ST309 Group Project Report

Building Models to Estimate Poverty Levels

[redacted]%), [redacted]%), [redacted]%)

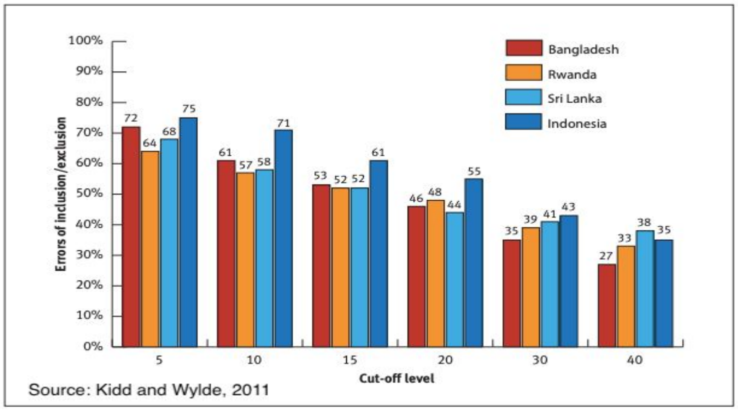
February 14, 2019

I. Introduction

In many developing parts of the world it is difficult for aid organizations and government policymakers to determine what subsets of the population are in greatest need of financial assistance, as many impoverished people work in the informal job sector and records of income in these areas are generally poorly kept. Additionally, the household is the unit most commonly targeted for financial assistance, as individual treatments are often distributed amongst family members anyways. This complicates the identification process even further, in the sense that households are internally diverse and thus sometimes that noise masks common signals of poverty.

Due to these constraints, a Proxy Means Test (PMT) has become a very popular method used to estimate a household's level of financial need. The PMT is, in the most technical sense, a multiple regression formula that is employed practically to produce a "score" that is an estimate representative of a given household's level of wealth, based a variety of features or "proxies". According to an International Labor Organization assessment of PMT methodology in *Kidd, Gelders, and Bailey-Athias* (2017, p.2), these "proxies" are "usually based on: demographics (such as age, gender, and number of people in the household); human capital (such as level of education of the household head); type of housing (such as the type of roof, walls, floor and toilet); durable goods (such as whether a household has a radio, refrigerator or television); and productive assets (such as whether a household owns animals or land)."

Quite often, however, these models are built on faulty data assumptions and struggle with overfitting, resulting in high levels of error when applied in practice. For example, *Alatas et al* (2016) found that the PMT model used to facilitate the *Program Keluarga Harapan* (PKH) conditional cash transfer scheme in Indonesia resulted in 93 percent of the poorest 5 percent of households being excluded. Another study of the *Oportunidades* (formerly *Progresa*) program in Mexico found that a PMT selection process meant to target the poorest 20 percent of the population had inclusion or false-positive errors of 36% and exclusion or false-negative errors of 70% (*Veras et al,* 2007). As seen in the figure on the next page, these error levels tend to increase with more specific coverage levels or to the the extent that they aim to target smaller, poorer, subsets of the general population (*Kidd and Wylde*, 2011). These shortcomings have led to pushback against the PMT model by academics and the development community at large, as critics claim that such inaccurate dispersal of funds is essentially arbitrary and thus could lead to negative stigmatization of aid recipients within their communities as well as mistrust of aid organizations in the future.



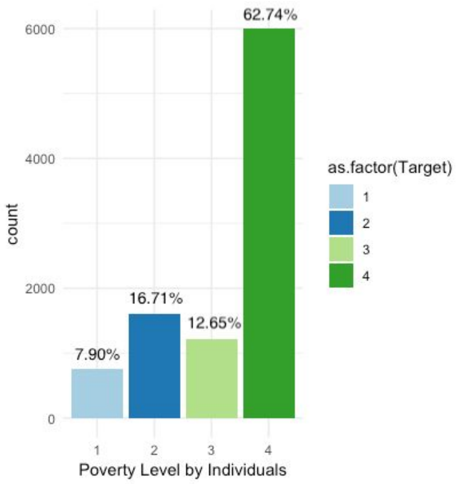
Source: Kidd and Wylde, 2011

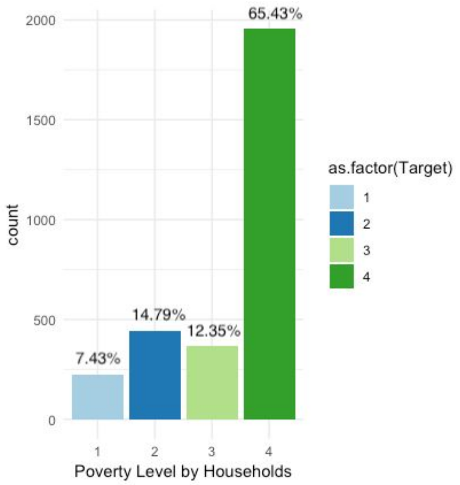
It is clear, however, that the informational scarcity that has necessitated the use of the PMT model in the first place is not going to be alleviated any time soon. Furthermore, while these criticisms of PMT methodology are valid, little effort or research has been committed towards creating a viable alternative. Thus, we believe that the general idea behind the Proxy Means Test model is worth salvaging and we aim to create a more sophisticated and replicable PMT model through the use of rigorous statistical and machine learning techniques that will provide more accurate results when used on testing data, going beyond the standard econometrics approach. Specifically, through careful feature selection and the development of models for different coverage levels, we aim to address the two large concerns of academic assessments of PMT, arbitrariness in selection and retention of accuracy when identifying the poorest subsets of the general population. Our hope is that our findings will offer improvement for the current selection process for means-based aid and thus enable NGOs and governments to better assist those in need.

II. Data Observations

Our data is sourced from the Inter-American Development Bank's "Costa Rican Household Poverty Level Prediction" (https://www.kaggle.com/c/costa-rican-household-poverty-prediction) Kaggle challenge. The dataset contains 9,557 of each member in 2,988 unique households and 141 features associated with each individual. Each individual matches to a unique household ID (i.e., each individual belongs to one and only one household.). The remaining 140 features include specific details about rent payment, overall and detailed physical settings and characteristics of the house, household size, gender distribution, geographic region, age, marital status, and education level of each individual in the household. The majority of features are dummy variables. The dataset also includes 12 engineered features, including dependency rate, overcrowdedness, average years of education for adults squared, number of children in the household squared, etc.

The outcome of interest is **Target**, an ordinal variable indicating groups of income levels, of which 755 observations are in extreme poverty (1), 1597 in moderate poverty (2), 1209 in vulnerable households (3), and 5996 in non vulnerable households (4). As shown in the figures below, the data is unbalanced: the extreme poverty class accounts for ~7% of the observations and the non vulnerable class accounts for ~60%. Hence the information on the most needy households is overwhelmed by that of less vulnerable ones. If we use the whole dataset for multiclass classification, such imbalance will result in a fitted model predominantly led by the information on level 4s because the signal on others might be too weak to be picked up. Our solution is elaborated further in section III.





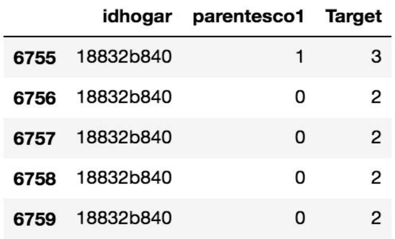
Besides household-level characteristics, we are given individual-level attributes in the dataset, knowing that each individual belongs to one and only one household. According to the Kaggle Challenge, we are required to make predictions at the household level. Therefore, we look at each attribute across individuals grouped by household and summarize statistics (mean, median, ratio, etc.) to determine features of households. We want to create a condensed dataset which only has household attributes. To do that, we need to clean the original dataset.

i) Data Cleaning

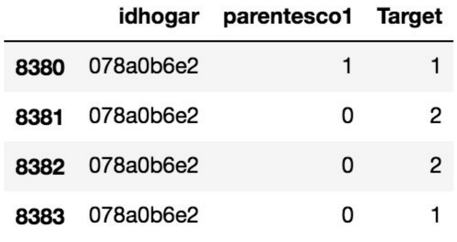
First, there are missing values in columns monthly rent payments (v2a1), number of tablets household owns (v18q1), years behind in school (rez_esc), average years of education for adults (meaneduc), and squared average years of education for adults (SQBmeaned). We will address the missing value issues one by one. First, we have 6,860 missing values in monthly rent payments, but of all the missing data 5911 individuals live in a fully paid house. We then assign all of these missing values to 0 to indicate no monthly rent payments and leave the rest as NA. We also generate a boolean variable **v2a1.missing** for the left out missing data in case they indicate some importance for our prediction model later. Similarly, we realize that the number of missing values in **v18q1** is the same as the number of households which does not own a tablet. Therefore, all the NA's in **v18q1** mean 0 count of tablets. In addition, **rez_esc** has 7,928 missing values. After we plot relationship between age and years behind in school, we realize that this variable is only defined for individuals between 7 and 19. Anyone younger or older than this range presumably has no years behind and therefore the value should be set to 0. For anyone else, we once again leave it as NA and add a boolean variable **rez_esc.missing**. Lastly, the 5 missing values in **meaneduc** and **SQBmeaned** exist because there are no adults in the household.

Second, we notice that there are a mix of characters and numbers in engineered features **dependency**, **edjefe**, and **edjefa**. The documentation indicates a "yes" equals to 1 and a "no" equals to 0, so we change the texts into factors 0 and 1, and convert them into numerical values.

Third, we need to address the wrong target labels. We notice that individuals in the same households do not have the same target labels, in which case we are told to use the poverty level of the head of household as the common agreed target labels. Some of the mis-labeled targets across individuals within a household are exemplified below. We have correct all the labels as instructed accordingly[1].



	idhogar	parentesco1	Target
6755	18832b840	1	3
6756	18832b840	0	2
6757	18832b840	0	2
6758	18832b840	0	2
6759	18832b840	0	2



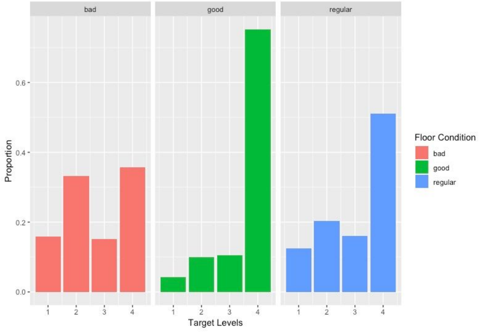
	idhogar	parentesco1	Target
8380	078a0b6e2	1	1
8381	078a0b6e2	0	2
8382	078a0b6e2	0	2
8383	078a0b6e2	0	1

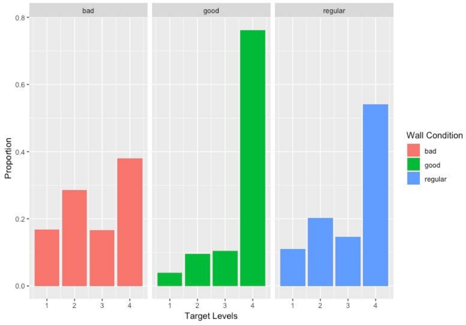
Finally, after deleting some repetitive columns, we come to the stage of feature engineering - reducing the number of dummy variables and transforming individual-level characteristics into household-level characteristics.

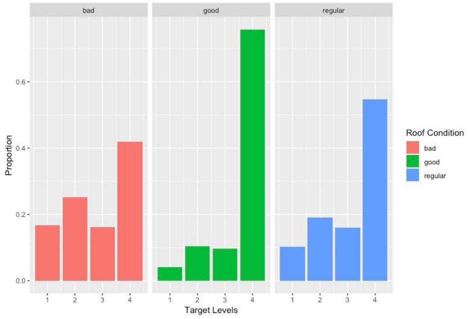
ii) Feature Engineering

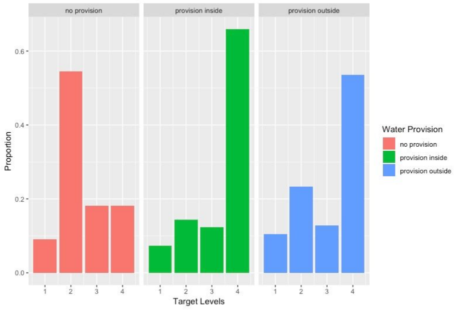
- Household- Level Categorical Variables

First, we look at household-level categorical variables. Since we have many dummy features, plotting them against the target levels help us determine which variables might be relevant for prediction.





