**Abstract**

In predicting the poverty level of individuals in Costa Rica we converted a multinomial classification problem into a binomial one. The dataset had large number of attributes. Techniques such as exploratory data analysis, correlation and PCA were done to reduce the number of attributes for machine learning models. Out of the three machine learning techniques we employed (Naïve Bayes, Logistic Regression and Random forest), we found that Random forest gave us the best results and all the three performed better than the baseline established­. A working web application can be found at vishu133.shinyapps.io/Costarica for users to try and test various attribute combination on the three models stated above

1. **Introduction**

In Latin American, Costa Rica is facing a huge poverty issue. The Inter-American Development Bank is trying to help the poorest segment of the society. Many social programs have a hard time making sure the right people are given enough aid. It’s especially tricky when a program focuses on the poorest segment of the population. The worlds poorest typically can’t provide the necessary income and expense records to prove that they qualify.

In Costa Rica, 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 a 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.

The challenge here then is to use the other information we know about individuals and households - features such as their family size, home construction type and their access to adequate sanitation - to help identify those individuals most in need before it’s too late.

The problem is posed as one of multinomial classification. Specifically, we have four (ordinal) classes indicating different income levels and several features describing characteristics of the individual, their household, and the property occupancy.

1. **Goal**

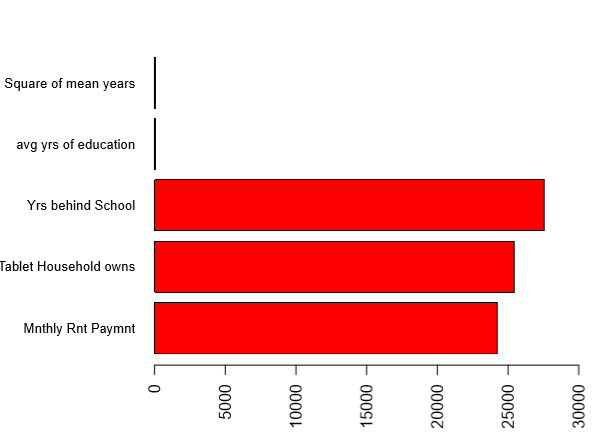
The main goal is to predict the household poverty levels and determine the driving factors for that decision.

* Reducing poverty and social inequalities
* Addressing the needs of small and vulnerable countries
* Fostering development through the private sector
* Addressing climate change, renewable energy, and environmental sustainability
* Promoting regional cooperation and integration

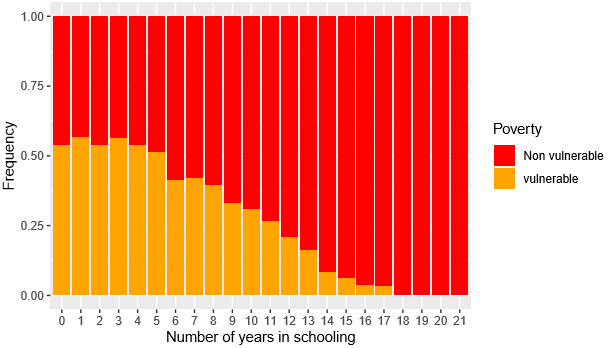
1. **Data Extraction & Exploration**

Data was extracted from kaggle.com[[1]](#footnote-1). In the dataset one row represents one person in data sample. Each row has 142 columns describing a range of attributes such as income, education, rent paid etc. The dataset already had attributes one hot encoded, making it easier to create machine learning models. Prediction variable is stored in attribute named Target. The Target attribute is an ordinal variable indicating groups of income levels. They are :

1(extreme poverty), 2 (moderate poverty),   
3(vulnerable households),4(non-vulnerable households). We performed some exploratory data analysis on the dataset by creating bar charts of each attribute overlapped with the poverty status which can be explored using the web app. We also checked the number of missing values in the data by visualizing it. Plot 1 describes missing data in attributes.



