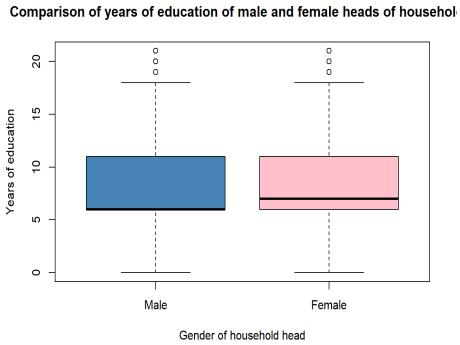
A graph showing poverty by location was plotted from which it can be inferred that households in the urban region for every poverty class are more affected than the rural households. From the graph of Average Rent payment per poverty class, it is observed that the average rent for non-vulnerable household is the highest which suggests a steady financial status for these households. The overall analysis of our dataset can be divided into three parts:

1)Visualizations of distributions of members of Household:
The proportion of people per poverty class graph shows proportion of people in each household, that is, number of adults, children and seniors and it can be observed that there is a lesser proportion of seniors in every household. The extreme poverty class seems to have the greatest number of children compared to the other poverty classes. Thus, the number of children in a household seems have a significant impact on the poverty level. Analysis of the age distribution of male and female based on poverty class was also plotted for better understanding of the features.



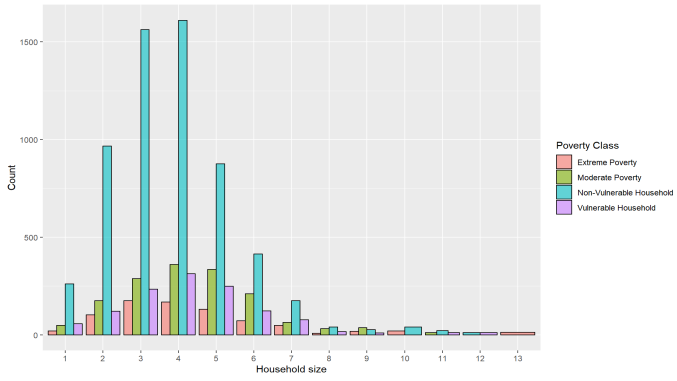
The status of the household members per poverty class was also considered during the analysis. This graph highlights the fact that the number of children in a household has a significant impact on the poverty level. The extreme poverty and moderate poverty classes have a higher number of children in the household compared to the non-vulnerable class.

2)Visualizations on Education levels of members of Household The graph showing the mean education of the household members per poverty class conveys that the non-vulnerable households have the highest average years of education while the class of extreme poverty seems to have the least average years of education which suggests that importance of overall education years of the members of the household.

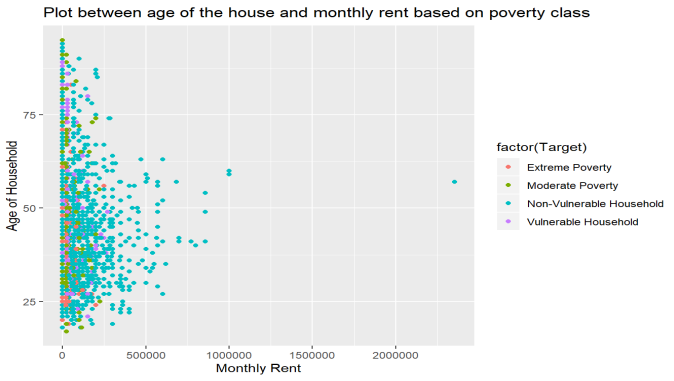


This boxplot is used to compare years of education between female and male household heads. The inference drawn here is that the average years of education for a female household head is significantly lesser than that of a male household head. Further analysis was done to find poverty designation based on the split of education levels of the members of the household.

3)Visualizations pertaining to the households was also made.
An important insight gained from this analysis of the data is the fact that basic amenities such as water, a toilet, a bathroom and electricity are available to people of all poverty classes. It can be observed that the roof, wall and floor quality for households of all poverty levels are regular or good by majority, but Non-vulnerable has the least number of ”BAD” categorizations while number of ”BAD” categorizations are more for extreme and moderate poverty state.



A graph showing count of household size per poverty class is plotted. This shows that a household size of three,four and five is common across the poverty classes. There is evidence that Non-vulnerable class’s count is significantly higher than others. This suggests that there may be an imbalance of the distribution of data. Thus, data balancing is required. Further analysis proved that a mobile phone, refrigerator and television may be owned by households of all poverty classes. However, owning a tablet or a computer in a household other than non-vulnerable household is not common.



This graph helps understand the distribution of the rent paid depending on the age of the house per poverty class. It can be seen that most of the points are clustered between the range of monthly rent 0-500000. Outliers can also be seen for the monthly rent.

C. Modelling

1) Random Forest: Random forests is a supervised learning algorithm and is an ensemble of decision trees where they are trained using a sample of the data and the class that receives the majority of votes is obtained. It can be used for Classification and Regression. Random forest classifier :

- Run considerably fast and provides higher accuracy.
- Will handle the missing values and maintain the accuracy of a large proportion of data.
- If there are more trees, it won't consider overfitting trees in the model.
- It has the power to handle a large data set with higher dimensionality.
- It also provides a good indicator of the feature importance.

Thus, due to the factors mentioned above, Random Forest would be well suited for our application and is used in the modelling phase.
Algorithm:
Select random samples from a given dataset. Construct a decision tree for each sample and get a prediction result from each decision tree. Perform a vote for each predicted result. Select the predicted result with the most votes as the final prediction.