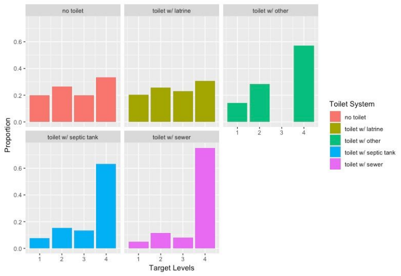


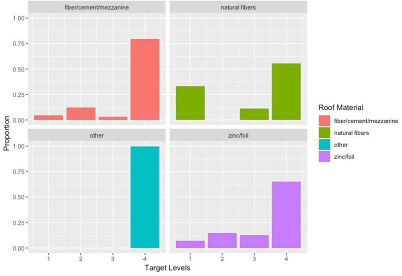


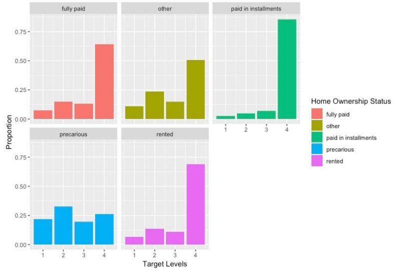
[1] We have used reference from one of the kernels in Kaggle to address the wrong labeling issue. See the reference page for more detail.

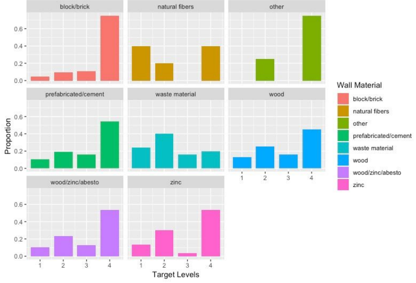
Floor, roof, and wall conditions are labeled as dummies to indicate bad, good, and regular conditions in the original dataset. Water provision also has three dummies indicating no provision, provision inside, and provision outside. The proportion of each poverty level changes dramatically when plotted against the three conditions. They nonetheless all suggest an ordinal trend so we are able to rank those conditions as 0, 1, 2. As a result, we change the 12 dummies indicating wall, roof, floor, and water provision qualities into numerical variables and label them wall, roof, floor, and waterprovision, respectively.

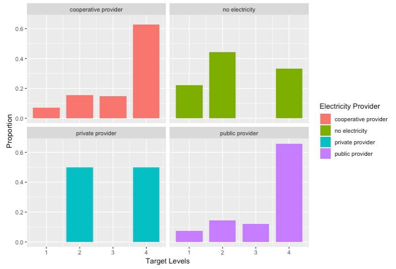
However, other characteristics such as floor materials, roof materials, wall materials, electricity providers, toilet systems, cooking energy sources, homeownership status, regions, water provisions, rubbish disposal methods, do not provide us with enough evidence to create ordinal rankings. There may be some clear comparisons among some of the subcategories in a feature, but unlike the clear distribution change in all of the subcategories of the features plotted above, we cannot tell which condition in each feature listed below is superior than another. We also have the "other" and "no" subcategories in our these features: it is difficult to justify which should be considered better than the other based on the plots alone. Therefore, we leave them as dummies to avoid subjective bias and allow the predictive models to decide whether any of them is important. However, to address the importance of the lack of basic living facilities, we generate a **warning** indicator: we add one score to the variable if the household has no toilet, no electricity, no floor, no water provision, no ceiling, or no bathroom.

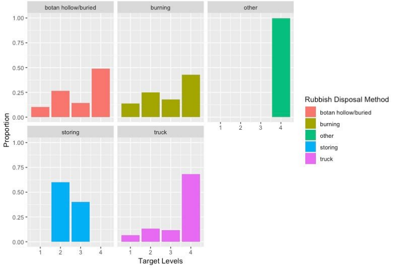


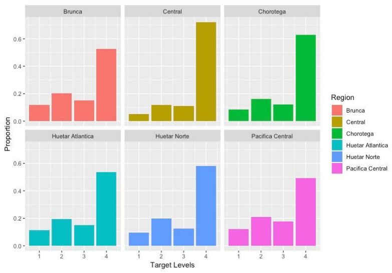


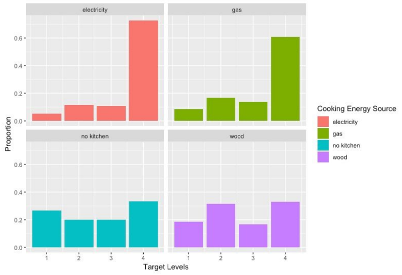


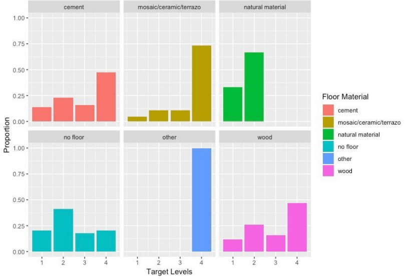




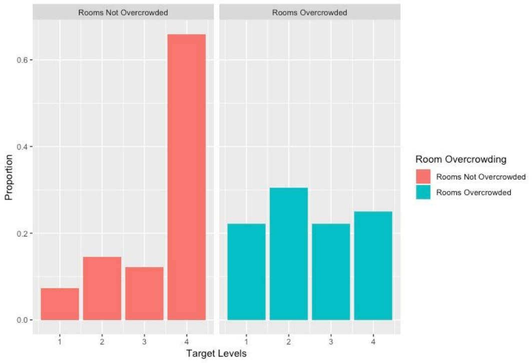


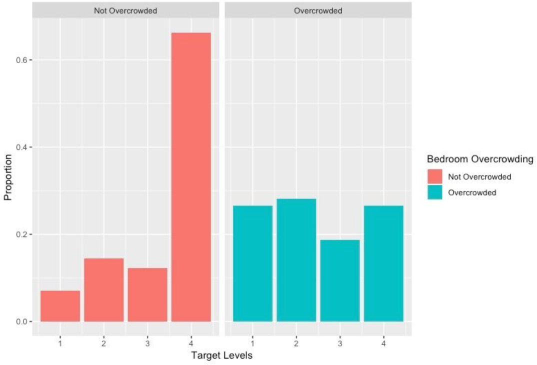


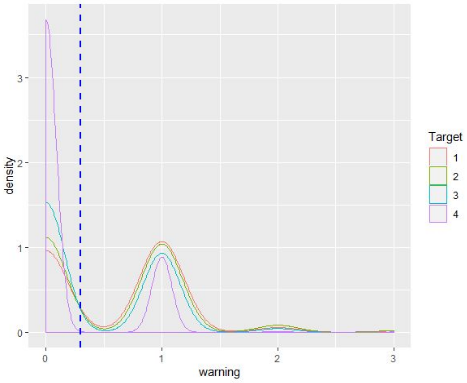


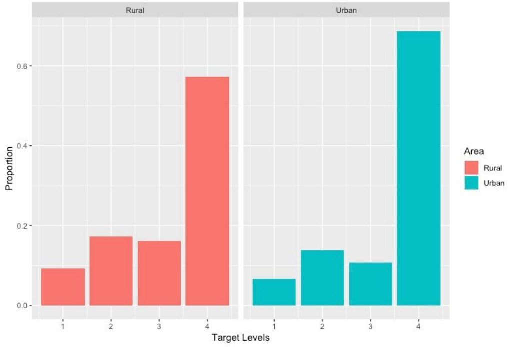


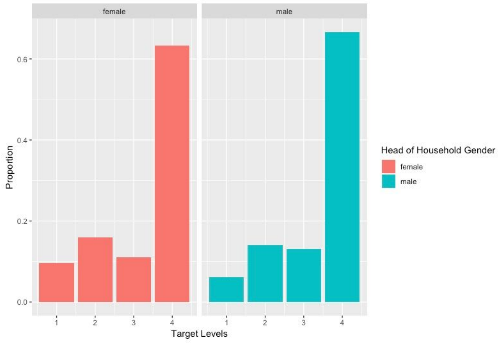
Apart from the features with multiple categories, we also observe real dummy variables which only have two values for each described feature. For example, **hacdor** (overcrowding by bedrooms) and **hacapo** (overcrowding by rooms) might be strong indicators for separating vulnerable households from the rest, but not as strong for differentiating other three levels. We observe significant lower proportion of level 4's in overcrowded households. Additionally, rural versus urban areas also show a small difference in the proportion of vulnerable households each area has, but also not so much difference for other three poverty levels. In the end, we exclude **area2** (a dummy for rural) from the dataset to avoid its perfect collinearity with **area1** (a dummy for urban). Lastly, whether the head of household is male or female shows a slight significance, suggesting female heads lead to poorer household conditions.



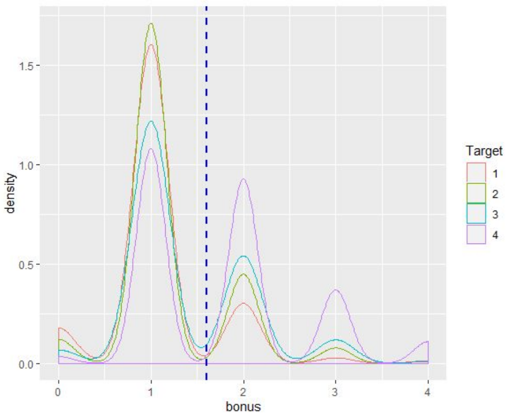






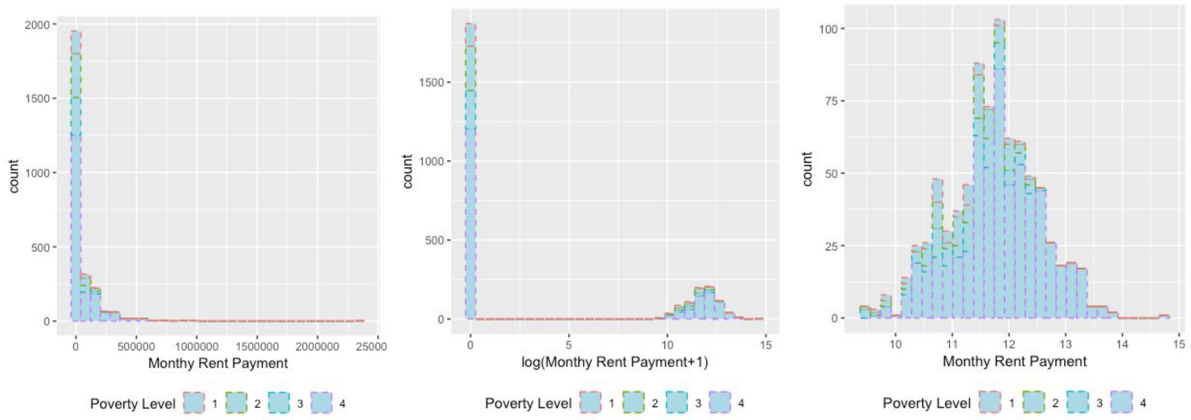


Moreover, to address the importance of the existence of electric equipments and other types of technologies, we generate a bonus indicator: we add one score to the **bonus** variable if the household owns a refrigerator, a computer, a tablet, a television, or a mobile phone.