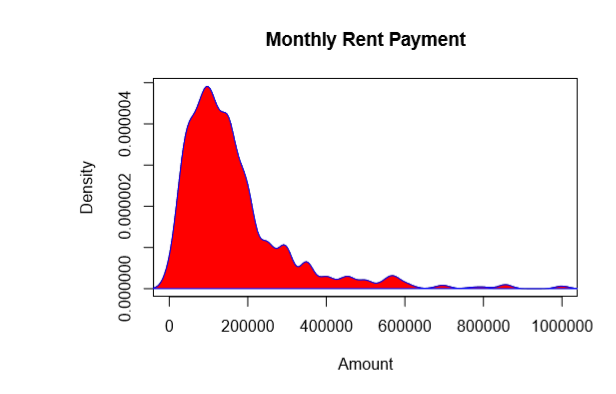
**Plot 1**

Plot 12 describes proportion of vulnerable and non-vulnerable households based on number of years in schooling. 

**Plot 12**

1. **Data Preprocessing**

To understand the dataset better we reverse encoded all the one hot encoded feature and created additional columns. We also dealt with the missing data by digging deeper into them. Monthly rent had a lot of missing data. Plot 13 describes the distribution of monthly rent.



**Plot 13**

One assumption was that the data missing is for those who own the house so technically they pay no rent. This was proved true on exploring the attribute tipovivi which describes the ownership status of household. Therefore, we replaced missing values with 0. Similar findings were observed in case of other attributes with missing values and 0 was used to replace missing values in v18q1(tablet) and rez\_esc(years behind school) whereas median of mean education was used to replace missing values in mean education. To prioritize identifying the vulnerable households we converted the multinomial Target variable into a binomial one with all vulnerable household holding the value 1 and non-vulnerable holding 0. Some of the attributes had a lot of categories. Random forest algorithm cannot take more than 53 categories; Therefore, we discretized some attributes such as mean education and monthly rent while excluding some others that had no considerable impact on learning about the variance in data such as Id and Idhogar. Idhogar is an attribute that describes the household. This variable can be important in the future when we aggregate data and do analysis on household level, but for now we considered removing the attribute since our research is focused on individual level.

1. **Data Selection**

After adding reverse hot encoded features our dataset now had 156 features. Large number of features can overwhelm a machine learning algorithm and it could take forever to learn and execute. It was important to select only those features that contribute towards predicting the poverty level. We implemented correlation and Principle Component Analysis to understand features which contribute towards predicting the target variable and selected only those when training our models. For naïve bayes we could easily add all the features to training the model, however for random forest and logistic regression we added only the top selected features. Table 1 below shows the selected features.

**Table 1: Selected Features**

|  |  |  |
| --- | --- | --- |
| overcrowding | r4m1 | r4h1 |
| pisocemento | epared1 | hacdor |
| eviv2 | epared2 | r4m3 |
| instlevel2 | tamviv | instlevel1 |
| hogar\_adul | bedrooms | lugar1 |
| qmobilephone | instlevel8 | rooms |
| eviv1 | hogar\_nin | r4t1 |
| etecho1 | energcocinar4 | pisomoscer |
| paredmad | meaneduc | escolari |
| tamhog | cielorazo | computer |
| epared3 | v18q | eviv3 |
| v2a1 | paredblolad | etecho3 |

1. **Modeling**

Holdout test was selected as the method to split the data into training and testing. The holdout method is a very simple version of cross validation. Data is randomly separated into a training and testing set. The models are only trained on the training set, and then used to predict appropriate class for the test set. These predictions are then compared to actual test set labels to evaluate the model. The dataset has 9557 rows which was split into 6691 rows for training(70%) and 2866 rows for prediction (30%). Before executing any machine learning algorithm a baseline was set so that any accuracy above the baseline will be considered as good. In the test data we have 1798 individuals living in non-vulnerable household and 1068 individuals living in vulnerable household. If the model predicts everything as non-vulnerable then the accuracy of the model is 62%. Any prediction accuracy beyond 62% will be considered a good accuracy.