2) XGBoost(eXtreme Gradient Boosting): XGBoost is a decision-tree-based ensemble algorithm that uses a gradient boosting framework. Generally, XGBoost is fast when compared to other implementations of gradient boosting. Gradient Boosting trains many models in a gradual, additive and sequential manner. Gradient boosting identifies shortcomings by using gradients in the loss function (y=ax+b+e,e needs a special mention as it is the error term). The loss function is a measure indicating how good the model's coefficients are at fitting the underlying data. Gradient boosting involves three elements:

- A loss function to be optimized.
- A weak learner to make predictions.
- An additive model to add weak learners to minimize the loss function.

The two reasons to use XGBoost are also the two goals of the project:

- Execution Speed.
- Model Performance.

3) Ridge Classifier: Ridge regression classifier is an extension of linear regression. It is basically a regularized linear regression model. The parameter is a scalar that should be learned as well, using a method called cross validation, it does not get rid of irrelevant features but rather minimizes their impact on the trained model. Concretely, this is implemented by taking advantage of the multi-variate response support in Ridge.

4) Decision Trees: Decision Tree algorithm is a supervised learning algorithm which can be used for solving regression and classification problems. It creates a training model which can be used to predict class or value of target variables by learning decision rules inferred from prior training data.

- A decision tree does not require normalization and scaling of data.
- Missing values in the data also do not affect the process of building decision tree to any considerable extent.
- Suitable for handling both categorical and quantitative values
- The number of hyper-parameters to be tuned is almost null.

VI. Evaluation Metrics

A. Recall

Recall is the number of true positives divided by the number of true positives plus the number of false negatives. True positives are data point classified as positive by the model that actually are positive, and false negatives are data points the model identifies as negative that actually are positive.

$$Recall = TP/TP + FN$$

B. Precision

Precision is the number of true positives divided by the number of true positives plus the number of false positives. False positives are cases the model incorrectly labels as positive that are actually negative

$$Precision = TP/TP + FP$$

C. F1-score

When the optimal blend of precision and recall is to be taken, we combine the two metrics and arrive at the F1-score. The F1-score is a the harmonic mean of both precision and recall.

$$F1-Score = (2*Recall*Precision)/(Recall+Precision)$$

VII. Results

Model	Recall	Precision	F1-Score
Ridge Classifier	0.54	0.53	0.53
XGBoost Classifier	0.73	0.73	0.73
Decision Trees	0.96	0.96	0.96
Random Forest	0.98	0.98	0.98

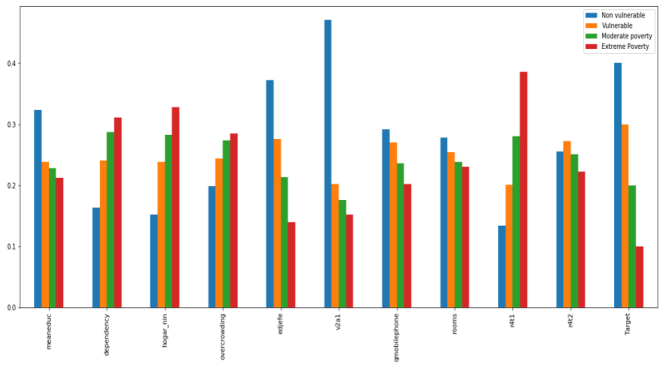
The above table represents the values of the evaluation metrics for the various models attempted. Random Forest proves to be the most effective in accurately classifying households. This is likely due to the ensemble of decision trees being more effective with respect to categorical data classification as even a singular decision tree showed positive results.

VIII. Conclusions

The model that proved to be most well suited for our dataset was the Random Forest classifier. When tested on our dataset, an F1 score of 0.98, a precision of 0.98 and recall of 0.98 was obtained. Thus, the random forest classifier gives the best results in terms of the evaluation metrics we used.

The XGBoost model resulted in a poor F1 score, precision and recall. This is because gradient boosting is a greedy algorithm and can overfit a training dataset quickly. Furthermore, boosting techniques are not suitable in cases where there is a high correlation between the attributes of the dataset.

On training the Random Forest Classifier, the top ten attributes that contributed the most to the classification of poverty levels were selected. For further analysis, the mean values for each of these important attributes, based on their poverty level, was found.



The above graph shows a comparison of the mean values of the top ten attributes for each of the labels classifying based on poverty level.
For every new entry representing an individual, classification can be implemented by testing it with the model and using the above method, comparison of these attributes for an individual can be made with the mean values of the “Non vulnerable” poverty level.This comparative study can be used to further analyze the basic amenities that an individual may require to help make their socio-economic status better and help improve their living conditions.

IX. Contributions

Namrata R	Preprocessing
C Diya	Exploratory Data Analysis
Chiranth J	Model Testing and Comparison

References

[1] L. Breiman, ”Random Forests,” Machine Learning, vol. 45, pp. 5-32, 2001.

[2] Ruben Thoplan, ”Random Forests for Poverty Classification” in International Journal of Sciences: Basic and Applied Research (IJSBAR), August 2014

[3] L. Breiman, ”Manual on Setting Up, Using, And Understanding Random Forests V3.1,” Technical Report, 2002.

[4] L. Torlay, M. Perrone-Bertolotti, E. Thomas, M. Baciu, Machine learning–XGBoost analysis of language networks to classify patients with epilepsy

[5] Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378

[6] Mykola Pechenizkiy, Alexey Tsymbal, Seppo Puuronen, PCA-based Feature Transformation for Classification: Issues in Medical Diagnostics, 2004.

[7] Mahendra Sahare, Hitesh Gupta, A Review of Multi-Class Classification for Imbalanced Data, 2012.

[8] Anita Prinzie, Dirk Van den Poel, Random Forests for multiclass classification: Random MultiNomial Logit, 2012.