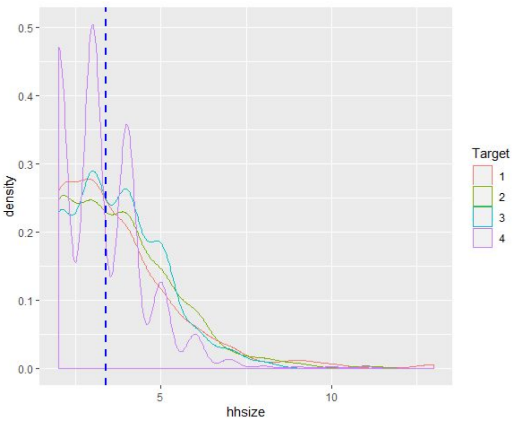


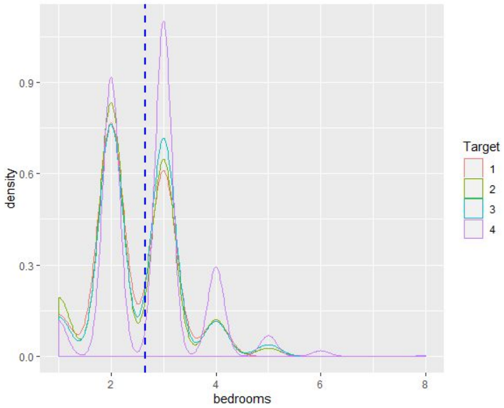
- Household-Level Continuous Variables

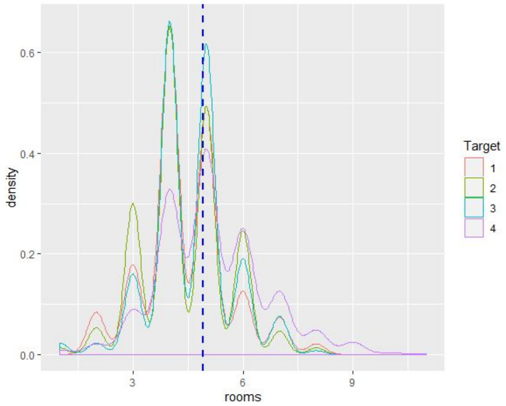
Second, we look at household-level continuous variables. The distribution of monthly rent payment is severely skewed to the right, with most of the the payments around 0. However, after log-transformation (we also add one to the variable before because we have 1871 zeros in the dataset), the distribution seems to be normal only at very high payments range. This is due to the fact of high exchange rate of colóns. Ignoring the zero rents, we do see a more normally distributed monthly payment plot.



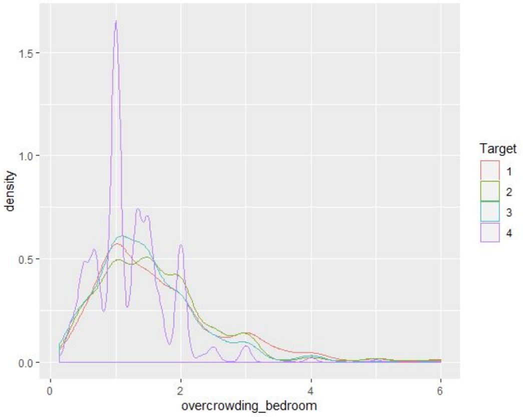
The first density plot below shows that more vulnerable households tend to have smaller family sizes. This feature might be important from a policy perspective on debating whether the government should limit the number of children each household can have to increase standard of living. However, the number of rooms and the number of bedrooms each house has do not seem to be important factors.

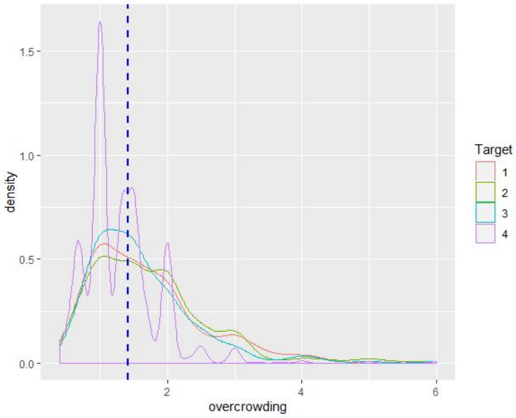




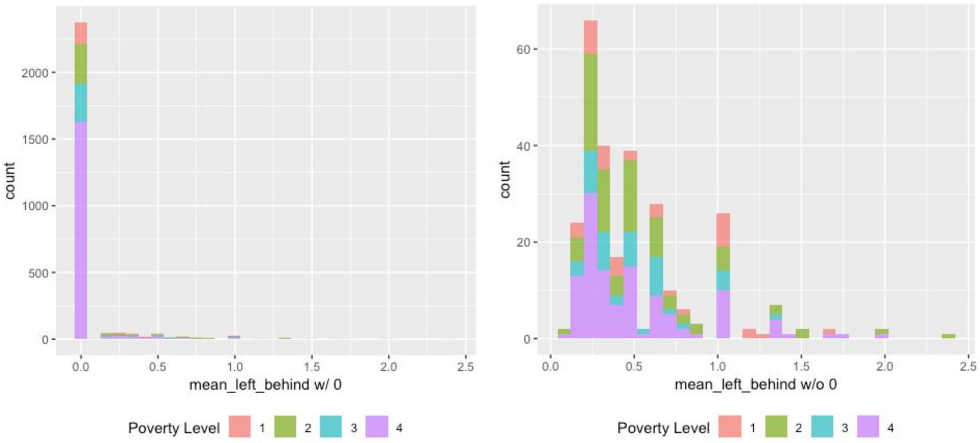


As a result, we divide household size by number of rooms and bedrooms, we generate **overcrowding** and **overcrowding_bedroom** features - the larger the number of persons per room/bedroom has, the more crowded it will be. We see poorer households have more crowded living conditions.



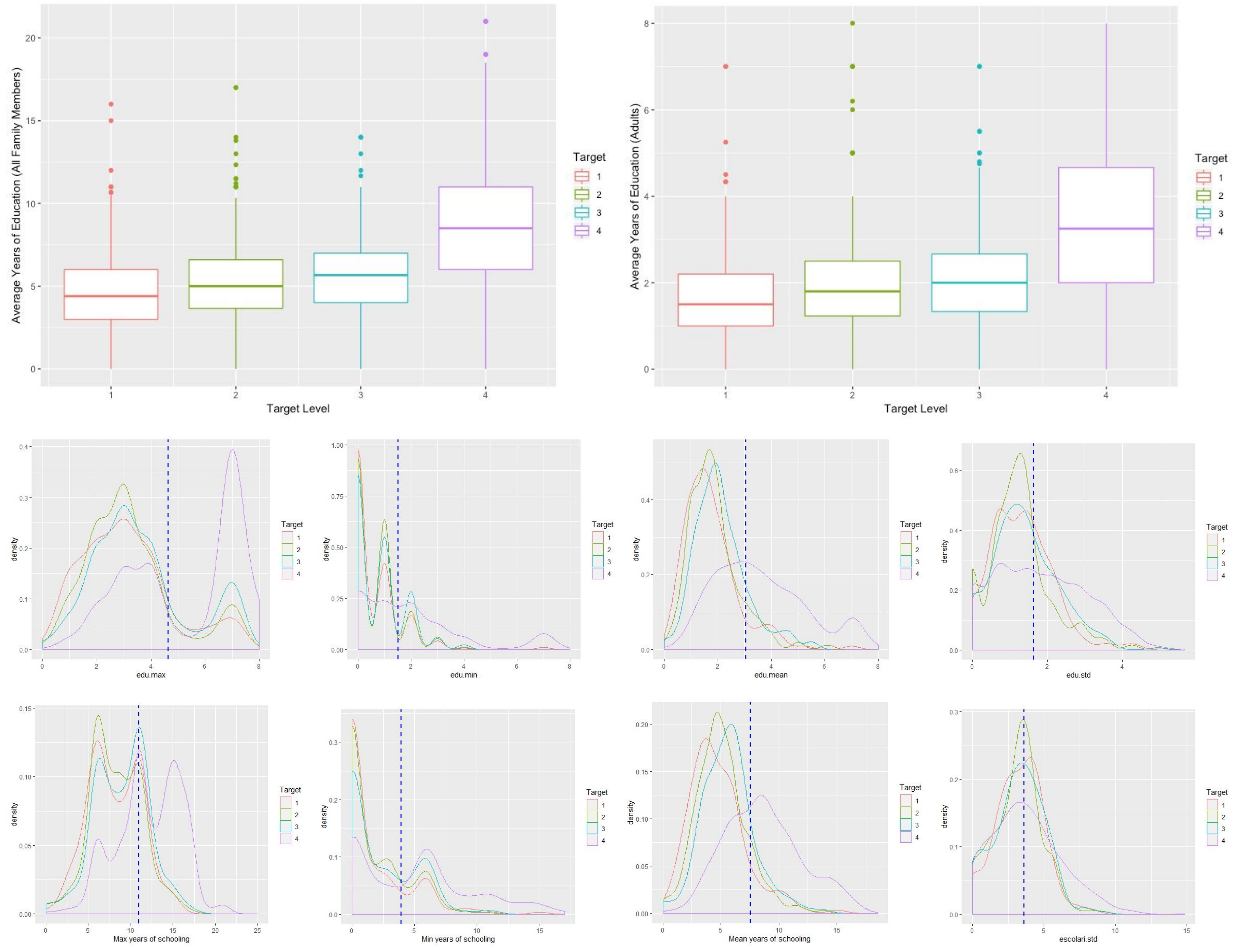


Years behind school, **rez_esc**, might also be a strong indication of what the poverty level of households are. Most of the well-off households' children have an average years behind school of zero. This can be seen from the bar plot on the left below: most proportion of zero comes from level-4s' households, with level 3, 2, and 1s' following. However, if we filter out all the data with years behind school equal to zero, we can see that level ones do not occupy a large proportion of each bin in the variable any more; poorer households instead take up more.

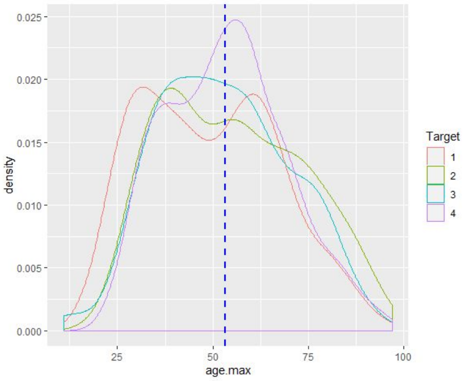


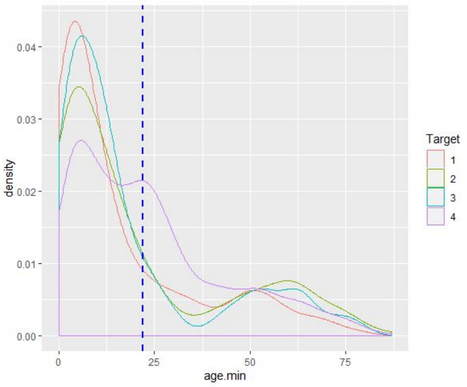
- Individual-Level Variables

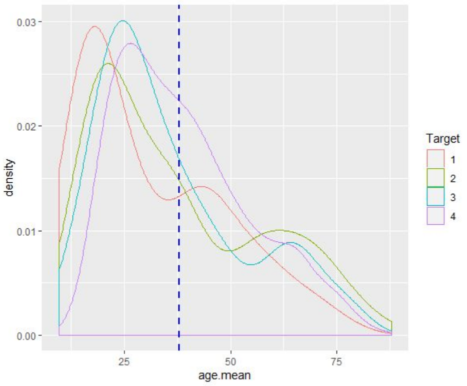
Third, we look at individual-level attribute: education, age, gender, and disability. There are multiple variables in the dataset that contain information about education. We suspect that education level is one of the most deterministic factors for predicting poverty level. As shown in the box plots, we can see a clear positive relationship between average years of education and poverty level, so we decide to reverse the 9 **instlevel** dummies indicating levels of education received (1-9) and summarize education statistics including maximum, minimum, mean, and standard deviation across households. We generate four aggregate level features, **edu.max**, **edu.min**, **edu.mean**, and **edu.std**. Similarly, we summarize the four statistics for **escloari**, years of schooling, across each household. The density plots for each are also shown below. We can see that education levels among vulnerable, non-vulnerable, moderately poor, and extremely poor households differ significantly. Standard deviation of education obtained within richer households tends to be smaller.

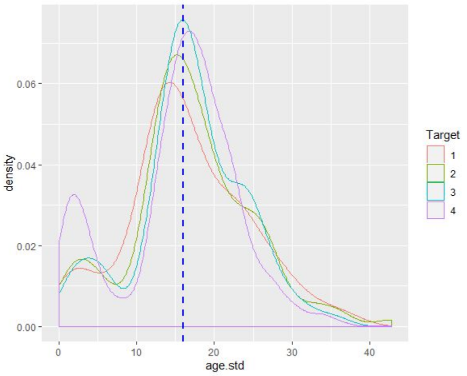


We also create features **age.max**, **age.min**, **age.mean**, and **age.std** like what we do for education levels. As we can see, richer households in general have more adults (19-25) but poorer households have more children (<19) and elders (>65). This can be seen from the maximum and minimum age distributions across the four household levels. The mean and the standard deviation of age in each household do not seem to play an important role.