**Naïve Bayes:**

Naïve Bayes algorithm from package “e1071” was used on the training dataset.

It is a simple probabilistic classifier that relies on the assumption of independence among the features. We trained two classifiers using this algorithm on the training set; one with all the features and the other with top features selected by us as a result of correlation and PCA. We set the laplace value to 1 and then applied these models on the test set to check for performance. The model outputs a raw accuracy scores of 75.44% and 74.6% respectively. The AUC curve of the best model is shown in Plot 2.

A screenshot of a cell phone

Description automatically generated

**Plot 2**

**Logistic Regression:**

Logistic Regression was performed using the function glm. Logistic regression is a machine learning algorithm that works well for binary classification problems. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. The top features were used to train the model. We achieved 77.84% accuracy using this model. The AUC curve is shown in Plot 3A screenshot of a cell phone

Description automatically generated

**Plot 3**

**Random Forest:**

Random forest was implemented using the package “randomForest”. Random forest is an ensemble learning method that operates on the premise that combination of learning models increases the overall accuracy results. The top features were used to train the model and the number of ntrees was set to 1000 because it gave slightly better accuracy than setting the number of trees as 500. Plot 4 gives the top 10 important features as ranked by random forest algorithm.A screenshot of a social media post

Description automatically generated

**Plot 4**

Random forest gave us an accuracy of 93%. According variable importance of Random forest, numbers of years spent in school is one of the strongest signals to predict individual poverty in Costa Rica, followed by overcrowding and number of mobile phones. The AUC curve of Random forest is shown in Plot 5.

A screenshot of a cell phone

Description automatically generated

**Plot 5**

1. **Results:**

Random forest gave us the best results compared to the other two machine learning algorithm. The confusion matrix for the models is shown below to get a better understanding of number of misclassified data for each model.

|  |  |  |
| --- | --- | --- |
| **Naïve Bayes** | | |
| **Accuracy: 75.4%** | **Non-vulnerable** | **Vulnerable** |
| **Non-vulnerable** | 1407 | 313 |
| **Vulnerable** | 391 | 755 |

|  |  |  |
| --- | --- | --- |
| **Naïve Bayes (Selected Features)** | | |
| **Accuracy: 74.6%** | **Non-vulnerable** | **Vulnerable** |
| **Non-vulnerable** | 1430 | 360 |
| **Vulnerable** | 368 | 708 |

|  |  |  |
| --- | --- | --- |
| **Logistic Regression (Selected Features)** | | |
| **Accuracy: 77.8%** | **Non-vulnerable** | **Vulnerable** |
| **Non-vulnerable** | 1548 | 385 |
| **Vulnerable** | 250 | 683 |

|  |  |  |
| --- | --- | --- |
| **Random Forest (Selected Features)** | | |
| **Accuracy: 93.3%** | **Non-vulnerable** | **Vulnerable** |
| **Non-vulnerable** | 1743 | 137 |
| **Vulnerable** | 55 | 931 |

1. **Conclusion**

To summarize, we can say that the Random forest works better for this task than the naive Bayes classifier or logistic regression.

The analysis was done on an individual scale, it will be interesting to do a similar analysis on household scale by using the hogar id to aggregate the data on household level. There was also a slight data imbalance and using techniques such as SMOTE, down sampling and up sampling may improve model performance even more.

**References**

Kaggle.com. (2018). *Costa Rican Household Poverty Level Prediction | Kaggle*. [online] Available at: https://www.kaggle.com/c/costa-rican-household-poverty-prediction/data [Accessed 10 Dec. 2018].

Kaggle.com. (2018). Predicting Poverty Levels with R | Kaggle. [online] Available at:https://www.kaggle.com/taindow/predicting-poverty-levels-with-r [Accessed 10 Dec. 2018].

1. https://www.kaggle.com/c/costa-rican-household-poverty-prediction [↑](#footnote-ref-1)