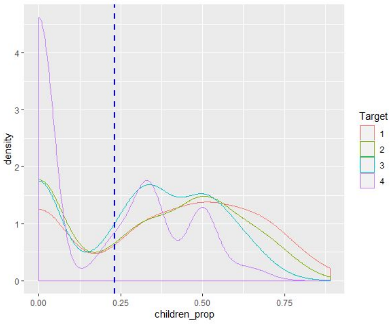


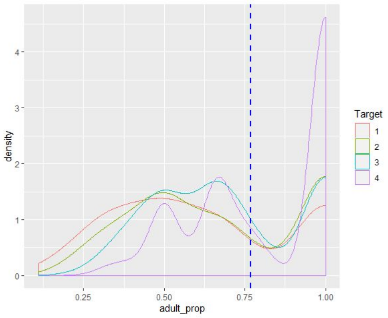


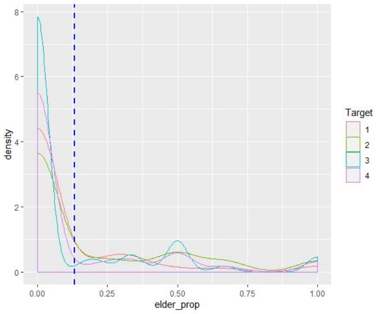


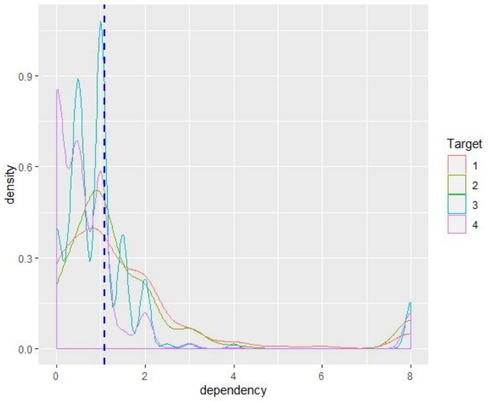


Therefore, we generate household-level features related to age by including proportion of children (<19), adults, and elders (>65). Fortunately, we also have an engineered feature in the original dataset named **dependency**, which is the ratios of the number of members of the household younger than 19 or older than 64 and the number of member of household between 19 and 64. This might be a very useful feature since the non-vulnerable households apparently are more independent than the poor ones.

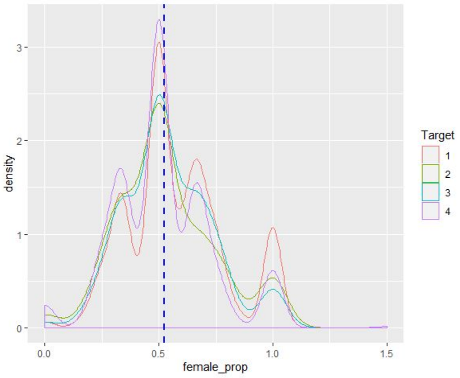


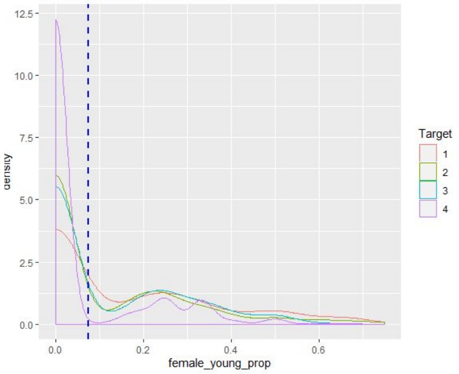


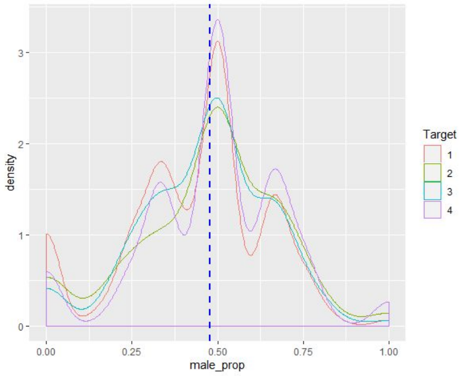


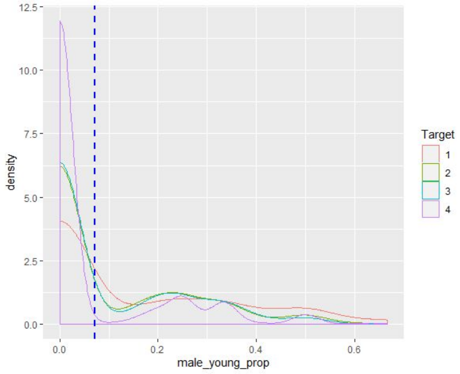


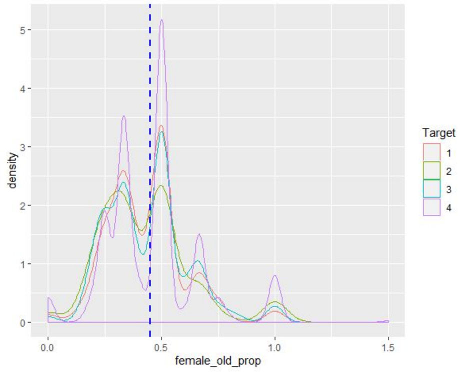
Since we mentioned previously that the gender of a household's head seems to be related to poverty level differences, we also combine age characteristics with gender proportions when generating new features. The engineered features include proportions of female, male, young female, young male, old female, and old male in each household.

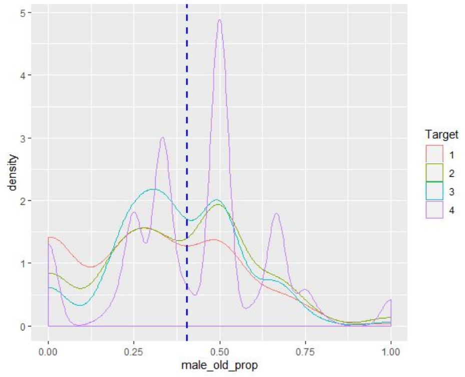




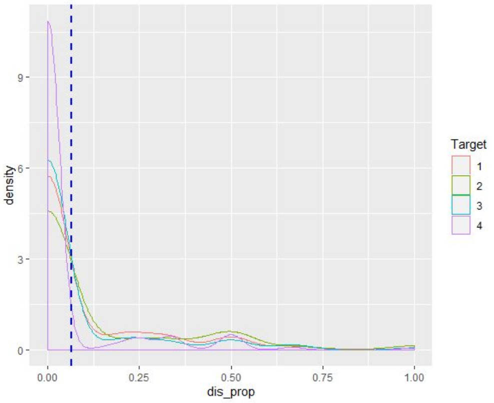






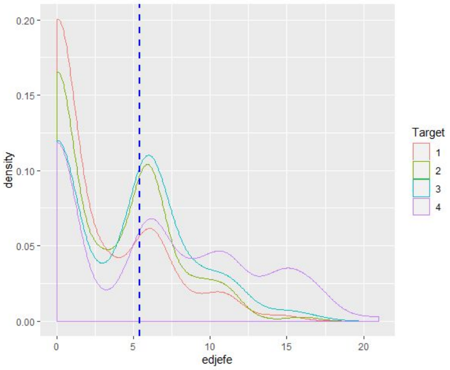


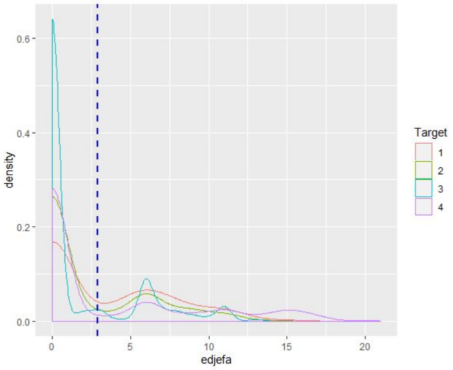
Lastly, we plot the proportion of disabled persons in the household. Not surprisingly, households in severe poverty have larger proportion of disabled members than others.

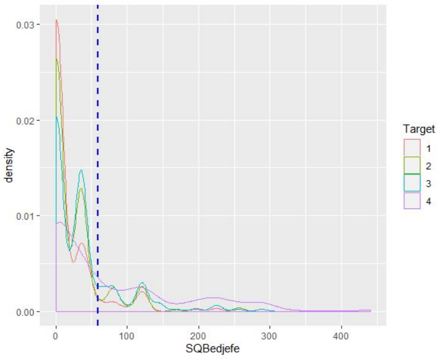


- Interaction Variables

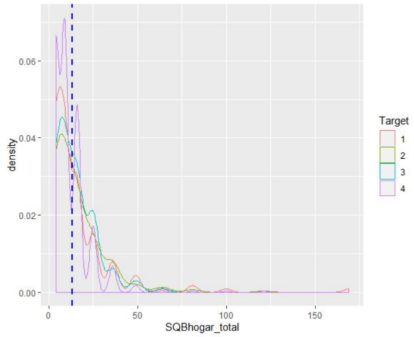
Fourth, the original dataset also contains interaction variables **edjefe** (years of education of male head of household, based on the interaction of years of education) and **edjefa (**years of education of female head of household, based on the interaction of years of education). We find that most of poorer households have female heads and education is positively correlated with income levels regardless of the gender of each household head. This confirms our findings above, which suggest that gender and education issues lead to income level differences across households. **SQBedjefe**, squared **edjefe**, shows a more obvious distinction, whereas **edjefa** is less obvious since fewer well-off households have female heads and the differences among levels 1, 2, and 3 are almost indistinguishable.

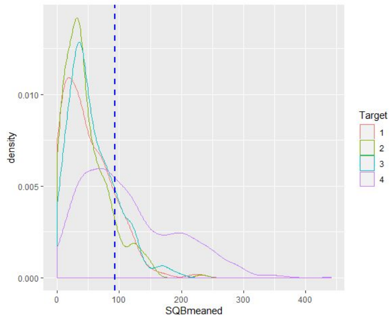


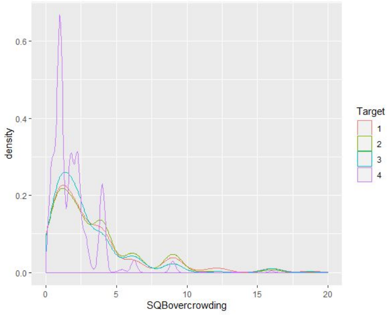




Additional engineered features involve **SQBhogar_total** (squared household size), **SQBmeaned** (squared average years of education) and **SQBovercrowding** (squared number of persons per room). All of them seem to be very predictive features so we leave them in the dataset.



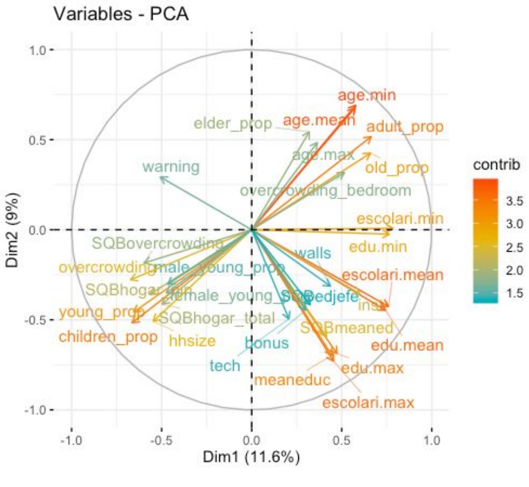




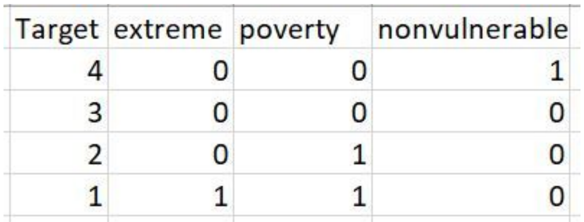
In the end, we get rid of all the individual-level attributes and condense the large dataset with 9,577 individual observations and 149 features to 2,988 unique household observations and 104 features. We now proceed with our next step of data analysis - features selection and models building.

III. Data Analysis and Results

To gain a better insight into the variables, we apply principal component analysis and construct a plot for the first two principal variables. This unsupervised learning technique allows us to look at the variation within each feature after standardizing all variables. The ones with most variation provide a guide for us to reduce the number of features in our dataset. The color-coded arrows tell us the features that we want to pay more attention to. Fortunately, most of our engineered features appear on the plot and therefore suggest certain level of significance. We can then move on to our model training stage with the cleaned datasets.



First, we divide the training data into two parts – one group for training and the other for testing. To preserve the overall class distribution of our Costa Rican population, we split the data in a way that random sampling occurs only within each income level class. Second, we use cross validation methods to train our models with the training set and select the model that yields the lowest validation error rate. Finally, we run the selected model on testing data and record the test error rate. To address the imbalance issue, we first merge levels 1, 2, and 3 and build a binary classification model on a more balanced dataset to find out the non-vulnerable households (level 4). Using our predicted results, we remove the non-vulnerable households to form a new subset. Then we merge the levels 1 and 2 of the subset and build the second classifier to find out the vulnerable households (level 3). Finally, whether to use a subset of data to train our third model to find out the extremely poor households (level 1), we have different considerations when implementing logistic regression and trees algorithms. The reasoning behind this methodology is the assumption that poor households of different categories, moderate and extreme poverty for example, are more similar to each other than they are to less vulnerable households. By removing those less vulnerable households from the training data, we are able to train models that key in on the particular factors that set levels of poverty apart.



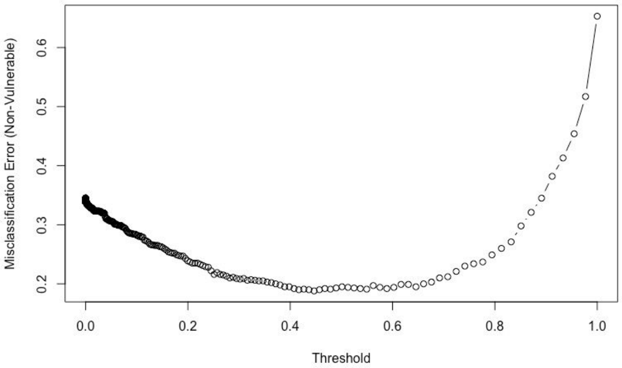
Target	extreme	poverty	nonvulnerable
4	0	0	1
3	0	0	0
2	0	1	0
1	1	1	0

[2]

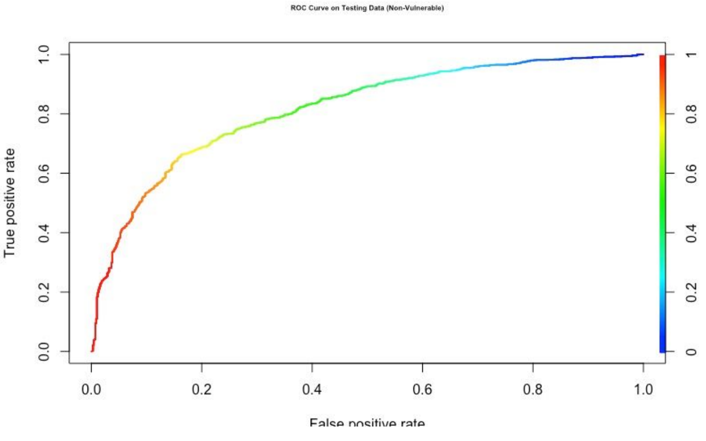
i) Stepwise Logistic Regression

The first classification method we attempted was logistic regression, trained under the stepwise iterative feature selection process. Logistic models regress the likelihood of a binary outcome on a variety of features and through a logistic transformation restrict the predicted outcomes to values between zero and one. Putting our logistic regression model through the stepwise feature selection process optimized our model to only include features that would yield the lowest prediction error when applied to testing data by omitting features that would lead it to overfit the training data. This is done by iteratively adding or removing predictors to the model, seeking the set of features that results in the lowest possible Akaike Information Criterion (AIC) score. Furthermore, the stepwise process can be run in either a “forward” (starting with no features), “backward” (starting with all features) or “both” direction. Due to the fact that our dataset contains a large number of variables (107), many of which are dummy variables, we chose to run the stepwise process in the “backwards” direction so that the model considers all of the features and that a variable added at the beginning of the process does not become redundant later after other features have been added. Finally, after the generation of each model, we selected a probability threshold through cross-validation that yields the highest prediction accuracy.

Model 1: Training Accuracy Rate 81.2%, Testing Accuracy Rate: 75.4%, False-Negative Rate: 31.5%, False Positive Rate: 22.1%

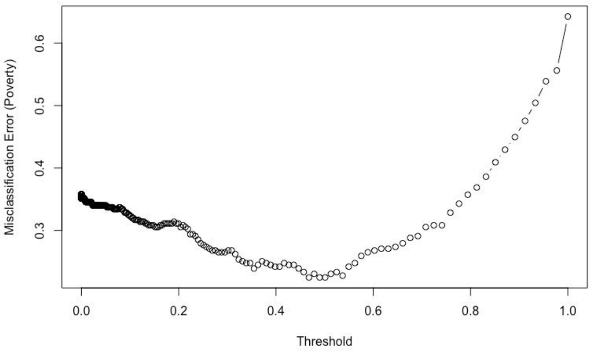


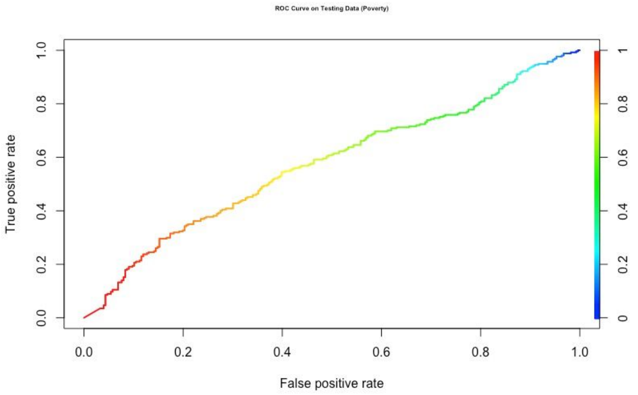
ge
els
thi



The first model, separating non-vulnerable (level 4) households from vulnerable (levels 1,2,3) households, was unsurprisingly the most accurate. Of the 89 variables that were selected through the stepwise process, 18 were significant at the 10% level or better. The nature of these features were highly diverse from variables such as number of tablets and mobile phones owned by the household to whether the household had a private electrical provider to the proportion of the household that was children or female to the quality of the home's roof, walls, and floor. The above figures show the probability threshold used in the model (44.67%) and the ROC curve which indicates the tradeoff between true-positive accuracy and false-negative error for the model.

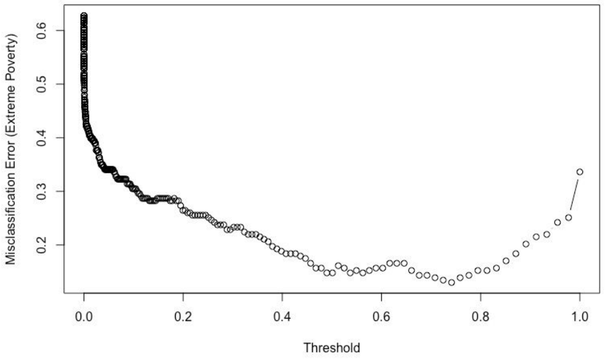
Model 2: Training Accuracy Rate: 77.5%, Subset Testing Accuracy Rate: 49.2%, Subset False-Negative Rate: 47.9%, Subset False-Positive Rate: 51.7%, General Testing Accuracy Rate: 52.8%

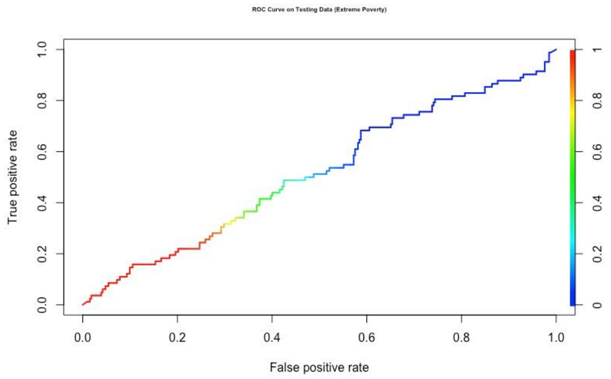




The second model showed a large decrease in prediction accuracy when used on the testing data, signalling that separating vulnerable (level 3) households from poor (level 1,2) households is relatively more difficult. Of the 82 variables selected for the logistic regression, 19 were significant at the 10% level or better. The majority of these variables were related to the physical properties of the house, while variables related to gender distribution, number of children, and education were also important. The cutoff threshold used in this model was 46.77%, similar to the threshold in model one. Due to large disparity between the training accuracy rate and testing accuracy rate and the very high number of features, it is likely that this model was adversely impacted by overfitting. This is also evidenced by the ROC curve.

Model 3: Training Accuracy Rate: 87%, Subset Testing Accuracy Rate: 62.4%, Subset False-Negative Rate: 19.7%, Subset False-Positive Rate: 80%, General Testing Accuracy Rate: 67.5%





The final model shows a decent improvement in accuracy but an alarming increase in the false-positive rate. 74 variables were chosen for this model, of which 32 were significant. The important features were all related to the composition of the house, education, and gender distribution. The threshold used in the third model was 74.1%. Despite that high cutoff, false-positive rates were still extremely high, also as evidenced by the nearly linear ROC curve.

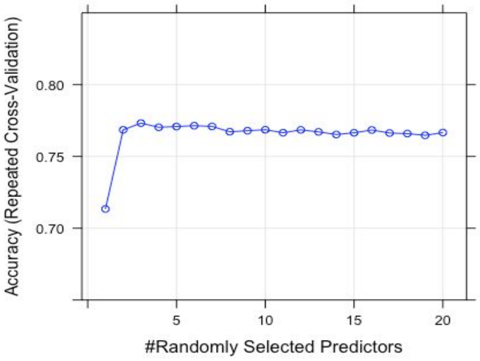
In all three of the models variables relating to gender and age (**female_prop, female_young_prop, female_old_prop**) and education (**mean_educ, edjefe**) were significant. As the models sought to distinguish between lower levels of poverty, features regarding the household's physical properties such as roof material, floor material, wall material, etc. became more important. Lastly, with each successive model the false-positive rate increased. Generally speaking, the results of this model are encouraging to the extent that accuracy levels were relatively high, but that revelation comes with the caveat that the selected models contained far too many variables, thus feeding the tendency to overfit and generate false classifications. Being regression-based, the logistic model was also susceptible to data weaknesses such as multicollinearity. Due to the fact that there were so many dummy variables as features, we believe this likely impacted our predictions negatively. Our other models are not regression-based and thus should prove to make an interesting comparison on the basis of accuracy.

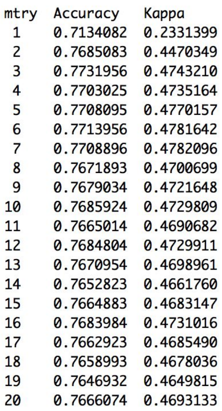
ii) Decision Trees

Decision trees make no assumptions on relationships between features. It constructs splits on single features that improve classification, based on an impurity measure like Gini or entropy. Therefore, this mechanism is by nature immune to multicollinearity. However, a single decision tree is very vulnerable to overfitting, so we must either limit depth, prune heavily or average many using an ensemble. Such problems get worse with many features, but we can use random forests and boosting trees to rescue.

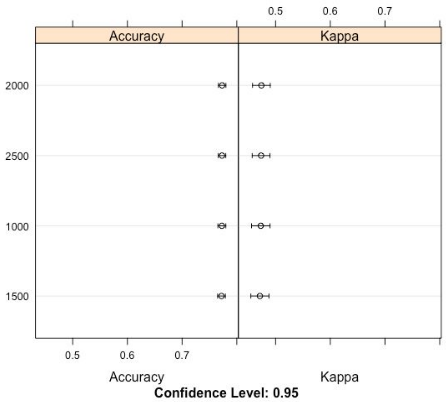
- Random Forests

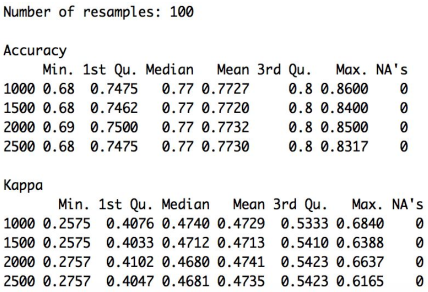
The first model used the entire dataset to identify non vulnerable households (level 4). To tune the parameter *m*, the number of splitting variables, we fit a series of random forests by setting a loop for *mtry* in 1 to 20. As with bagging, random forests do not overfit even if we increase the number of trees, so we just used a value of *B* sufficiently large for the error rate to settle down. Cross-validated number of splitting variables 3 and number of trees 2,000 yielded the highest result, even though the latter parameter did not matter too much as shown in the second plot. The first model gave us a cross-validated training accuracy rate of 77.3% and a testing accuracy rate of 75.67% - this was a very good result, so we explored the importance of variables plot and check which variables were of the most deterministic.



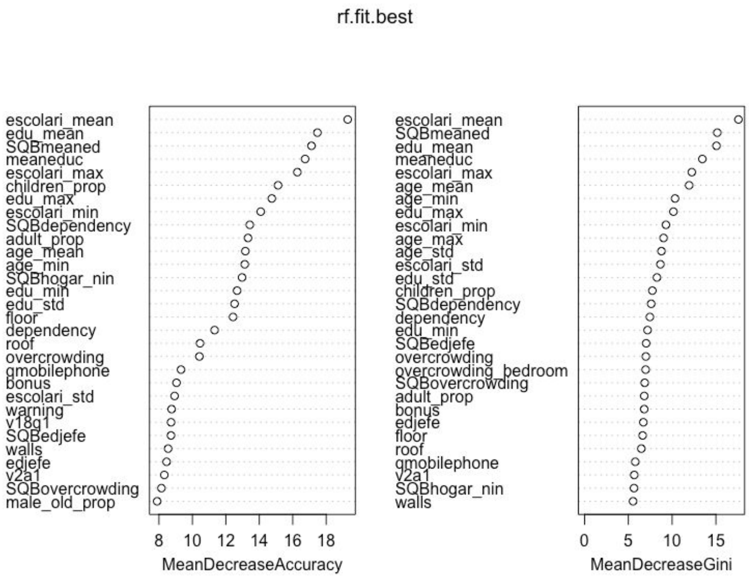


```
mtry  Accuracy   Kappa
 1    0.7134082  0.2331399
 2    0.7685083  0.4470349
 3    0.7731956  0.4743210
 4    0.7703025  0.4735164
 5    0.7708095  0.4770157
 6    0.7713956  0.4781642
 7    0.7708896  0.4782096
 8    0.7671893  0.4700699
 9    0.7679034  0.4721648
10    0.7685924  0.4729809
11    0.7665014  0.4690682
12    0.7684804  0.4729911
13    0.7670954  0.4698961
14    0.7652823  0.4661760
15    0.7664883  0.4683147
16    0.7683984  0.4731016
17    0.7662923  0.4685490
18    0.7658993  0.4678036
19    0.7646932  0.4649815
20    0.7666074  0.4693133
```



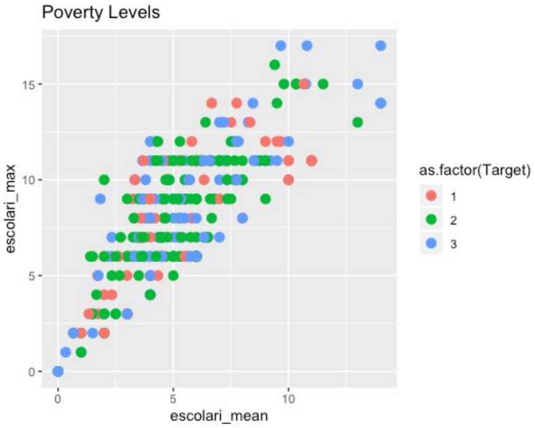


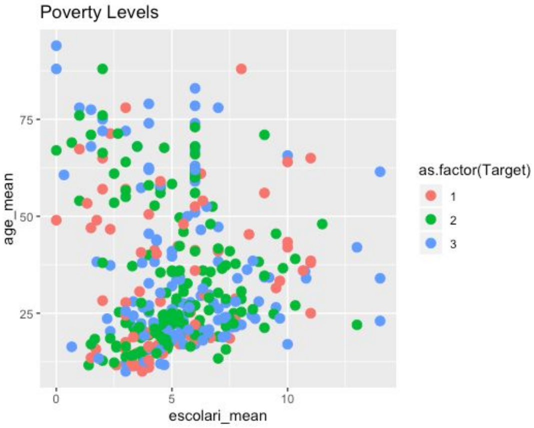
```
Number of resamples: 100

Accuracy
     Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
1000 0.68  0.7475   0.77 0.7727     0.8 0.8600    0
1500 0.68  0.7462   0.77 0.7720     0.8 0.8400    0
2000 0.69  0.7500   0.77 0.7732     0.8 0.8500    0
2500 0.68  0.7475   0.77 0.7730     0.8 0.8317    0

Kappa
       Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
1000 0.2575  0.4076 0.4740 0.4729  0.5333 0.6840    0
1500 0.2575  0.4033 0.4712 0.4713  0.5410 0.6388    0
2000 0.2757  0.4102 0.4680 0.4741  0.5423 0.6637    0
2500 0.2757  0.4047 0.4681 0.4735  0.5423 0.6165    0
```



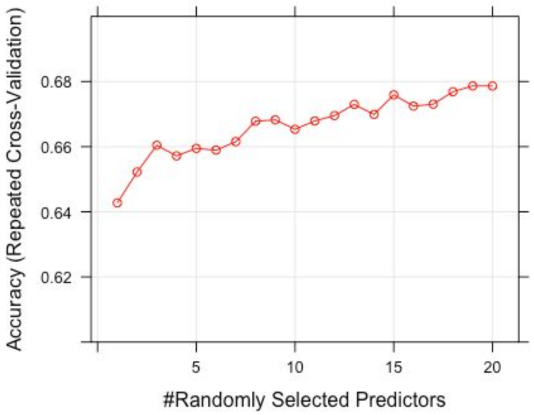
It is worthy of attention that **escolari_mean**, **edu_mean**, and **SQBmeaned** all have high levels of importance. Note that this might not be that all of them held true predictive power, but their high correlations among one another contributed to the result. Since each split only considered a subset of the possible variables, a variable that was correlated with an “important” variable may be considered without the “important” variable, which would cause the correlated variable to be selected for the split.

Before repeating the same processes for the second and third model, we plotted average years of education against the other two important variables, maximum years of education and average age, with points color-coded by the poverty levels. We suspected that these three classes are not well-separated.

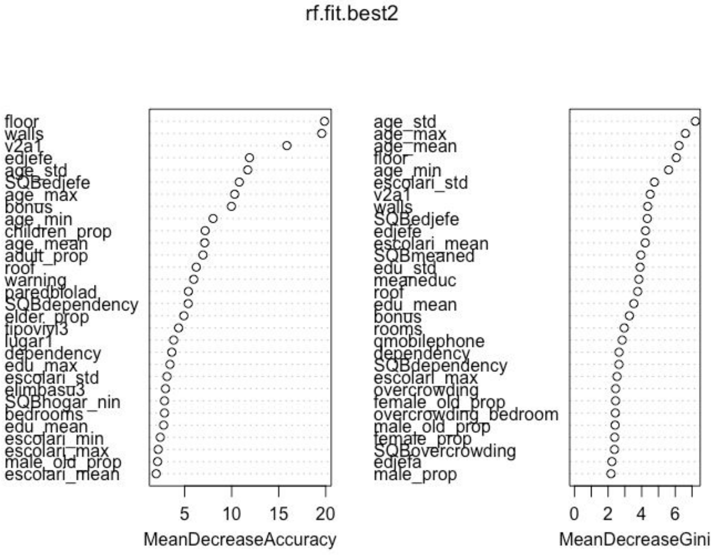




The second model used a subset of training data to identify vulnerable households (level 3) after filtering out the predicted non vulnerable households using the first model. We repeated the processes to tune the parameters. The optimal model gave us a cross-validated training accuracy rate of 67.9% and a testing accuracy rate as low as 54.3% - this was very poor compared to our other models. Here we consider two possible explanations: one is that because we used the predicted results from the first model, the error rate built up, which resulted in a less accurate second model; the other is that level 3 households were inherently not easy to distinguish from the rest. We will address potential improvements later.

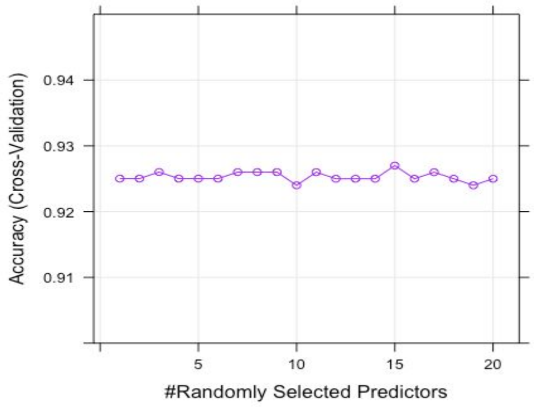


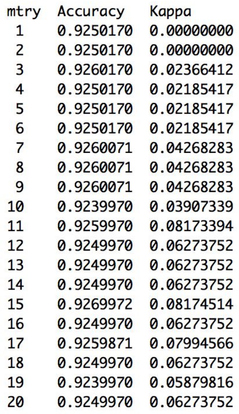
```
mtry  Accuracy   Kappa
 1    0.6427264  0.00000000
 2    0.6522157  0.07466409
 3    0.6604407  0.14271993
 4    0.6571723  0.14640105
 5    0.6594669  0.15902384
 6    0.6589290  0.16071455
 7    0.6615415  0.17190461
 8    0.6678067  0.19049233
 9    0.6682232  0.19429258
10    0.6653333  0.18732734
11    0.6679066  0.19431766
12    0.6695556  0.20187661
13    0.6729757  0.21051749
14    0.6698978  0.20607743
15    0.6759085  0.21819391
16    0.6724701  0.21481285
17    0.6730588  0.21536316
18    0.6768735  0.22471133
19    0.6786709  0.22973800
20    0.6786377  0.22696156
```



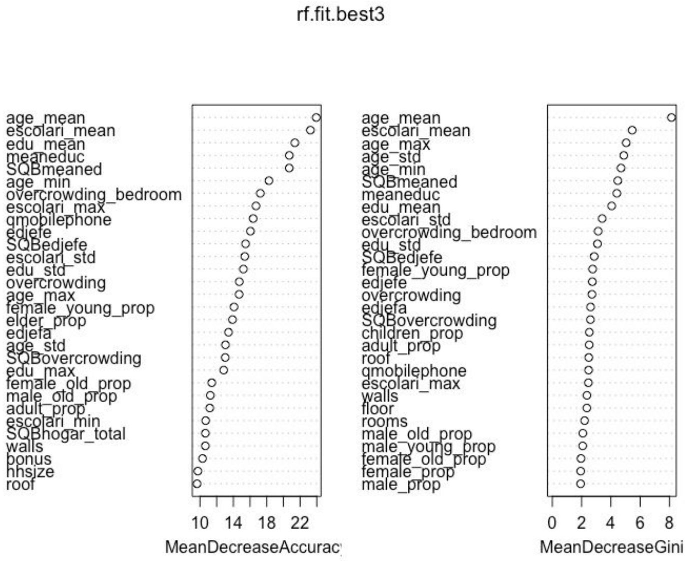
The third model first used a subset of training data (levels 1 and 2)[3] to train and the error rate was surprisingly low (~25%). We expected the extreme levels are easier to identify no matter what. To testify such expectation, we also tried building models just for level ones and threes only. The accuracy rate was quite high as well. Therefore, we merged the income levels 2,3,4 and built our final model to predict level ones on the entire dataset instead. We repeated the processes again and the optimal model gave us a cross-validated training accuracy rate of 92.7% and a testing accuracy rate as high as 91.6% - this was an excellent result, even better than the first model.

[3] We did not train this model based on the predicted results of models one and two; otherwise, we would have got extremely low accuracy since our second model was not so good. This might cause trouble if we actually implement it since the third model built independently from the others might result in conflicting nodes due to node impurity.

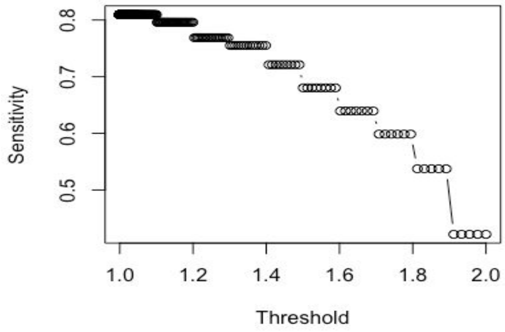


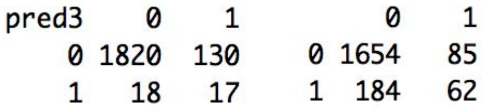


```
mtry  Accuracy   Kappa
 1    0.9250170  0.00000000
 2    0.9250170  0.00000000
 3    0.9260170  0.02366412
 4    0.9250170  0.02185417
 5    0.9250170  0.02185417
 6    0.9250170  0.02185417
 7    0.9260071  0.04268283
 8    0.9260071  0.04268283
 9    0.9260071  0.04268283
10    0.9239970  0.03907339
11    0.9259970  0.08173394
12    0.9249970  0.06273752
13    0.9249970  0.06273752
14    0.9249970  0.06273752
15    0.9269972  0.08174514
16    0.9249970  0.06273752
17    0.9259871  0.07994566
18    0.9249970  0.06273752
19    0.9239970  0.05879816
20    0.9249970  0.06273752
```



However, we should be very mindful that because we had very few observations in level ones, the misclassification rate should therefore be considered a poor benchmark for evaluation - the model could just predict as few as observations as non-extreme households and still generate very good accuracy rate. But if we wanted to find out the most needy households, we needed to optimize sensitivity (true positive rate). Our solution was to bootstrap non-extreme households data (level 2, 3, and 4) and create 10 separate sample sets, each with 100 observations. Then we combined the level ones with each sample set so that the extremes and the non-extremes were balanced in each new dataset. We then utilized the idea of bagging and built trees repeatedly upon each new dataset. Finally, we made ten sets of predictions on the testing data using the ten models and tuned the cut-off threshold to decide on the final predictions. As a result, as shown below, of all the extremely poor households, our modified model correctly predicted more of them (right), with a true positive rate of 42.2% , compared to the previous model (left) which only had a true positive rate of 11.6%. However, this came at the cost of sacrificing overall accuracy rate.

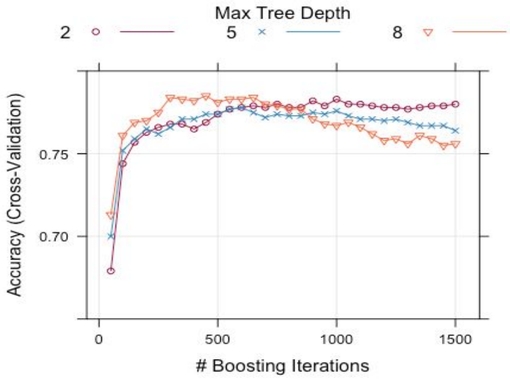


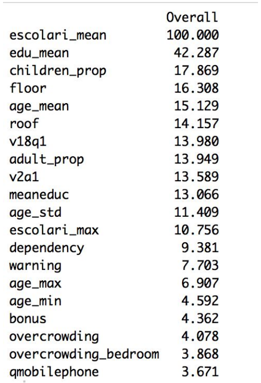

```
pred3    0    1          0    1
    0 1820  130      0 1654   85
    1   18   17      1  184   62
```

- Boosting

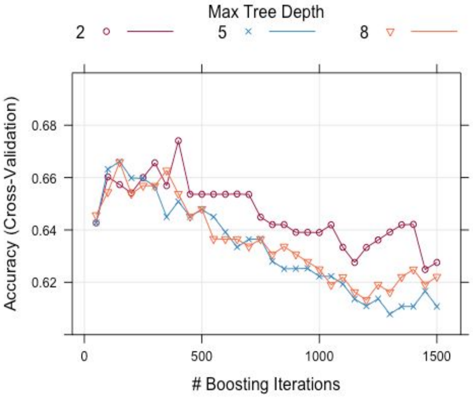
Trees of random forest are built on a bootstrapped dataset, independent of other trees. In "slow learner" boosting, however, trees are grown sequentially: the growth of each tree uses information from previously grown trees. In gradient boosting, each tree is growing to the residuals left over from the previous collection of trees and fitting small trees to the residuals. In order to perform boosting, we need to select 3 parameters: number of trees *B*, tree depth *d*, and step size λ. Unlike random forests, the number of trees is a tuning parameter because if we have too many we can overfit. The second parameter is complexity of the tree - we usually select the number of terminal nodes to be between 2 and 8. The last parameter is the learning rate, which we set to 0.01. The plots below show the tuning outcomes of performing gradient boosting in R.

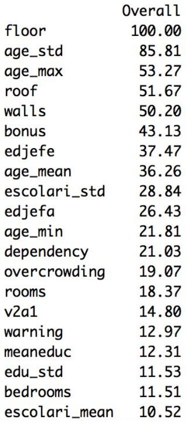
The final values used for the first model were n.trees = 450, interaction.depth = 8, shrinkage = 0.01. The cross-validated training accuracy rate was 77.4% and the testing accuracy rate was 76.7%

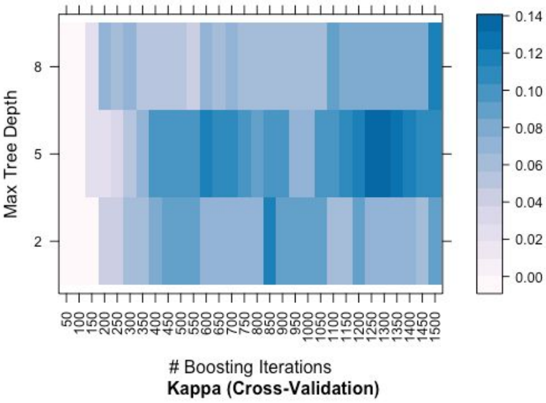


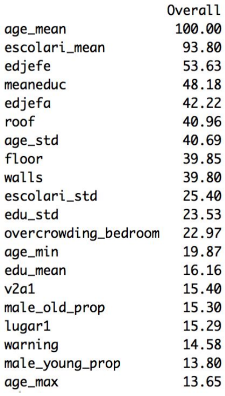

```
                     Overall
escolari_mean        100.000
edu_mean              42.287
children_prop         17.869
floor                 16.308
age_mean              15.129
roof                  14.157
v18q1                 13.980
adult_prop            13.949
v2a1                  13.589
meaneduc              13.066
age_std               11.409
escolari_max          10.756
dependency             9.381
warning                7.703
age_max                6.907
age_min                4.592
bonus                  4.362
overcrowding           4.078
overcrowding_bedroom   3.868
qmobilephone           3.671
```

The final values used for the model were n.trees = 400, interaction.depth = 2, shrinkage = 0.01. The cross-validated training accuracy rate was 67.5% and the testing accuracy rate was 51.0%.




```
              Overall
floor          100.00
age_std         85.81
age_max         53.27
roof            51.67
walls           50.20
bonus           43.13
edjefe          37.47
age_mean        36.26
escolari_std    28.84
edjefa          26.43
age_min         21.81
dependency      21.03
overcrowding    19.07
rooms           18.37
v2a1            14.80
warning         12.97
meaneduc        12.31
edu_std         11.53
bedrooms        11.51
escolari_mean   10.52
```

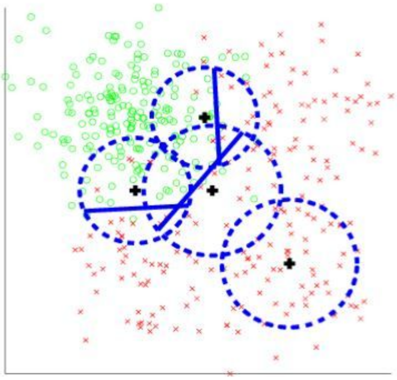



```
                     Overall
age_mean              100.00
escolari_mean          93.80
edjefe                 53.63
meaneduc               48.18
edjefa                 42.22
roof                   40.96
age_std                40.69
floor                  39.85
walls                  39.80
escolari_std           25.40
edu_std                23.53
overcrowding_bedroom   22.97
age_min                19.87
edu_mean               16.16
v2a1                   15.40
male_old_prop          15.30
lugar1                 15.29
warning                14.58
male_young_prop        13.80
age_max                13.65
```

For the final model, because we used the entire dataset for training, there was a low percentage of level ones, we used metrics "Kappa" instead of "Accuracy" to improve the quality of the final model. The final values used for the model were n.trees = 300, interaction.depth = 2, shrinkage = 0.01. The cross-validated training accuracy rate was 92.7% and the testing accuracy rate was 92.6%. We could have applied the same procedures as we did when implementing random forests utilizing the idea of bagging, i.e. we build trees repeatedly on balanced sets of data with all of observations from level 1 and bootstrapped samples from level 2,3,4. In general, the results for random forests and boosting were very similar in terms of the important variables selected. This should not be surprising since they are both machine learning mechanisms built on decision trees.

Given that our second model had such a low accuracy rate compared to the first and third ones, we also considered running models only for predicting level 2s and level 3s. The training error rate was as low as 0%, but the testing error rate was too high. All in all, we concluded that non-vulnerable households (level 1) and households in extreme poverty (level 4) are easier to distinguish, but since vulnerable households (level 3) and households in moderate poverty (level 2) are more similar and we were uncertain about the labeling criteria, it was hard for our models to make correct predictions. However, we still proposed two other possible remedies.

i) KNN



K-Nearest Neighbours (KNN) model is a non-parametric supervised learning technique which aims to classify each new pattern using the evidence of nearby observations.(Denoeux,1995) A model which shows the relationship between predictors and targets, is mapped by observing the nearest observations. We wished to employ this classification model[4] because we were unsure of the conditional density function of each **Target** class.

KNN Prediction Results

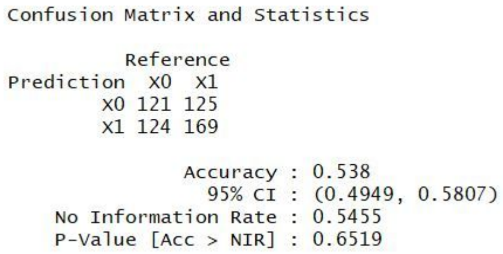
[4] In this model, our Target (1,2,3,4) is leveled to Target (X1,X2,X3,X4)

Predicting Target Level	K parameter used	Prediction accuracy
nonvulnerable	15	66.2%
poverty	23	59.77%
extreme	23	65.31%

The prediction results of KNN model showed that it was not as accurate as other models employed above.[5] However, the model helped us to verify that the KNN model could be employed to analyse our dataset.

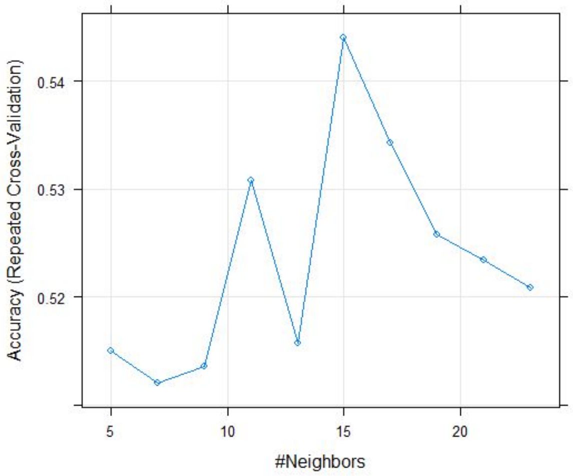
KNN Prediction of Target Level 2 and 3

Our previous models employed showed good prediction results for predicting non-vulnerable (Target 4) and extreme (Target 1) households. Using the KNN model, we wished to see if it was possible for us to distinguish between Target level 2 and 3 households. The dummy variable "poverty" showed 1 if the household belongs to Target level 2. When "poverty" was equal 0, it meant that the household was in Target level 3.



```
Confusion Matrix and Statistics

          Reference
Prediction  X0  X1
        X0 121 125
        X1 124 169

               Accuracy : 0.538
                 95% CI : (0.4949, 0.5807)
    No Information Rate : 0.5455
    P-Value [Acc > NIR] : 0.6519
```

Confusion matrix and prediction accuracy

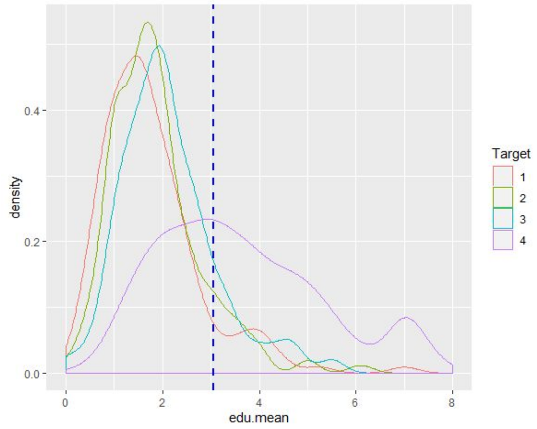


Accuracy plot to determine the K parameter

Our prediction result for the model was only 53.8%, which was significantly lower than other model results. This confirms with our previous findings: it is easy to distinguish extreme levels of welfare,

[5] The K parameter was chosen to maximise model accuracy

non-vulnerable and extreme poverty, but not so much with the middle ranges.



For example, if we look at the mean years of education density plot, it is easy to see that there is a strong similarity in distribution between Target level 2 and 3. KNN is a supervised learning model which will be as good as the quality of data that is inputted. Since our training data is skewed and show small differences between Target 2 and 3, our KNN model's use is limited in our study. As such, we may need a more sophisticated model to pick up the weak difference in characteristics between households in Target 2 and 3. Otherwise, we need more descriptive information about the income level labels.

ii) Neural Network

A neural network is a two-stage regression or classification model, as represented by a network diagram. A neural network is essentially a nonlinear statistical model. Because we are interested in differentiating class 1, 2, and 3, there are 3 units at the top of the network, with each unit modeling the probability of the observation in each class. There are 3 target measurements Yk, for k = 1, 2, 3, each being coded as a 0−1 variable for the kth class. Derived features (hidden units) in the middle of the network are created from linear combinations of the original inputs X, and then the target Yk is modeled as a function of linear combinations of them. Because the hidden units are not directly observed and we also have many unknown parameters (weights), neural network is less effective for problems where the goal is to describe the process that generated the predictions and the roles of individual variables. As a result, after implementing the *nerualnet* package in R, we got a ~34% testing error rate but could not interpret beyond the result. All in all, the mechanism is more useful if prediction rather than interpretation is the goal. (Hastie, et al., 2017)

IV. Conclusion

Model 1

	v2a1	qmobilephone	female_prop	female_young_prop	female_old_prop	children_prop	adult_prop	edu_mean[6]	edu_max	edu_min	edu_std	area1	floor	roof	walls	dependency	age_mean	age_min	overcrowding
Logistic	X	✔	✔	✔	✔	✔	X	✔	X	X	X	✔	✔	✔	✔	X	X	X	X
RF	X	✔	X	X	X	✔	✔	✔	✔	✔	✔	X	✔	X	✔	✔	✔	✔	✔
Boosting	✔	✔	X	X	X	✔	✔	✔	✔	✔	X	X	✔	✔	✔	✔	✔	✔	✔

Model 2

	v2a1	qmobilephone	female_prop	female_young_prop	female_old_prop	children_prop	adult_prop	edu_mean	edu_std	area1	floor	roof	walls	dependency	edjefe	bonus	warning	age_max	age_std	age_min	age_mean
Logistic	X	X	✔	✔	✔	X	X	X	X	✔	✔	X	X	X	X	X	X	X	X	X	X
RF	✔	X	X	X	X	✔	✔	✔	✔	X	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Boosting	✔	X	X	X	X	X	X	✔	✔	X	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

Model 3

	qmobilephone	female_prop	female_young_prop	female_old_prop	male_young_prop	adult_prop	edu_mean	edu_std	edu_max	edjefe	edjefa	overcrowding	age_mean	age_min	age_max	age_std	roof
Logistic	X	✔	✔	✔	✔	X	✔	X	X	X	X	✔	X	X	X	X	X
RF	✔	X	✔	X	X	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Boosting	X	X	X	X	✔	X	✔	✔	X	✔	✔	✔	✔	✔	X	✔	✔

The objective of our project was to improve upon the modeling methodology of the Proxy Means Test (PMT), the standard procedure used to estimate poverty in developing areas that lack proper records of income. Our approach and analysis succeeded in many regards, but also fell short of our expectations in others. Despite any shortcomings, we believe that we offer valuable insight into identifying key indicators of poverty and possible solutions to the data problems that caused difficulties when the PMT is utilized in practice.

Before we discuss the key takeaways from our analysis, we should acknowledge a key structural difference between our data and the data used by NGOs and governments when building PMT models. When the standard PMT model is applied it estimates the level of income for a household, generally

[6] This includes mean_educ, escolari_mean, SQBmeaned

specified as an amount of the given country's currency. Our data, on the other hand, is broken up into four categories each representative of a distinct type of household wealth (non-vulnerable, vulnerable, moderate poverty, extreme poverty). From the standpoint of policymakers and others interested in identifying poor households, this is only a nominal difference. Poverty is context based, thus it is more important to determine where a household stands relative to other households than it is to identify the particular household's income. Thus, our modeling results have similar significance to other PMT models as both rank households respective to the general population. For our purposes, target level one is representative of the 7.5 percentile of incomes and below, target levels one and two combined are roughly the 22nd percentile and below, target levels one, two, and three combined are roughly the 35th percentile and below, with target level four making up the rest.

Thus, we can compare the accuracy of our models to the accuracy of models used by actual practitioners of PMT. For model one, representative of predicting poverty at roughly a 35 percent coverage level, our logistic, random forests, and boosting models all outperformed the best PMT examples targeting similar levels found in *Kidd and Wylde* (2011) by up to five percentage points. For model two, representative of predicting poverty at roughly a 22 percent coverage level, our models performed more or less the same as analogous PMT examples found in *Kidd and Wylde* (2011). Finally our model three analysis, representative of predicting the lowest levels of poverty at roughly a 7.5 percent coverage level showed better accuracy than PMT attempts at similar levels found in *Kidd and Wylde* (2011) across the board. Notably, our random forest and boosting models yielded accuracy levels of over 90 percent on testing data. While these comparisons are not completely *ceteris paribus*, as many times implementation in practice runs into issues unforeseen in modeling environments, our results show significant improvements relative to PMT classification done in the past, particularly when identifying the poorest households in our dataset.

Evident in our data analysis are a number of important findings. First, it is apparent from the performance of our models that the most problematic classification targets are not households at the upper and lower ends of society, target levels one and four in our case, but households on the borderline of poverty. Even in our attempts to separate these households directly, excluding the households with higher and lower incomes, we were unable to produce significant gains in accuracy. Intuitively, this result is unsurprising. In countries with incomes below the world average such as Costa Rica (Trading Economics, 2019), the line distinguishing poor and vulnerable households is largely symbolic and these households very similar in most regards. Accordingly, in our analyses we found very few features that could differentiate the two on a significant basis.

With regards to poverty as a whole, however, we were able to establish particular features that were indicative across our models. In our logistic, random forests, and boosting attempts to separate out non-vulnerable households we found that variables measuring of technology (**qmobilephone**), household demographics (**children_prop**), education (**edu_mean**), and house material (**floor, walls**) were important in classification. Our measure of floor quality (**floor**), however, was also the only consistently significant variable in separating households at the borderline of poverty in all three models. Lastly, education (**edu_mean**) and household demographics (**overcrowding)** proved to be critical features in classifying extremely poor households throughout.

Finally, we believe a large degree of our success in improving accuracy rates can be attributed to our management of data and the methodology we employed to formulate our classifications. At the onset of the project we were given a large data set with a majority of features being dummy variables indicating various aspects of the household. We engineered a number of variables combining these dummy variables into an ordinal ranking pertaining to their respective feature i.e. floor composition, technology, etc. to better match our objective. Given that many of these engineered features appeared as significant classifiers throughout our analyses we believe their addition improved the predictive power of our models. Secondly, the approach we took to classification differed significantly from that of the standard PMT. Ordinarily, PMT is used to project the monetary value of the income of a particular household, then the household is classified as poor or aid-deserving based on that estimate. Our approach, however, treated the issue as a classification problem directly. This method allowed us to train our models on subsets of the data, operating under the assumption that low-income households of various levels (**target = 1,2,3**) were more similar to each other than non-vulnerable households. Seeing that our models generated by these subsets were very different from the models trained on the entire data set, we think this this assumption and approach has merit. Based on our results, we would recommend pairing classification tree methods such as random forests or boosting with this approach, as those analyses yielded the highest prediction accuracy. Furthermore, if an aid organization or government simply wished to treat poverty as a multi-tier classification problem, we recommend the use of neural network analysis, as our application of neural networks to poverty classification gave us accuracy rates well above those found in previously implemented programs.

Overall, we find that our analyses provided significant improvements in the prediction of poverty and identifying important features in poor households, especially with regards to separating out non-vulnerable households and identifying households living in extreme poverty. Although some of these gains in prediction power may be lost when taken out of a training and testing environment and put into practice, we believe that our identification of important features and our approach to classification can be useful in any case. In conclusion, we believe that if NGOs and policymakers treat poverty as a classification problem first, they have room to incorporate machine-learning and more advanced data mining techniques to improve upon the PMT approach to poverty identification. Hopefully, in doing so, they will be better equipped to fight poverty in the future.

V. Bibliography

1. Alatas, V. et al (2012). *Targeting the Poor: Evidence from a Field Experiment in Indonesia.* American Economic Review, Cambridge, MA
2. Brownlee, J. (2019). *How to Configure the Gradient Boosting Algorithm*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/configure-gradient-boosting-algorithm/ [Accessed 10 Feb. 2019]
3. Denoeux, T. (1995). A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(5), pp.804-813
4. Hastie, T., Friedman, J. and Tibshirani, R. (2017). *The elements of statistical learning*. New York: Springer, pp.389-409
5. Kidd, S. and Wylde, E. *Targeting the Poorest: An assessment of the proxy means test methodology*. Australian Agency for International Development, Canberra
6. Kidd, S., Gelders, B., and D. Bailey-Athias (2017). *Exclusion by design: An assessment of the effectiveness of the proxy means test poverty targeting mechanism.* International Labor Organization and Development Pathways, Geneva
7. Koehrsen, W. (2018). *Costa Rican Household Poverty Level Prediction*. [online] Kaggle. Available at: https://www.kaggle.com/willkoehrsen/a-complete-introduction-and-walkthrough?fbclid=IwAR1-sWPqiulePAHuumndx5phsVMl3LGoWbD3nBUpcpGghhvRfezo5MisCFo [Accessed 10 Feb. 2019]
8. Moise, I. (n.d.). *K-Nearest Neighbour Classifier*. [online] Ethz.ch. Available at: https://www.ethz.ch/content/dam/ethz/special-interest/gess/computational-social-science-dam/documents/education/Spring2015/datascience/K-Nearest-Neighbour-Classifier.pdf [Accessed 10 Feb. 2019]
9. Trading Economics. *Costa Rica GDP per Capita.* https://tradingeconomics.com/costa-rica/gdp-per-capita [Accessed 10 Feb. 2019]
10. Veras, F., Peres, R. and Guerreiro, R. (2007). *Evaluating the Impact of Brazil's Bolsa Família: Cash Transfer Programmes*. In Comparative Perspective, IPC Evaluation Note No. 1, International Poverty Centre: Brasilia, Brazil.

Harvard Formatting